



3 Subcapa de acceso al medio

3.3 Conmutación de nivel 2 (1/3)

RdE 2014-2015

3 Guión del Tema 3

- 3. SUBCAPA DE ACCESO AL MEDIO.
 - 3.1 Protocolos MAC.
 - 3.2 Estándares IEEE.
 - **3.3 Conmutación de nivel 2.**



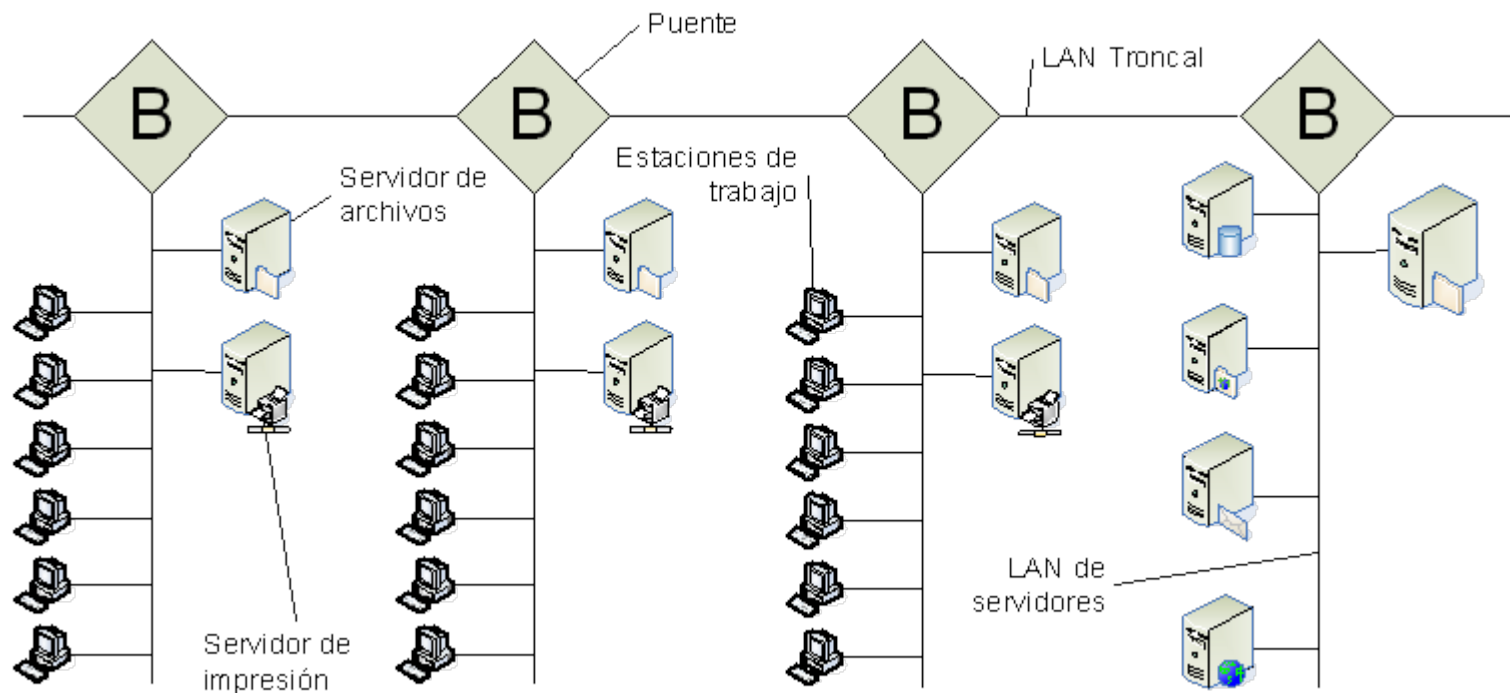
3.3 Agregación LAN

¿Por qué? limitaciones:

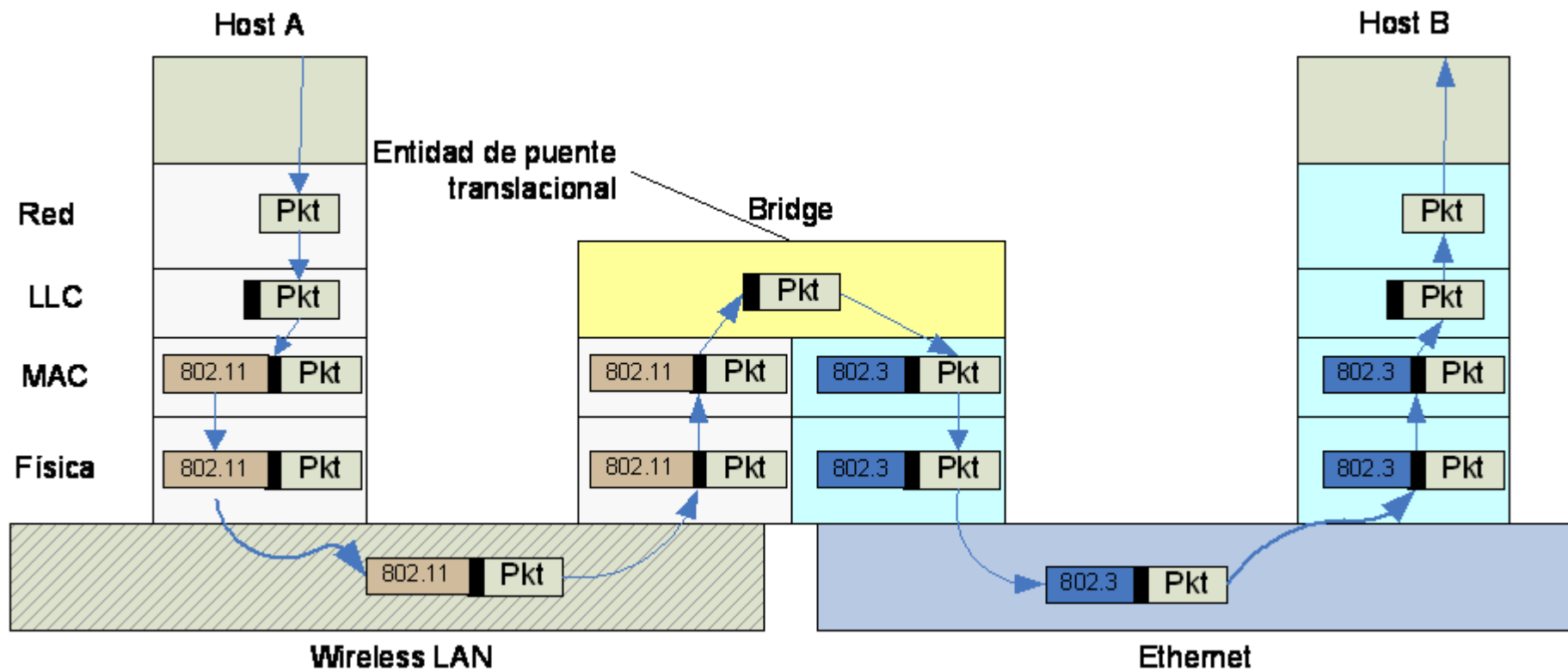
1. Número de estaciones.
2. Tamaño de la red.
3. Ancho de banda.

Ventajas:

1. Seguridad (separación broadcast).
2. Fiabilidad (fallos locales).
3. Interoperabilidad (802.3-802.11).

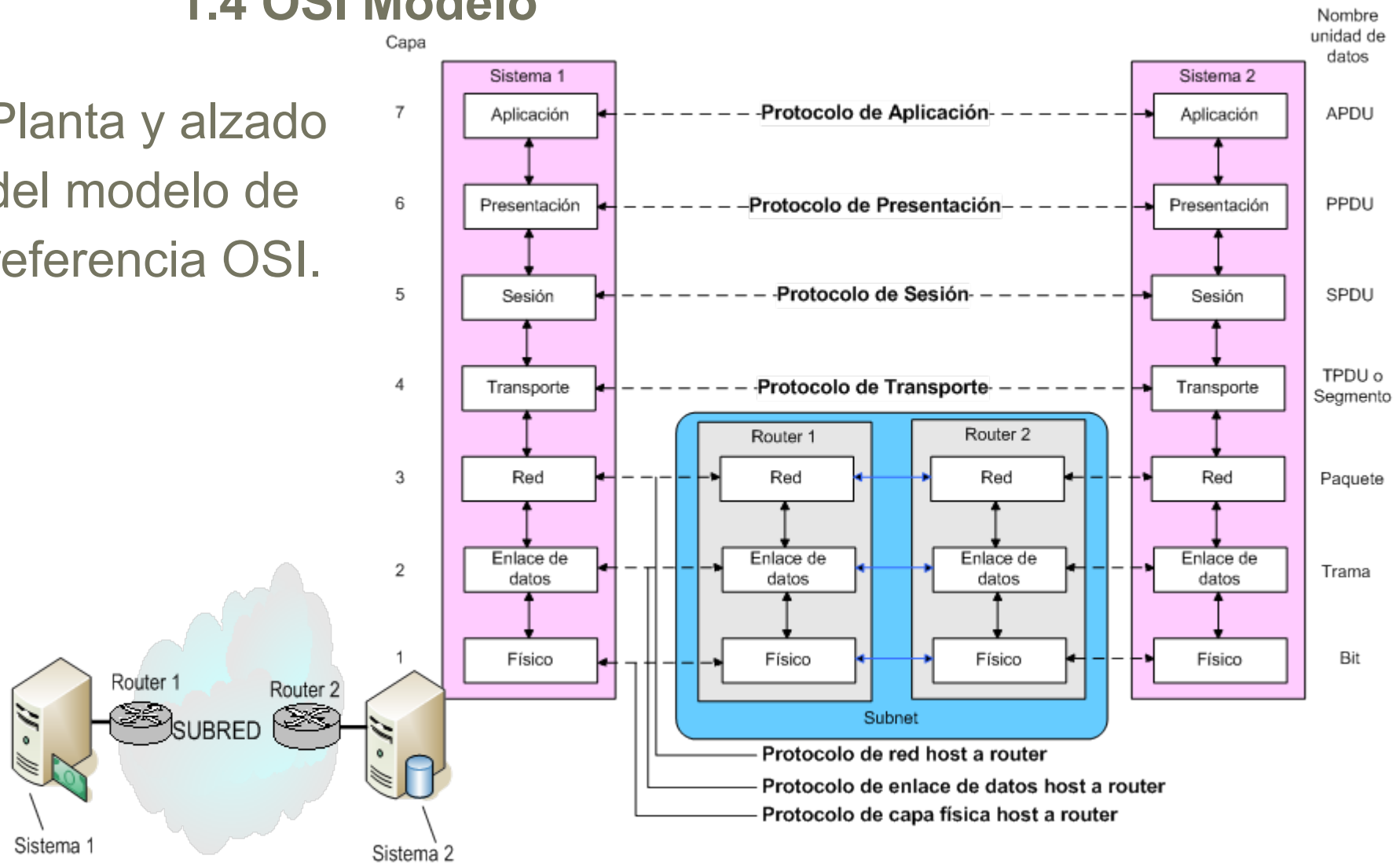


3.3 Puentes entre 802.x y 802.y



1.4 OSI Modelo

Planta y alzado del modelo de referencia OSI.





3.3 Tramas

Diferentes tipos de tramas en cada medio.

802.3	Dirección destino		Dirección Origen		Longitud	Datos			Relleno		Checksum							
802.11	Control de trama		Duración	Dirección 1		Dirección 2		Dirección 3		Secuencia	Dirección 4		Datos		Checksum			
802.16	0	EC	Tipo			C	EK		Longitud		ID conexión		CRC Cabecera		Datos		Checksum	



3.3 Funciones de los puentes (*bridges*)

Funciones:

- ☐ **Leer** todas las tramas, modo promiscuo, que pasan por sus interfaces, y **aceptar** aquellas cuya dirección conozca a través de otro interfaz diferente al cual se ha recibido (puente transparente). También es posible utilizar la información contenida en la cabecera para realizar el reenvío (puentes *source routing*).
- ☐ **Retransmitir** las tramas por el interfaz adecuado.
- ☐ **Reformatear** la trama recalculando el *checksum*, y tratando los campos de prioridad si los hubiese.

• Consideraciones:

- ☐ Los puentes no modifican el contenido de las tramas.
- ☐ Los puentes disponen de buffer para absorber los picos.
- ☐ Los puentes conocen todas las direcciones MAC (puentes transparentes).
- ☐ Pueden tener múltiples interfaces: puentes multipuerto y conmutadores.

