

## TEMA 1. REDES DE TELECOMUNICACIÓN

- **CONCEPTOS BÁSICOS**
- **ARQUITECTURAS DE PROTOCOLOS**
  - ✓ MODELO DE REFERENCIA OSI
- **CRONOGRAMAS**

## TEMA 2. REDES DE CONMUTACIÓN DE PAQUETES

- **REDES DE AREA LOCAL (LAN)**
  - ✓ MECANISMOS DE ACCESO AL MEDIO
    - ALOHA, ALOHA RANURADO, CSMA
  - ✓ EJEMPLOS DE REDES DE AREA LOCAL
    - ETHERNET (802.X), WIRELESS LAN (802.11)
- **REDES DE AREA EXTENDIDA (WAN)**
  - ✓ X.25
  - ✓ FRAME RELAY
  - ✓ TCP/IP
    - PROTOCOLOS IP, ICMP, TCP, UDP
    - DIRECCIONAMIENTO
- **CAPA DE ENLACE. CONTROL DE FLUJO**
  - ✓ FEC (FORWARD ERROR CORRECTION)
  - ✓ ARQ (AUTOMATIC REPEAT REQUEST)
    - STOP & WAIT
    - GO-BACK-N
    - ARQ-SELECTIVO

## TEMA 3. CONMUTACIÓN DE CIRCUITOS

- **DIMENSIONADO**
  - ✓ CONCEPTOS DE TRÁFICO
  - ✓ CADENAS DE MARKOV
  - ✓ SISTEMAS ER<sub>B</sub> Y ER<sub>C</sub>
- **MULTIPLEXACIÓN. JERARQUÍAS DIGITALES**

## TEMA 1. REDES DE TELECOMUNICACIÓN

### CONCEPTOS BÁSICOS

Red	conjunto de elementos que permiten el transporte de información
arquitectura de red	conjunto de elementos que forman la red y funciones que estos elementos realizan
terminal	elemento origen/destino de la comunicación
enlace	elemento de transmisión
Canal	fracción de la capacidad del enlace
Nodo	elemento de interconexión con unas determinadas funcionalidades

**TOPOLOGÍA** Concepto abstracto; esquema de interconexión de los terminales en una red.

**CLASIFICACIÓN** según el tamaño de la red:

- LAN (Local Area Network)
- MAN (Metropolitan Area Network)
- WAN (Wide Area Network)

**CLASIFICACIÓN** según la redundancia de la red:

- **REDES SIN REDUNDANCIA** Solo existe una posible ruta entre los terminales.  
BUS, ARBOL JERÁRQUICO, ESTRELLA,
- **REDES CON REDUNDANCIA** Existe más de una ruta entre terminales  
TOKEN RING, TOTALMENTE MALLADA, PARCIALMENTE MALLADA

**FLUJO DE DATOS** Este término se utiliza para expresar el sentido en el que viaja la información entre el origen y el destino. Puede ser:

- **UNIDIRECCIONAL (o SIMPLEX):** la info viaja en un solo sentido
- **BIDIRECCIONAL (o DUPLEX):** la info viaja en ambos sentidos. Distinguimos entre:
  - ✓ Half-dúplex: la info no viaja simultáneamente
  - ✓ Full-duplex: la info viaja simultáneamente

**CONGESTIÓN** Es un estado en el que entra la red cuando el tráfico que se le ofrece es demasiado elevado. Esta sobrecarga en la red provoca la pérdida y/o retardo de los datos transmitidos.

**CONTROL DE FLUJO** Existen mecanismos encargados de parar o limitar la velocidad a la que los terminales transmiten la información (ver Tema 3-Nivel de Enlace).

**SEÑALIZACIÓN** Es el intercambio de información entre las diversas partes de la red con el objetivo de establecer, supervisar y administrar las conexiones. Se puede ver como toda la información que intercambian las diversas partes de la red y que no son datos de usuario.

**ENCAMINAMIENTO / ENRUTAMIENTO** Hace referencia al camino que seguirá la información desde el terminal origen al terminal destino. Los mecanismos de decisión de enrutamiento dependen en gran medida del tipo de información que se transmite. El elemento que realiza esta función es un encaminador (router). Para enrutar los datos, los routers disponen de unas tablas (tablas de enrutamiento) en las que se especifica el enlace de salida por el que transmitir los datos. Los mecanismos de encaminamiento se pueden clasificar en : **FIJO, ALTERNATIVO, DINÁMICO.**

**DIRECCIONAMIENTO** Hace referencia al número de destinatarios de la información. Los tres modos de direccionamiento más conocidos son : **UNICAST, MULTICAST, BROADCAST.**

**CONMUTACIÓN** Se define como el proceso para establecer una conexión individual, solicitada por medio de un proceso de señalización, desde un punto de entrada hacia un punto de salida, dentro de un conjunto de entradas y salidas.

**MULTIPLEXACIÓN** Técnica que permite transmitir señales de distintos usuarios por un mismo enlace. La multiplexación es una de las formas de acceder a un medio compartido. Las principales técnicas de multiplexación son: ~~FDMA (Frequency Division Multiple Access)~~, ~~TDMA (Time Division Multiple Access)~~ y ~~CDMA (Code Division Multiple Access)~~.