3.3 Tipos de puentes (*bridges*)

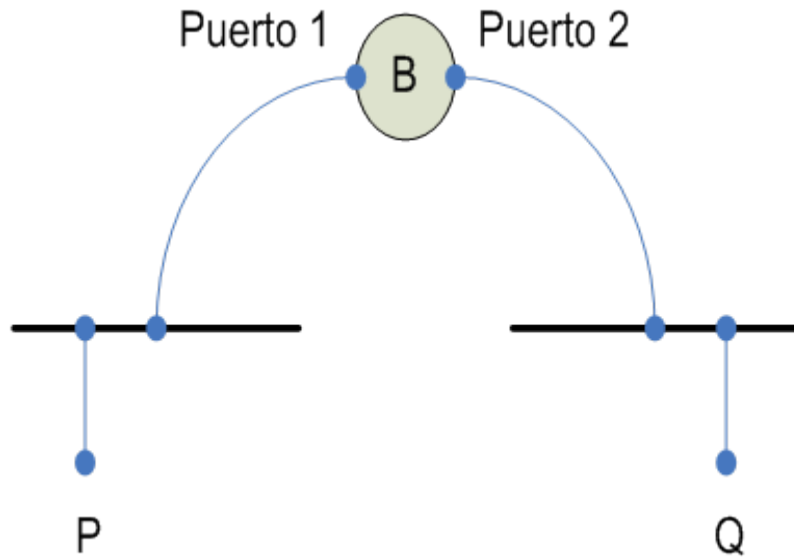
- **Por su funcionamiento:**
 - ☐ **Transparentes (802.1):** actúan de forma transparente. No se requiere modificación alguna en las estaciones.
 - ☐ **Con encaminamiento desde el origen (802.5):** las estaciones deben indicar el camino que seguirá la trama.
- **Por su interoperabilidad**
 - ☐ **Homogéneos:** solo interconectan LANs con el mismo formato de trama (p. ej. 802.3-802.3, o bien 802.11-802.11)
 - ☐ **Heterogéneos o Traductores:** interconectan LANs con diferente formato de trama (ej. 802.3-802.11)
- **Por su alcance.**
 - ☐ **Locales:** interconectan LANs directamente.
 - ☐ **Remotos:** enlazan LANs a través de conexiones WAN (líneas dedicadas, enlaces X.25, Frame Relay, ATM, RDSI, etc.).



3.3 Puentes transparentes

- Inicialmente desarrollados por DEC.
- Los puentes 802.1 tienen las siguientes características:
 1. La escucha promiscua y la capacidad de almacenamiento y reenvío (***store and forward***) del puente simple.
 2. El aprendizaje de direcciones (***FDB Forwarding DataBase***) del puente con aprendizaje.
 3. El algoritmo de árbol de expansión (***Spanning Tree***) del puente completo.

3.3 Puentes transparentes (1) Puente simple



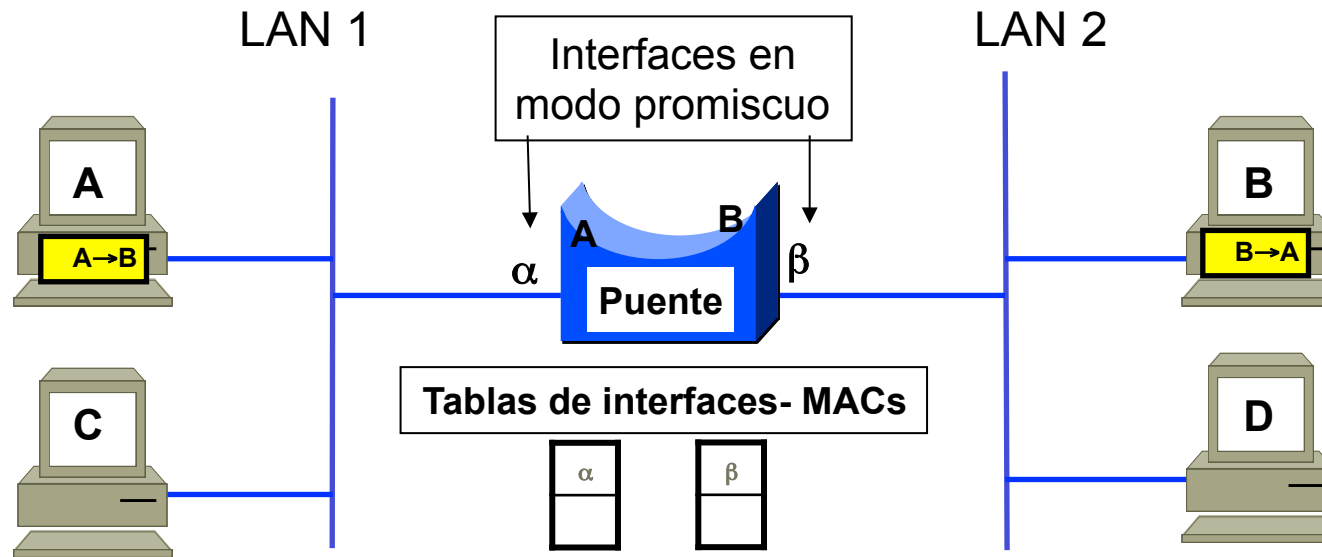
- Un puente transparente **lee** todas las tramas, modo **promiscuo**, que pasan por sus interfaces, y lo reenvía por todos los interfaces excepto por el que lo ha recibido.
- Un **repetidor** transmite bit a bit, por lo que puede haber colisión. Un **puente** almacena toda la trama y espera a que el medio esté libre. Tiene su propio nivel MAC.
- El puente tiene una memoria intermedia o **cache** en donde se almacenan las tramas a la espera de ser reenviadas. Puede haber pérdidas (*drops*) si hay un exceso de tráfico entre segmentos.



3.3 Puentes transparentes (2) Puente con aprendizaje (1)

- El puente con aprendizaje “conoce” las MAC de las estaciones conectadas a cada segmento (**FDB**).
- La estrategia de aprendizaje es la siguiente:
 1. El puente **escucha** todas las tramas de modo promiscuo en todos sus puertos.
 2. Para cada trama recibida, el puente **guarda** la dirección MAC **origen** en una memoria *cache* junto al puerto por el que la trama ha ingresado (FDB).
 3. Para cada trama recibida, el puente **busca** en la FDB la dirección MAC destino de la misma:
 - a. Si la dirección no se encuentra, el puente **reenvía** la trama por **todos** las interfaces excepto por el que ha sido recibida.
 - b. Si la dirección se encuentra, el puente **reenvía** la trama por la interfaz que figura en la tabla. Si el interfaz es por el que ha sido recibida, la trama se tira.
 4. El puente **elimina** cada entrada de la FDB pasado un periodo de tiempo, llamado **aging time** (típicamente 300 s), durante el cual no se ha recibido tráfico con esa dirección fuente. Esto permite la movilidad de estaciones.

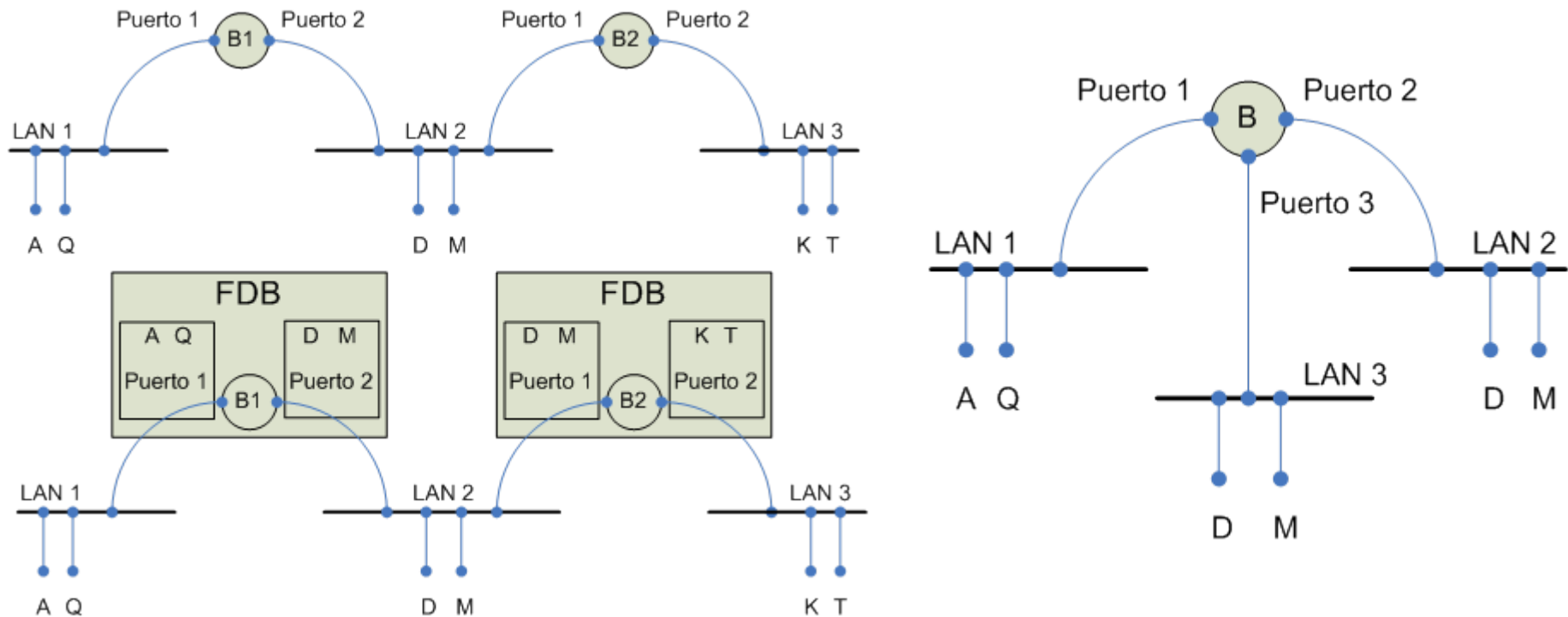
3.3 Puente con aprendizaje (2) Funcionamiento



1. A genera una trama con destino B que el puente recibe por α
 2. El puente busca a B en la tabla de direcciones de α ; si le encuentra descarta la trama, si no la reenvía por β
 3. El puente incluye A en la lista de direcciones de la interfaz α
 4. Cuando B envíe una trama de respuesta el puente le incluirá en la lista de la interfaz β
- La FDB solo se actualizan con **direcciones origen**. Si una estación nunca emite una trama (o no pone la dirección origen) su dirección no estará en la FDB.

3.3 Puente con aprendizaje (3) Topología

- El concepto de puente con aprendizaje funciona para topologías libres de bucles (loop-free).





3.3 Puente con aprendizaje (4) Tipos de tráfico

- El puente reenvía los siguientes tipos de tráfico:
 - ☐ Dirigido a dirección MAC, **Unicast**, conocida a través de otro puerto del puente. Sólo a través de ese puerto del puente.
 - ☐ Dirigido a **Broadcast** o **Multicast**. Por todos los puertos, excepto por el puerto de origen. No aparecen nunca como direcciones origen y no estarán nunca en la FDB.
 - ☐ Dirigido a dirección MAC no conocida (**UNK**), no está en la FDB. Por todos los puertos, excepto por el puerto de origen.
- El puente no altera la trama. La dirección MAC de origen es la inicial.
- Los puertos del puente tienen sus propias direcciones MAC, diferente para cada puerto.
- La FDB es exhaustiva, cada MAC es diferente. Tiene un tamaño limitado entre 2000-8000 conmutadores pequeños, a 30000 conmutadores grandes.



3 Subcapa de acceso al medio

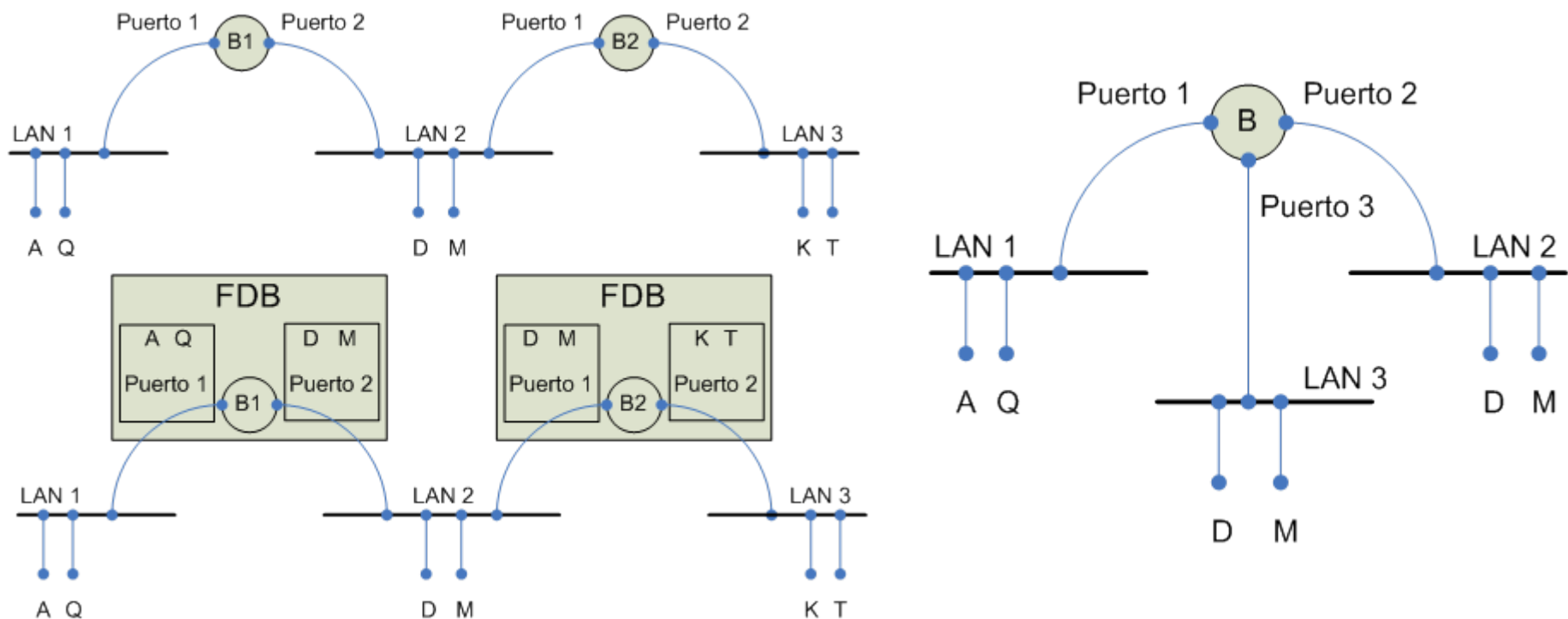
3.3 Conmutación de nivel 2 (2/3)



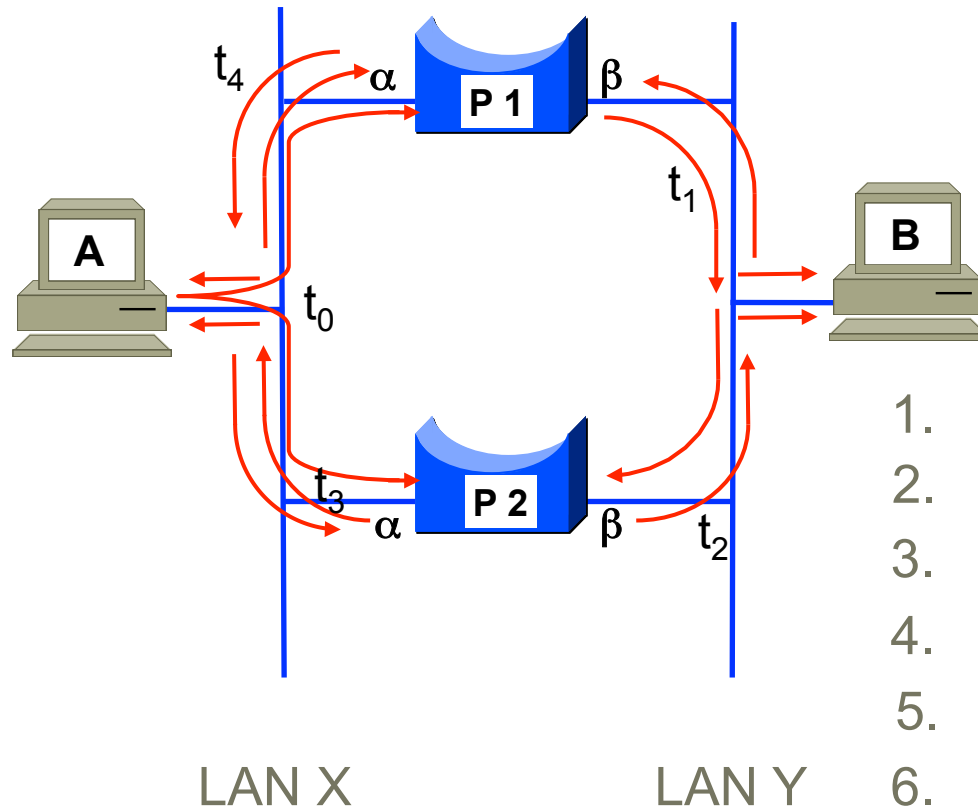


3.3 Puentes transparentes (3) *Spanning Tree* (1)

- El concepto de puente con aprendizaje funciona para topologías libres de bucles (*loop-free*).



3.3 Spanning Tree (2) Bucle

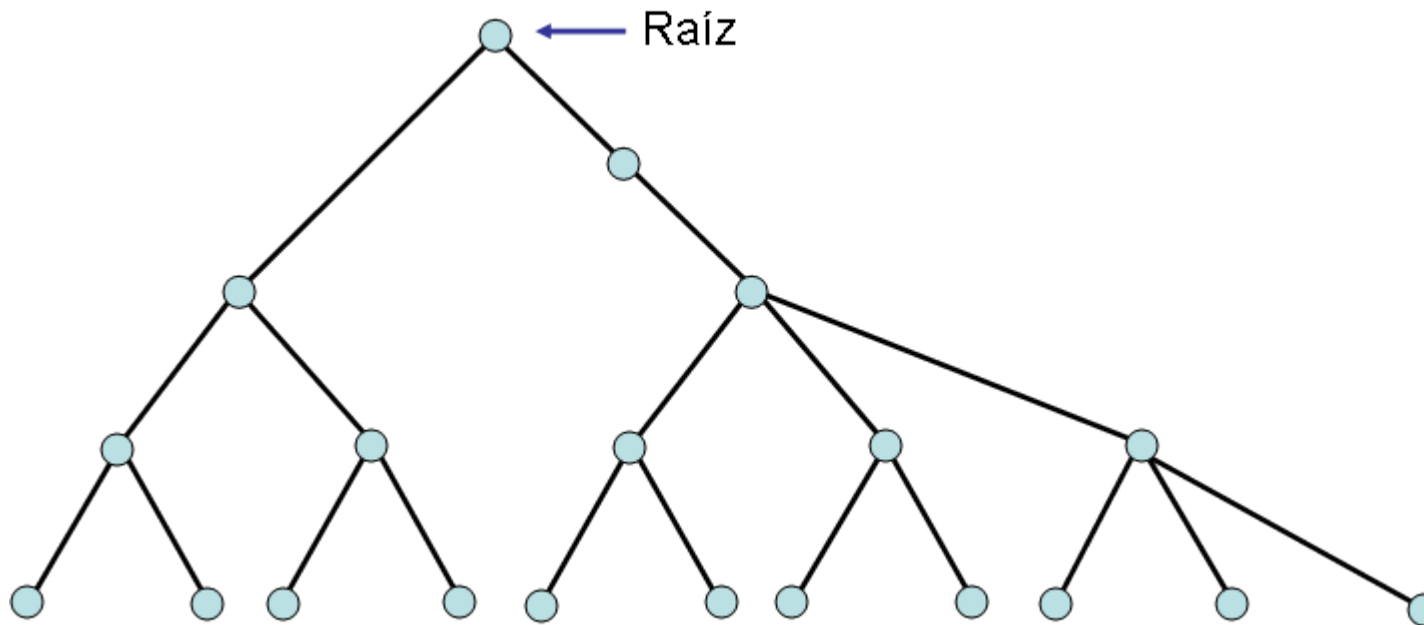


1. A envía trama t_0 a LAN X
2. P1 retransmite t_0 en LAN Y como t_1
3. P2 retransmite t_0 en LAN Y como t_2
4. P2 retransmite t_1 en LAN X como t_3
5. P1 retransmite t_2 en LAN X como t_4
6. ... y así sucesivamente.

Con la primera trama transmitida la red se satura eternamente.

3.3 *Spanning Tree* (3) Árbol

Un ***Spanning Tree***, o árbol de expansión, es un grafo en el que solo hay un camino posible entre dos nodos (un árbol sin bucles).



Si podemos pintar una red de puentes transparentes como un *spanning tree*, entonces el problema del bucle no puede darse. El objetivo del protocolo denominado *Spanning Tree* es evitar que la red tenga bucles



3.3 Spanning Tree (4)

Algorhyme

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

Algoritmo

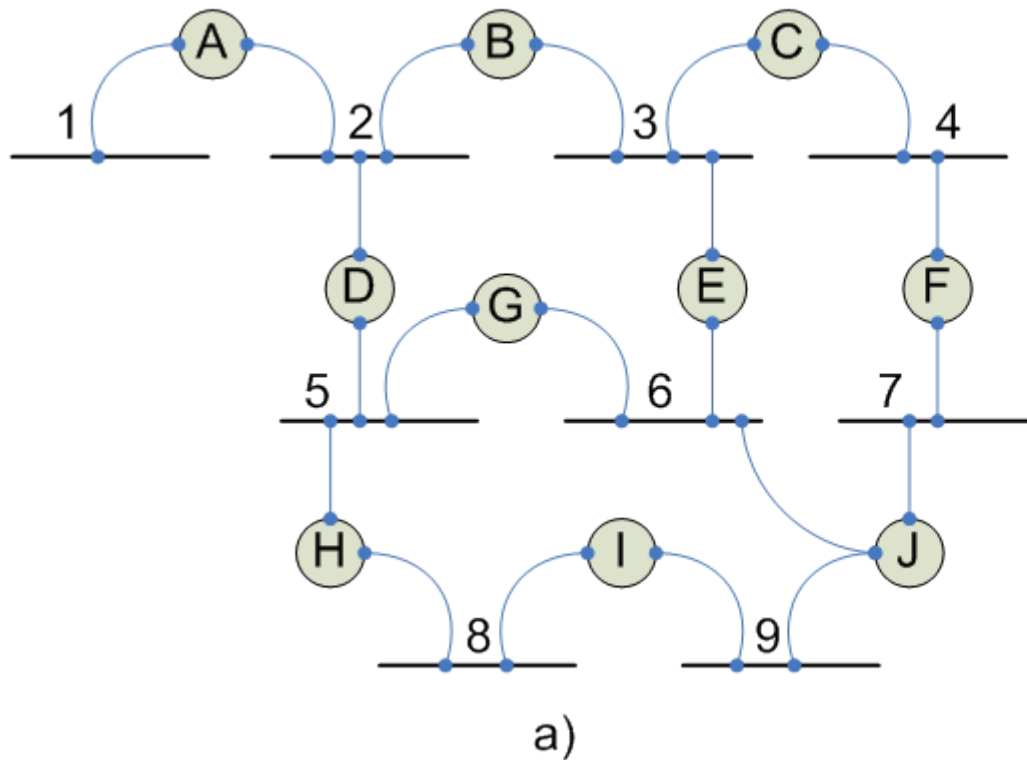
Creo que nunca veré
Un grafo más bonito que un árbol
Un árbol cuya propiedad crucial
Es estar libre de bucles.
Un árbol que debe estar seguro de abarcar
Para que los paquetes puedan alcanzar todas las LAN.
Primero, el *root* debe ser seleccionado.
Por ID, él es elegido.
Caminos de coste mínimo desde el *root* son trazados.
En el árbol, estos caminos se incluyen.
Una malla es hecha por gente como yo,
Entonces los puentes encuentran un árbol de
expansión.

—Radia Perlman

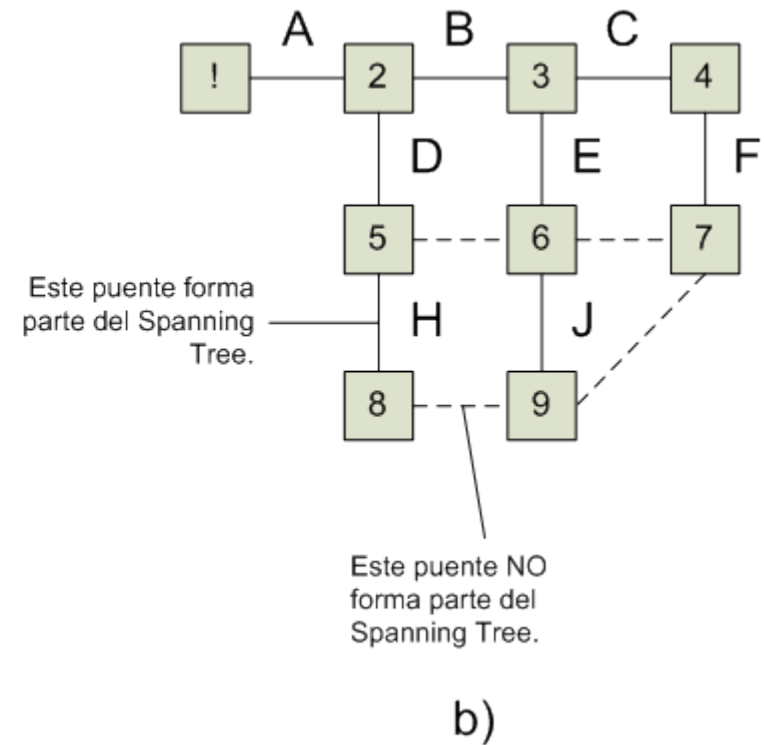
Basado en un poema de Joyce Kilmer http://en.wikipedia.org/wiki/Joyce_Kilmer



3.3 Spanning Tree (5)



(a) Conectividad física



(b) Conectividad lógica.

3.3 *Spanning Tree* (6) Bridge Protocol BPDUs



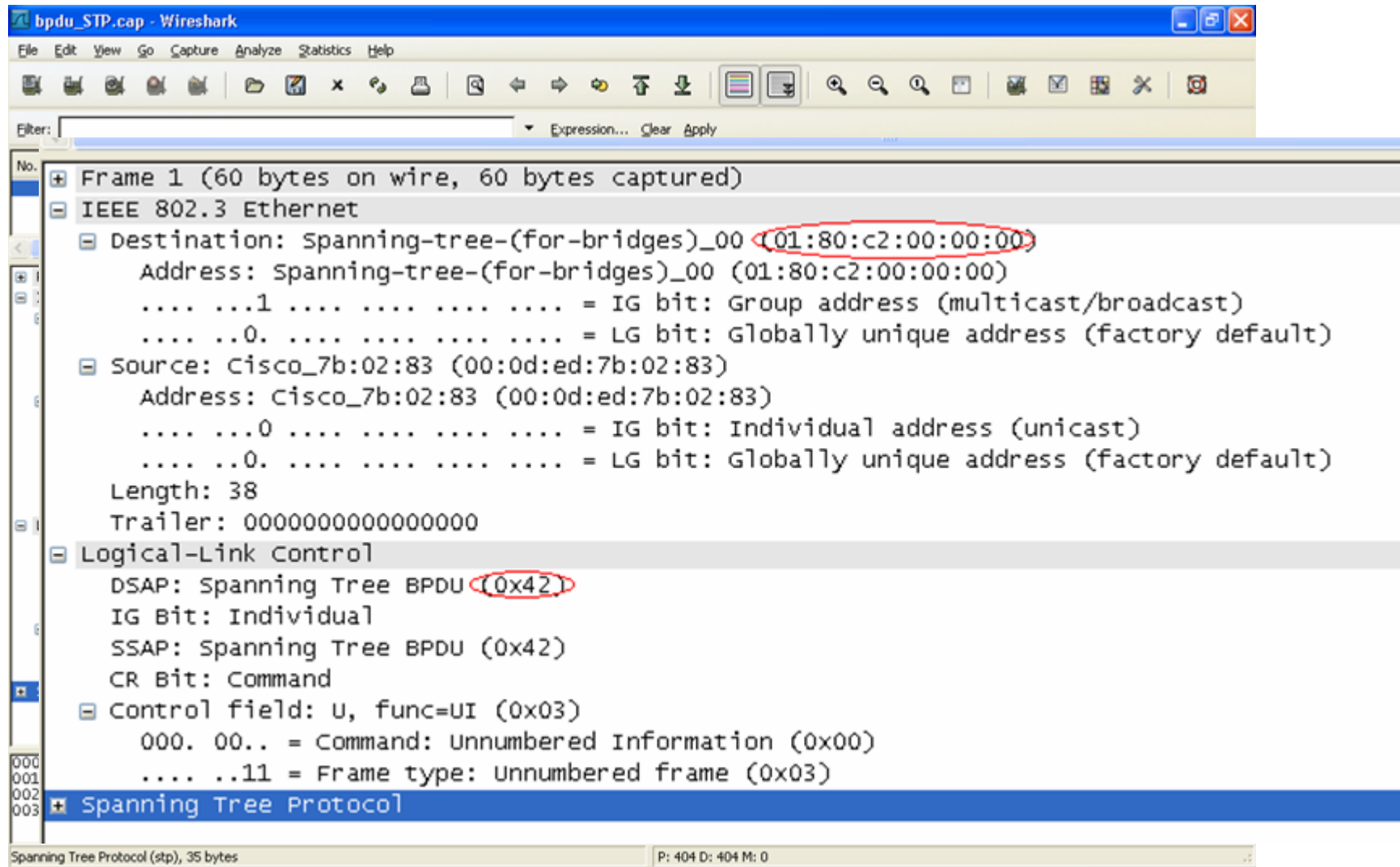
- Los puentes intercambian información sobre sus conexiones. La información se envía regularmente siguiendo un protocolo denominado **Bridge Protocol 802.1D**. Los mensajes se denominan BPDUs (*Bridge Protocol Data Units*).
- Las BPDUs emplean un **LLC** (son tramas 802.3) propio y se envían a una dirección **multicast** reservada, la 01-80-C2-00-00-00. Así se asegura que se identifican fácilmente y que llegan a toda la red.
- Cada puente recibe como **identificador (ID)** su dirección MAC ‘canónica’.
- Cada puerto recibe un identificador y tiene asociado un **costo** que suele ser inversamente proporcional a su velocidad (ej.: puerto 10BASE-T costo 100, puerto 100 BASE-T costo 10).
- Cada puente calcula el **grafo de la red** y observa si existe algún bucle; en ese caso se desactivan interfaces hasta cortar todos los bucles y construir un árbol sin bucles o ‘*spanning tree*’.
- La BPDUs transporta la terna “**root ID.cost to root.BridgeID**”. Inicialmente en la elección del raíz el root ID es el bridge ID.



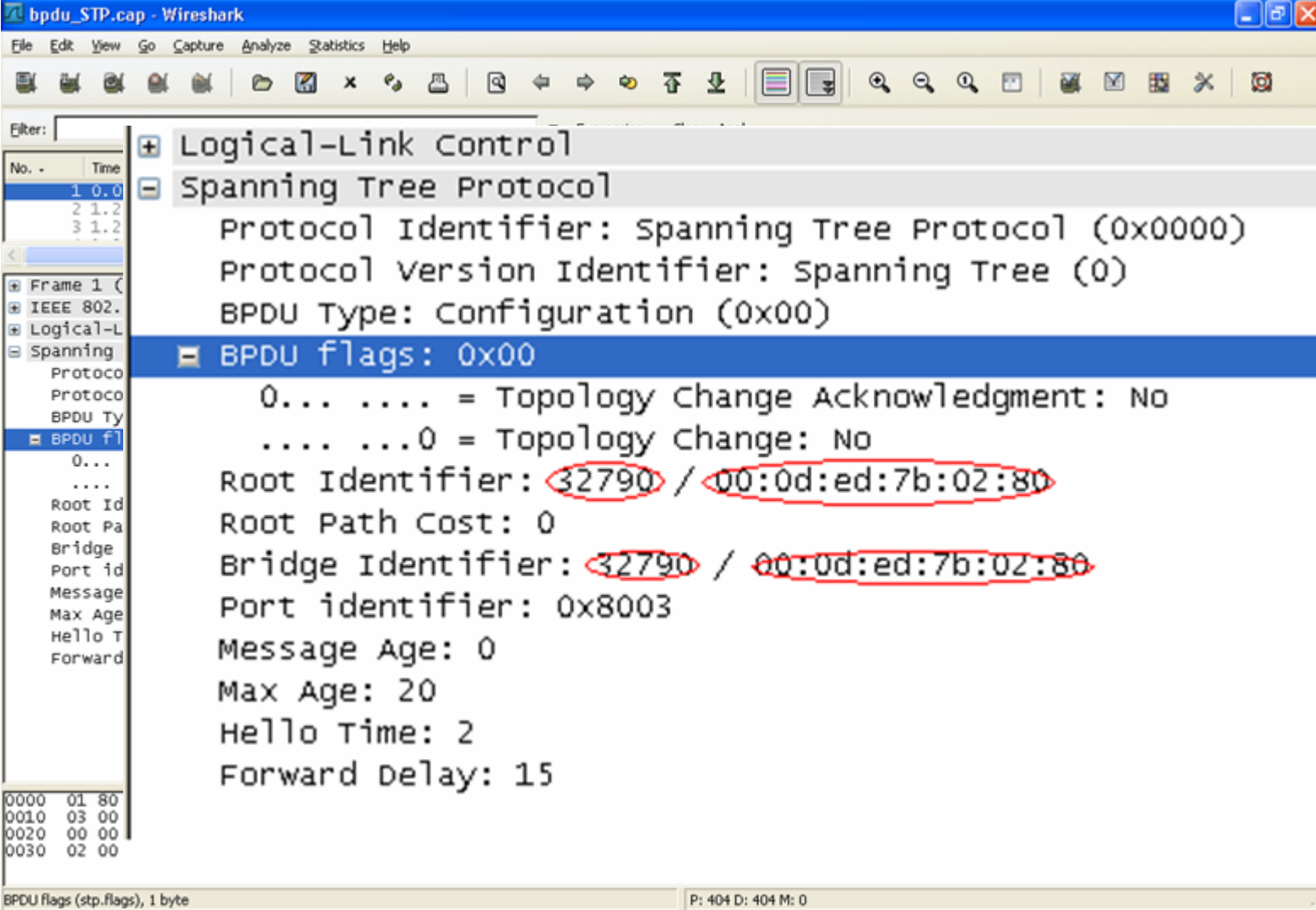
3.3 *Spanning Tree* (7) Bridge Protocol

- Cada puente envía por sus interfaces **BPDUs** indicando su ID, el ID de su puente raíz y el costo de llegar a él; los mensajes se van propagando por toda la red; cada puente al reenviar los mensajes les suma el costo de la interfaz por la que los emite.
- Los puentes eligen como **root** (raíz), a aquel que tiene el ID más bajo. Todos eligen al mismo. El ID se puede alterar con la prioridad del bridge, que se antepone a la MAC (0 a 65535, 32768 por defecto).
- Cada puente calcula por que puerto llega al root al mínimo costo. Ese es el **puerto raíz** (root port) de ese puente. En caso de empate se elige el puerto de ID más bajo.
- Para cada LAN hay un puerto **designado**, que es aquel por el que esa LAN accede al puente raíz al mínimo costo.
- En caso de que dos puertos empaten en coste, se desempata por un parámetro de **prioridad de puerto** (0 a 255, 128 por defecto).
- Los puertos que no son ni raíz ni designados se bloquean. Esos puertos son innecesarios para la comunicación y si se les deja funcionar provocan bucles.

3.3 Spanning Tree (8) Bridge Protocol



3.3 Spanning Tree (9) Bridge Protocol



bpdu_STP.cap - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter:

No. Time

1 0.0

2 1.2

3 1.2

Frame 1 (

IEEE 802.

Logical-L

Spanning

Protocol

Protocol

BPDU Ty

BPDU fl

0...

...

Root Id

Root Pa

Bridge

Port id

Message

Max Age

Hello T

Forward

0000 01 80

0010 03 00

0020 00 00

0030 02 00

Logical-Link Control

Spanning Tree Protocol

Protocol Identifier: Spanning Tree Protocol (0x0000)

Protocol Version Identifier: Spanning Tree (0)

BPDU Type: Configuration (0x00)

BPDU flags: 0x00

0... = Topology Change Acknowledgment: No

.... ...0 = Topology Change: No

Root Identifier: 32790 / 00:0d:ed:7b:02:80

Root Path Cost: 0

Bridge Identifier: 32790 / 00:0d:ed:7b:02:80

Port identifier: 0x8003

Message Age: 0

Max Age: 20

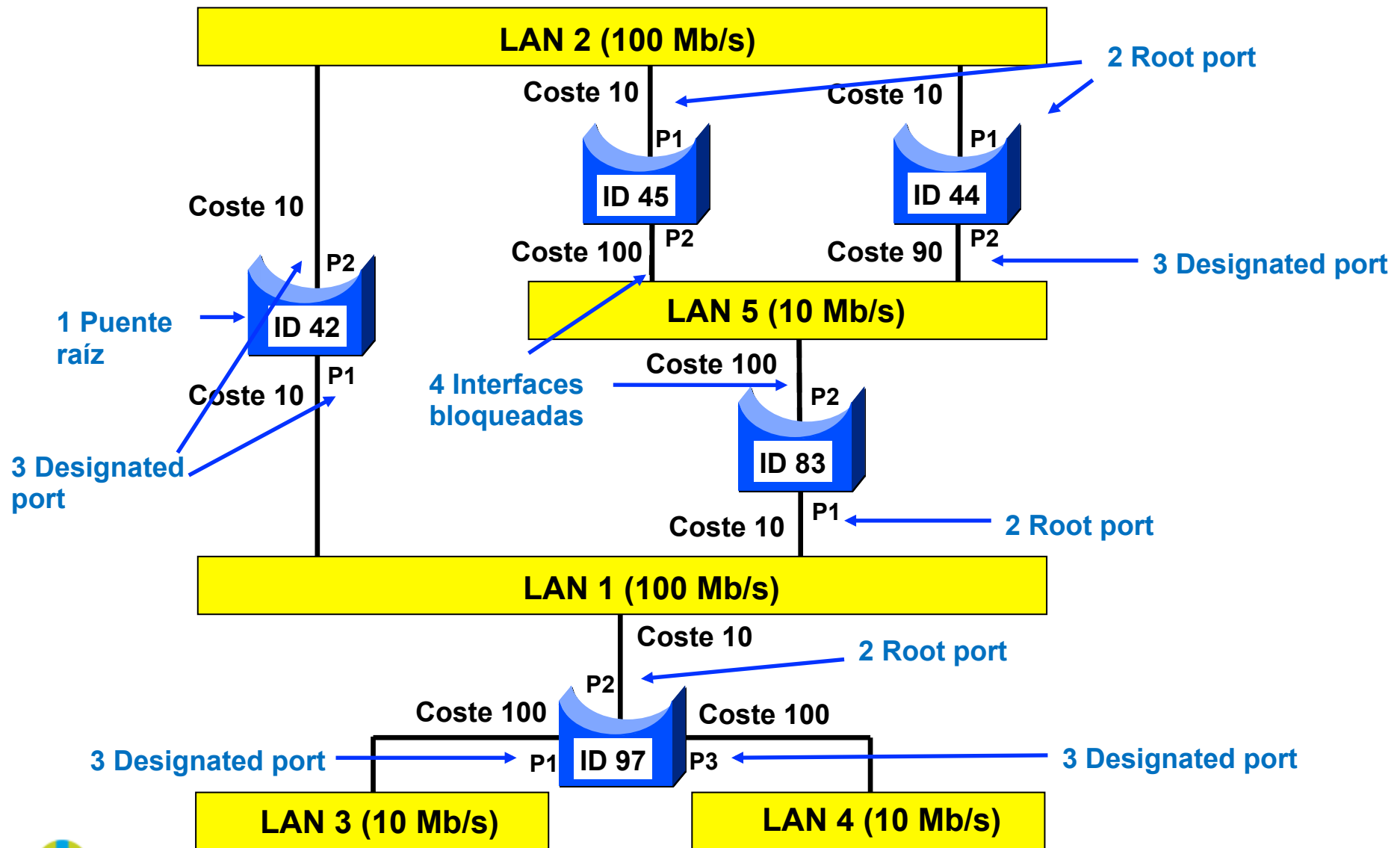
Hello Time: 2

Forward Delay: 15

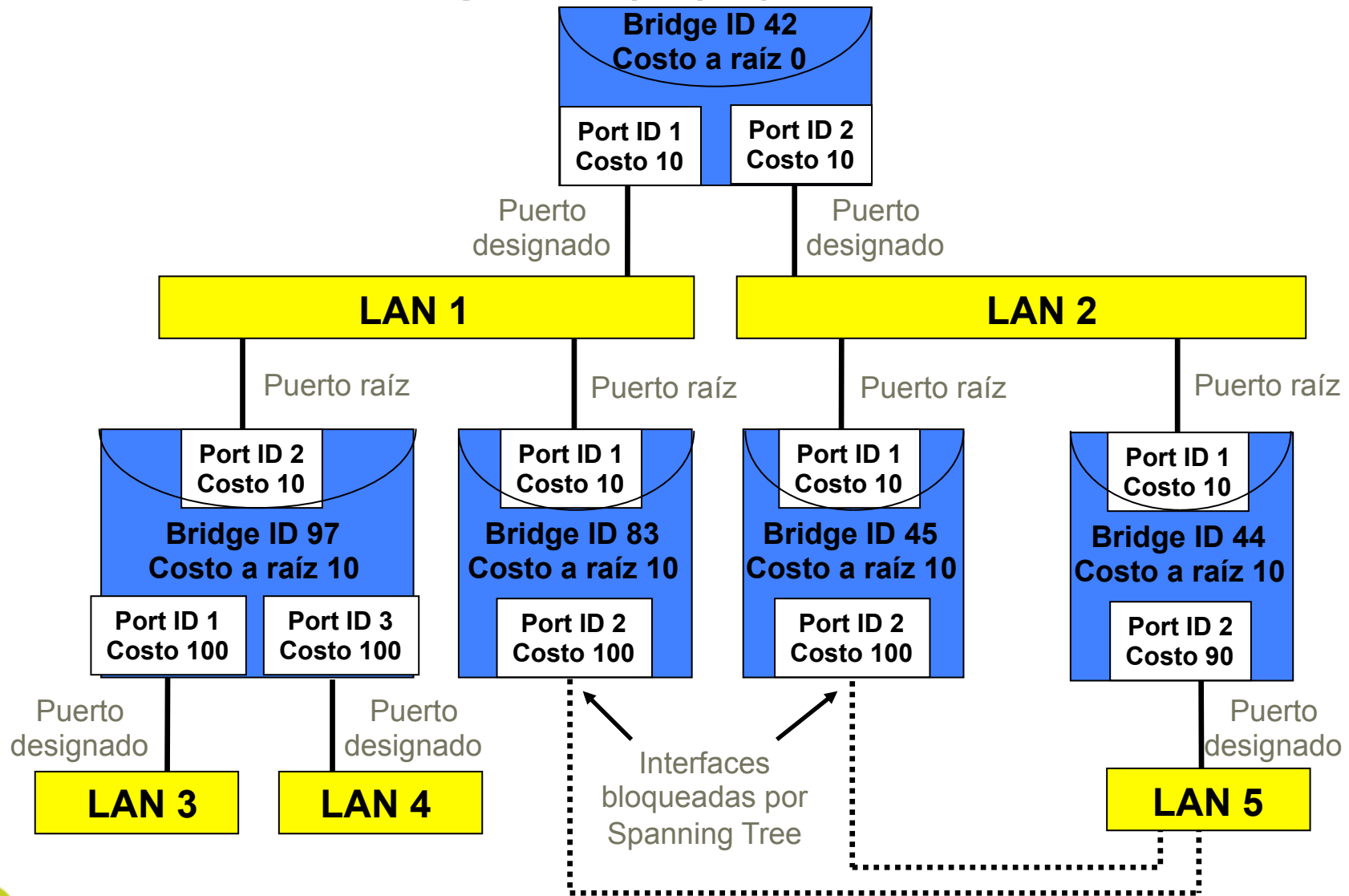
BPDU flags (stp.flags), 1 byte

P: 404 D: 404 M: 0

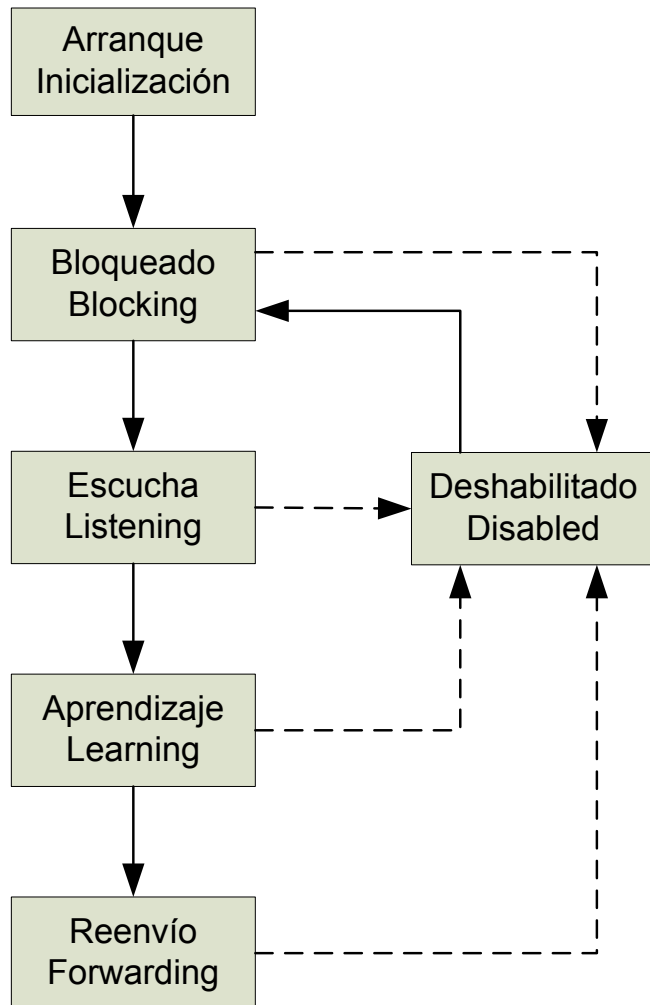
3.3 Spanning Tree (11) Ejemplo



3.3 Spanning Tree (12) Ejemplo



3.3 Spanning Tree (13) Estados de los puertos



- Cuando un puerto de un puente se conecta se pone inicialmente en estado '**blocking**'. En este estado no reencamina tramas; solo capta y procesa las BPDUs que le llegan.
- Si no detecta bucle el puerto pasa al estado '**listening**'. Ahora además de procesar las BPDUs recibidas genera y envía las suyas propias con su información topológica.
- Si todo va bien pasa a estado '**learning**'. Ahora además de procesar y generar BPDUs aprende las direcciones MAC de origen que hay en las tramas que le llegan, pero no reenvía nada (salvo BPDUs).
- Si todo sigue bien pasa a estado '**forwarding**' en el que además reenvía las tramas (siguiendo el algoritmo de los puentes transparentes).
- Este procedimiento evita bloquear la red de entrada si existe algún bucle.
- Cada vez que se conecta una interfaz se realiza el proceso desde el principio, por eso cuando se conecta una interfaz de un puente tarda unos 10-20 segundos en empezar a funcionar.

3.3 *Spanning Tree* (14) Conclusión

- El protocolo *Spanning Tree* permite crear topologías redundantes, que mejora la fiabilidad.
- Forma parte de la norma IEEE 802.1d.
- El tiempo necesario ante un cambio, depende del tamaño de la red (diámetro). Típicamente 45s. El máximo diámetro soportado es de 7 saltos (no es un mito).
- Se ha mejorado con los protocolos 802.1w *Rapid Spanning Tree*, convergencia en segundos, y 802.1s, múltiples instancias de *Spanning Tree* para entornos con VLAN.

3.3 Puentes de encaminamiento basado en la fuente (*Source Routing*)



- Usada en *Token Ring*, aunque existe versión para Ethernet.
- Se numeran las LAN y los puentes.
- Las estaciones, envían tramas exploradoras, ARE (*All Router Explorer*), que se copian a todas las redes.
- Los puentes van añadiendo información de camino.
- La estación escoge la respuesta más rápida, utilizando en adelante la información de ruta.
- Existen optimizaciones en conmutadores.



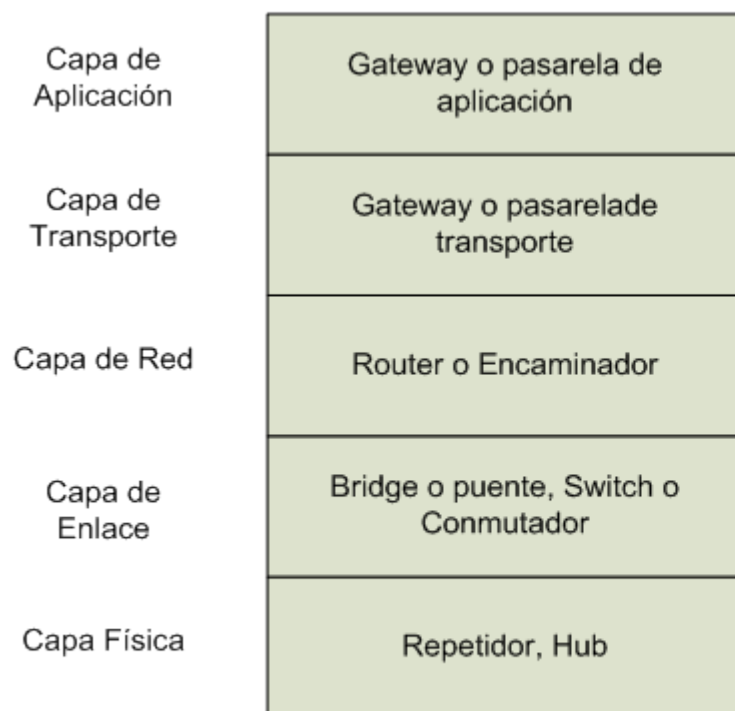
3 Subcapa de acceso al medio

3.3 Conmutación de nivel 2 (3/3)



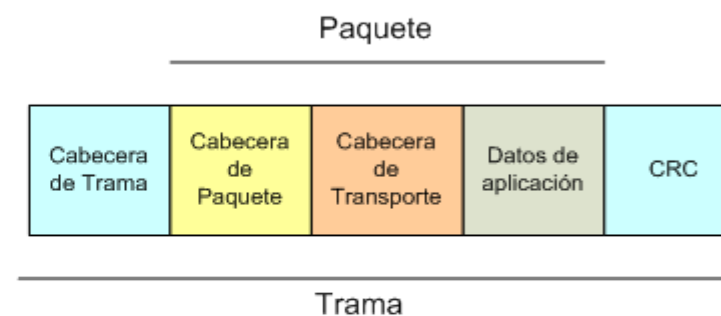


3.3 Elementos de red (1)



(a)

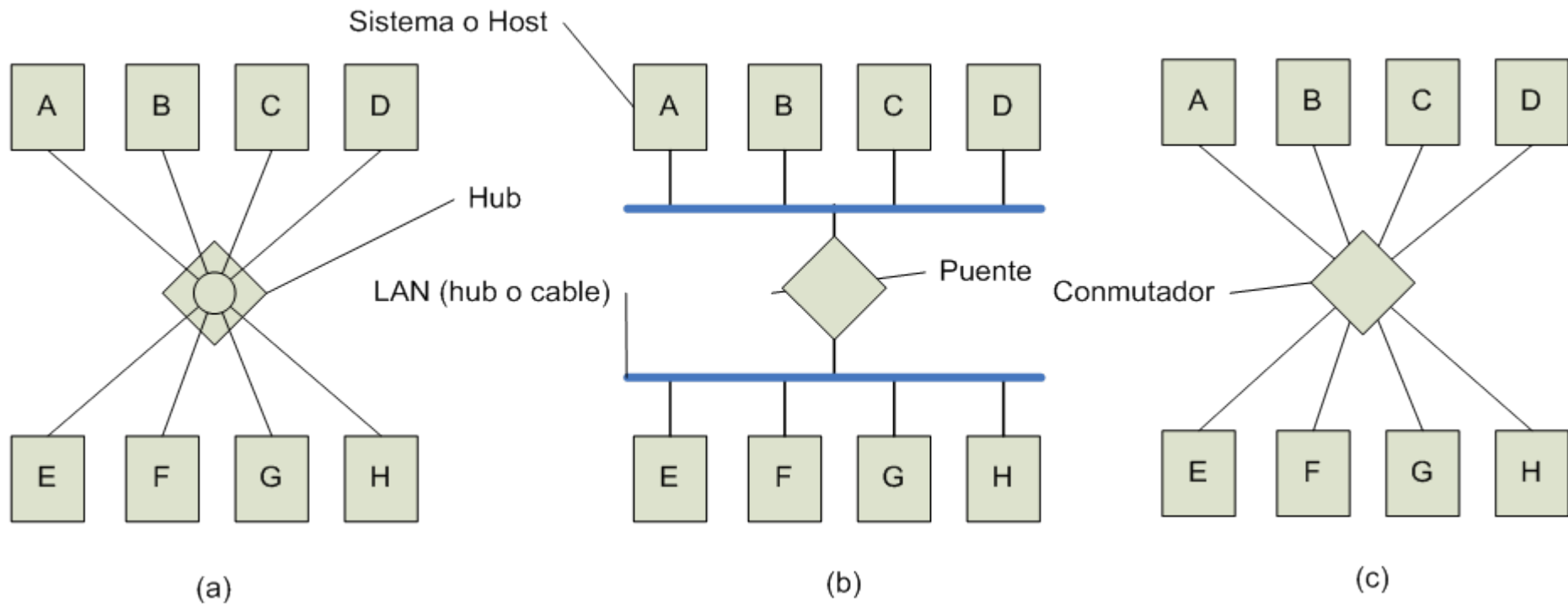
(a) Dispositivos y Arquitectura de protocolos



(b)

(b) Estructura del paquete

3.3 Elementos de red (2)



(a) Hub L1

(b) Bridge L2

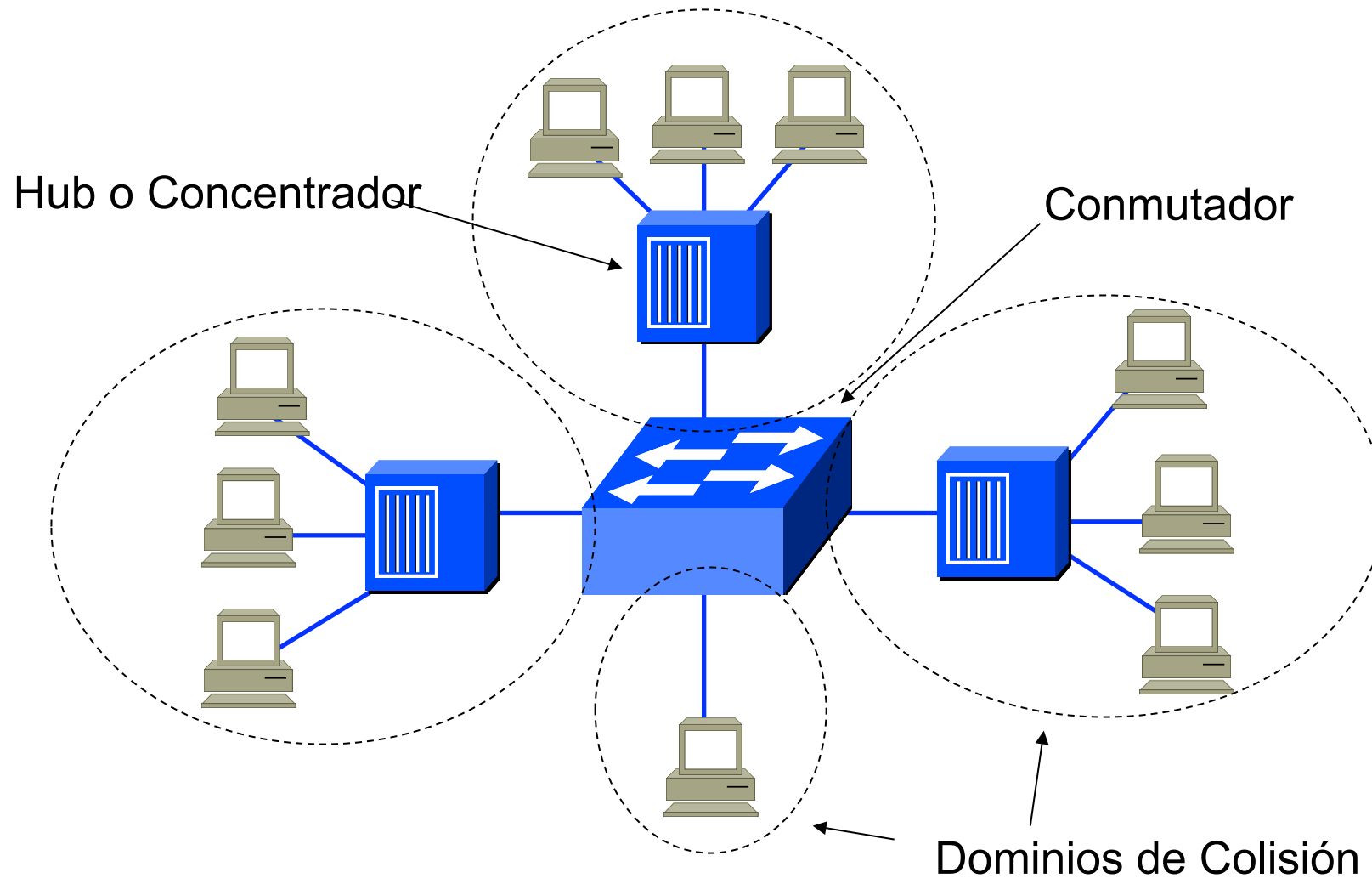
(c) Switch L2



3.3 Conmutadores LAN

- Son puentes en los que se ha implementado el algoritmo de reenvío de tramas en hardware, en circuitos ASIC (*Application Specific Integrated Circuit*).
- Tienen un rendimiento muchísimo mayor que los puentes que realizan el algoritmo por software. Pueden funcionar a ‘*wire speed*’, es decir a la velocidad nominal de la interfaz.
- Tienen muchas interfaces, normalmente 12 o más. Hay modelos que pueden llegar a tener más de 1200 puertos Gigabit Ethernet.

3.3 Conmutadores LAN. Dominio de colisión.





3.3 Conmutadores LAN: Formas de conmutación de tramas

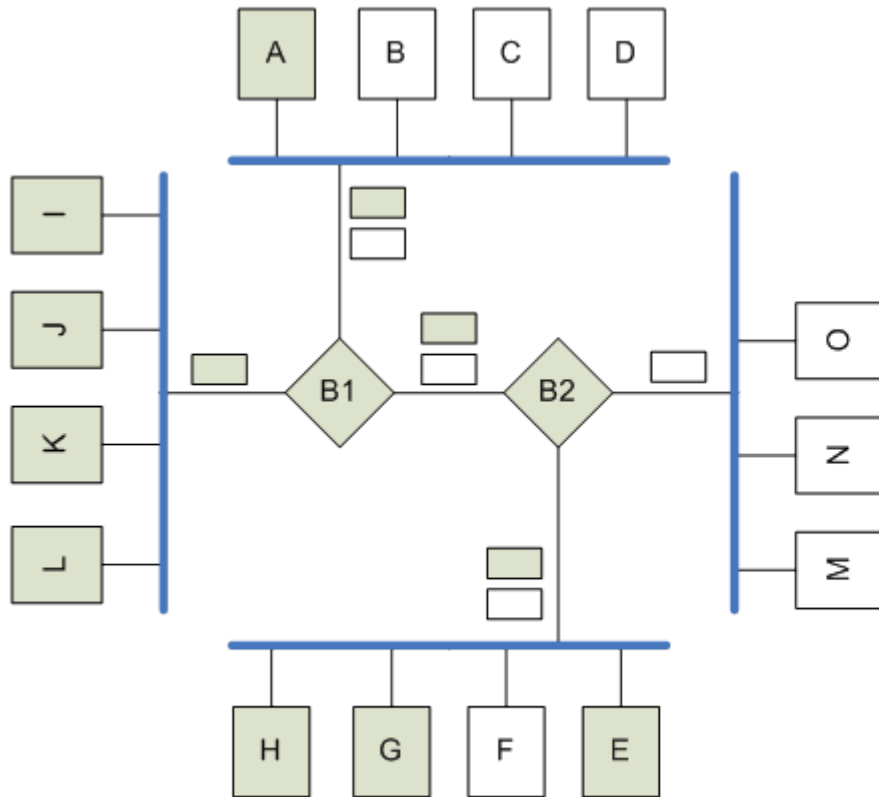
1. **Store and Forward (Almacenamiento y reenvío).** El conmutador recibe la trama en su totalidad, comprueba el CRC y la retransmite si es correcta (si no la descarta).
2. **Cut-through.** El conmutador empieza retransmitir la trama tan pronto ha leído la dirección de destino (6 primeros bytes). Aunque el CRC sea erróneo la trama se retransmite. Menor latencia que almacenamiento/reenvío.
3. **Cut-through libre de fragmentos.** Es igual que *Cut-through* pero en vez de empezar enseguida espera a haber recibido 64 bytes. Así se asegura que no es un fragmento de colisión.
4. **Híbrido.** Usa *cut-through* inicialmente, pero si detecta que una estación genera tramas erróneas pasa a modo *store and forward* para las tramas que vienen de esa dirección MAC.



3.3 VLAN

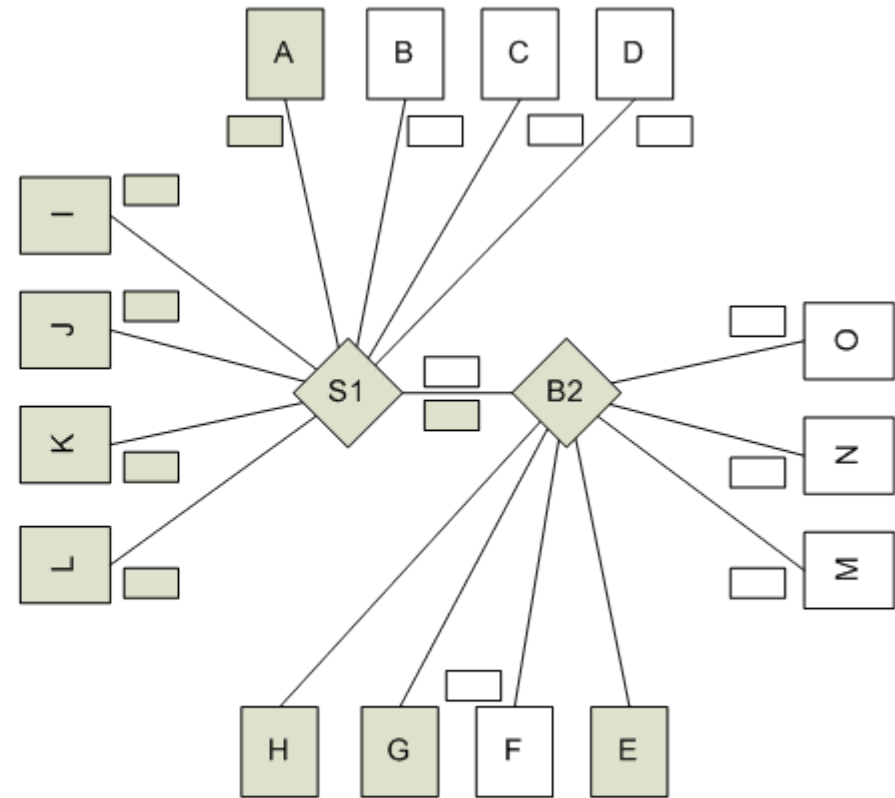
- Las VLAN, *Virtual LAN*, equivalen a dividir un conmutador en varios más pequeños. No es un protocolo. Es propio de cada conmutador.
- La interconexión de conmutadores con VLAN se define en la norma 802.1q.
- Objetivos:
 - ☐ Rendimiento (reducir tráfico *broadcast*)
 - ☐ Gestión. Organización de los puertos.
 - ☐ Seguridad. Separación en varias redes.
- Normalmente la interconexión entre VLANs se hace con un *router*.

3.3 VLAN



(a)

(a) *Bridges* o puentes.

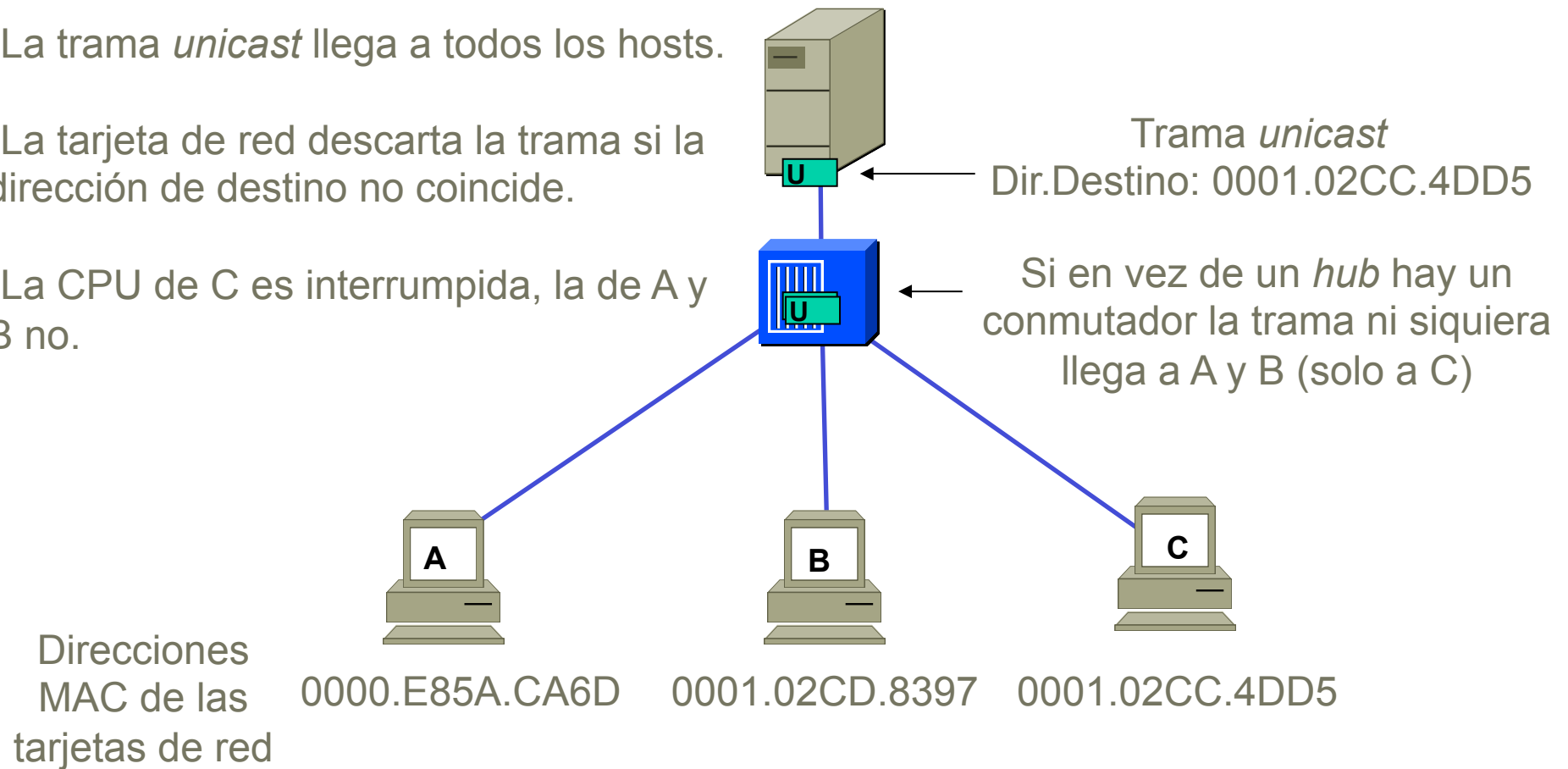


(b)

(b) *Switches* o conmutadores.

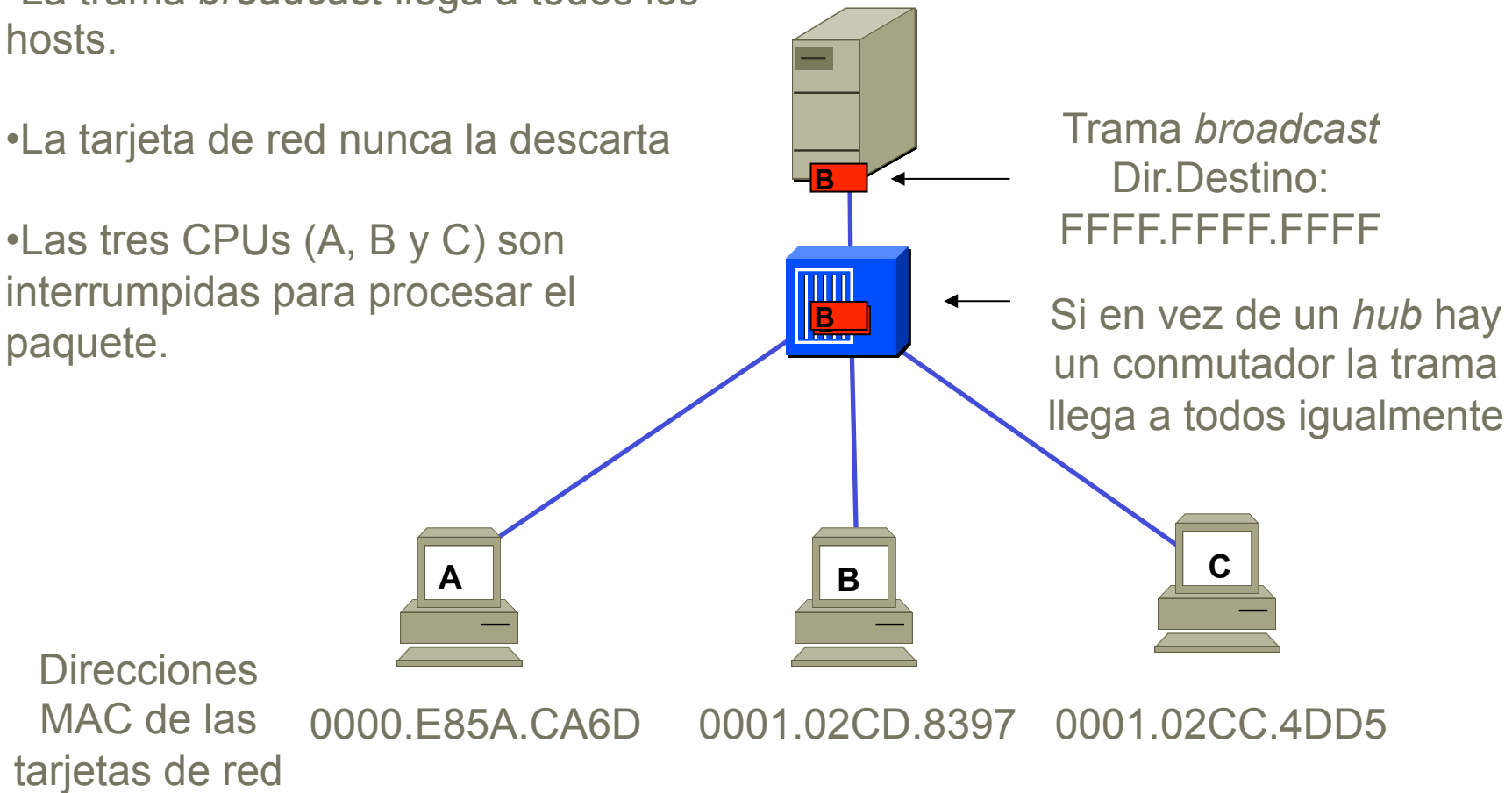
3.3 VLAN Envío de tramas Unicast

- La trama *unicast* llega a todos los hosts.
- La tarjeta de red descarta la trama si la dirección de destino no coincide.
- La CPU de C es interrumpida, la de A y B no.

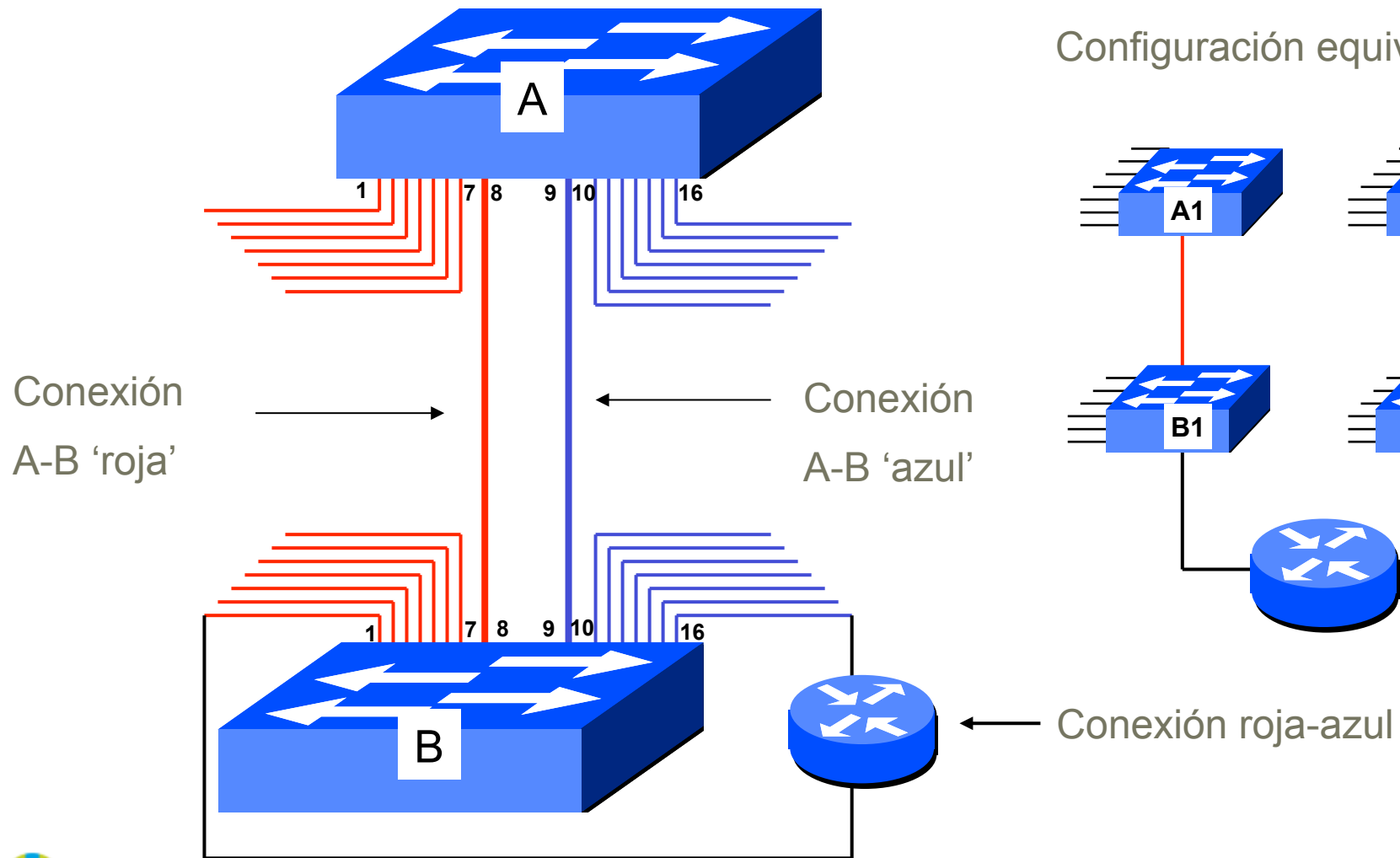


3.3 VLAN Envío de tramas Broadcast

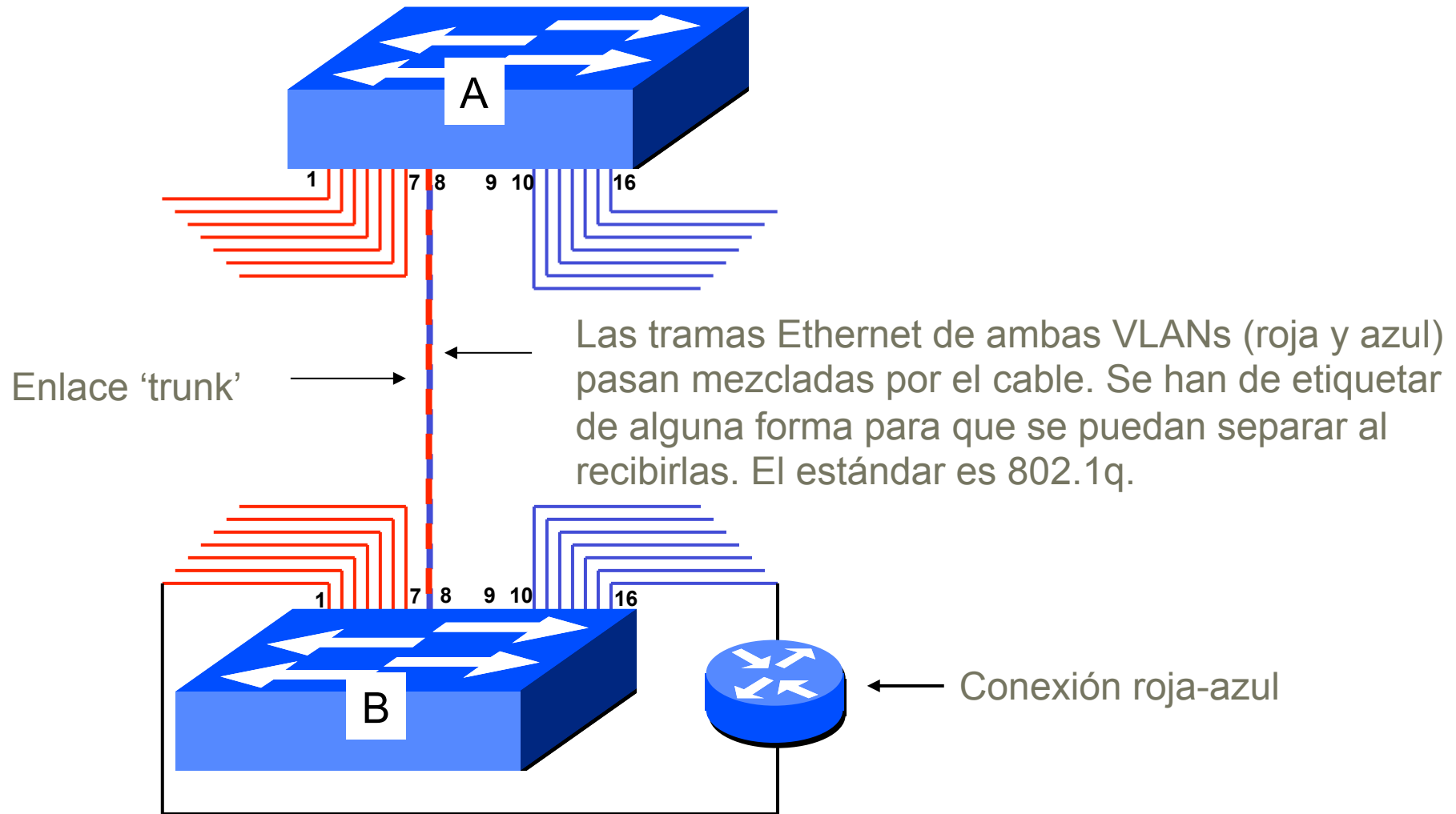
- La trama *broadcast* llega a todos los hosts.
- La tarjeta de red nunca la descarta
- Las tres CPUs (A, B y C) son interrumpidas para procesar el paquete.



3.3 VLAN Ejemplo



3.3 VLAN 802.1Q Ejemplo

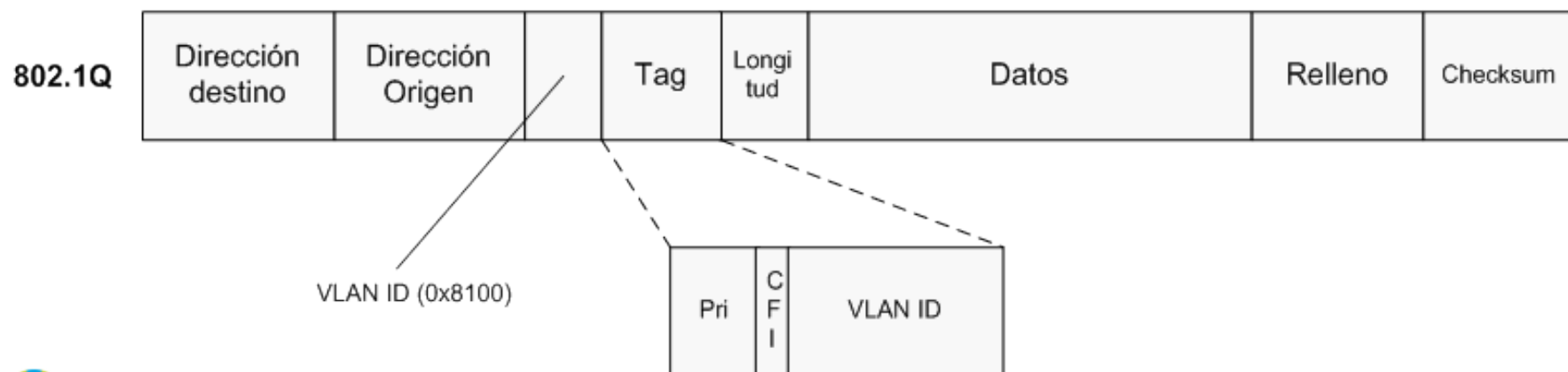




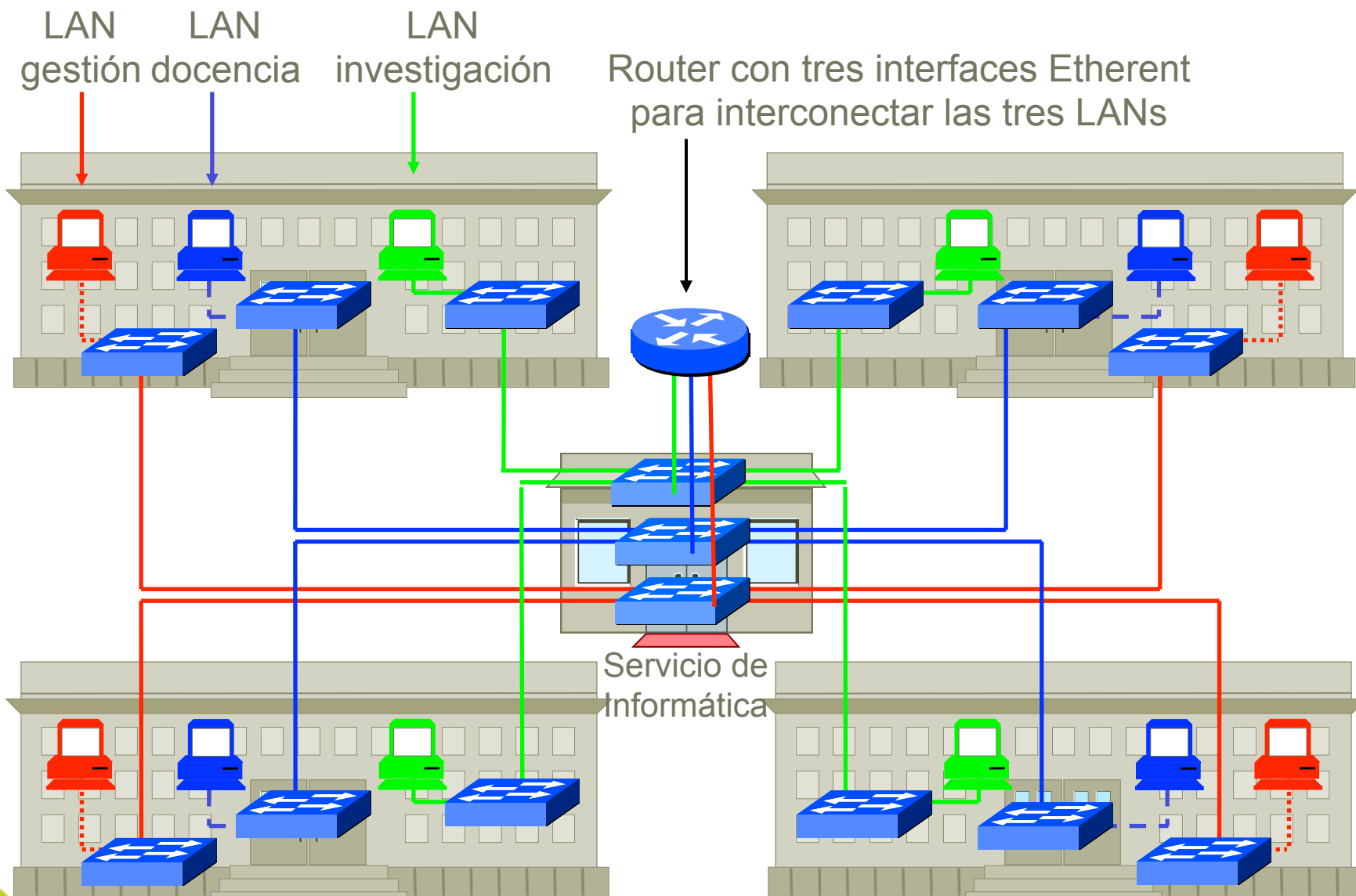
3.3 VLAN 802.1Q

A la trama 802.3 se le añade un nuevo *ethertype* 0x8100 y un tag:

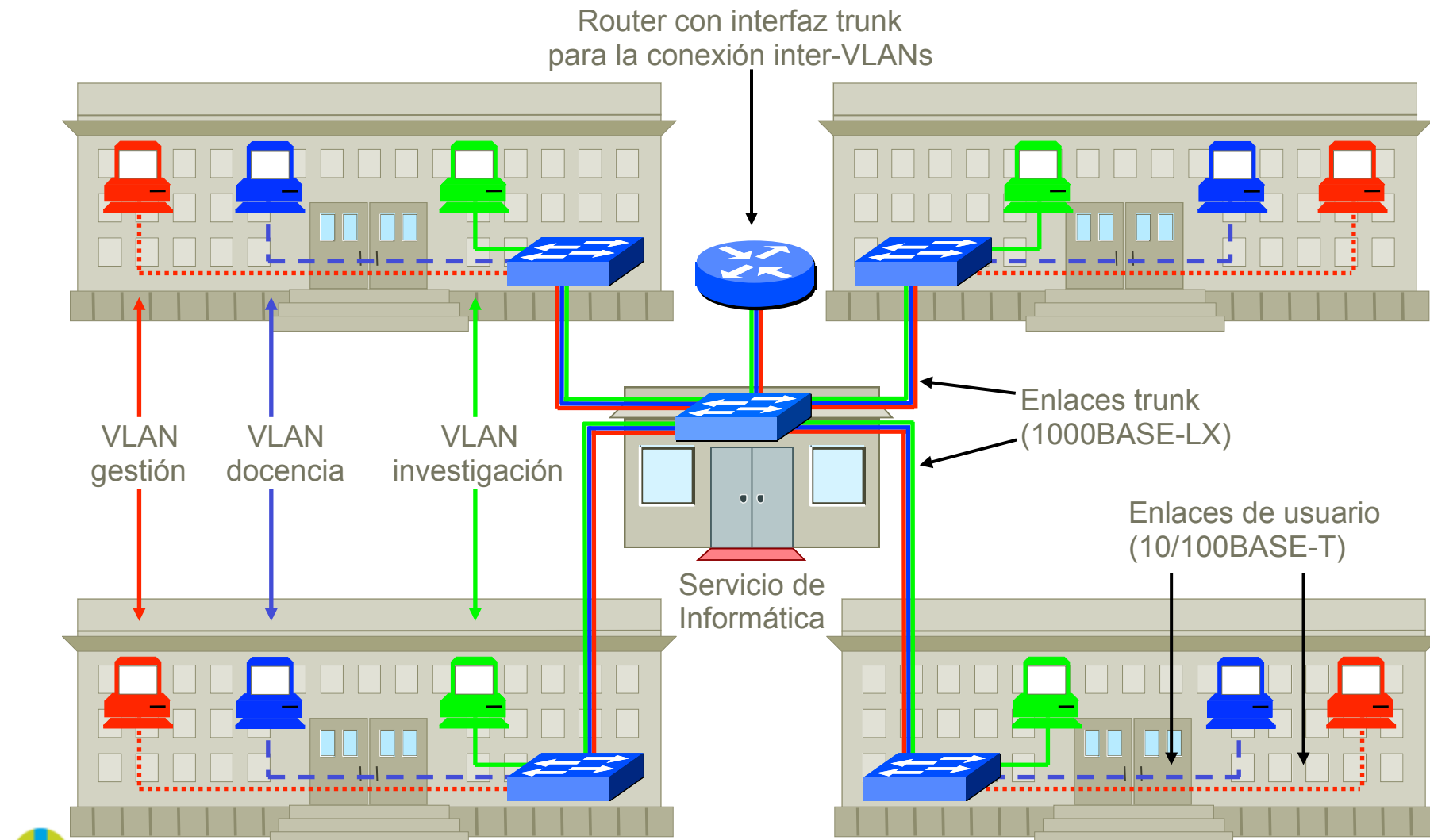
- **Pri o COS:** Prioridad o Class of Service, 3 bits.
- **CFI:** Canonical Format Indicator, 1 bit.
- **VLAN ID.** Identificador de VLAN, 12 bit.



3.3 VLAN Ejemplo de red Campus sin VLAN



3.3 VLAN Ejemplo de red Campus con VLAN



3.3 Bibliografía

[1] Tanenbaum, A. S., Computer Networks, 4ª Ed Pearson 2003, apartado 4.7.

[2] Stallings, W., Comunicaciones y Redes de Computadores, 6ª Ed Prentice Hall 2000, apartado 13.7.

[3] Perlman, R., Interconnections: Bridges, Routers, Switches and Internetworking Protocols, 2ªEd Addison Wesley, capítulos 3, 4 y 5.

3.3 *Spanning Tree* (10) Bridge Protocol



spanning_tree1.swf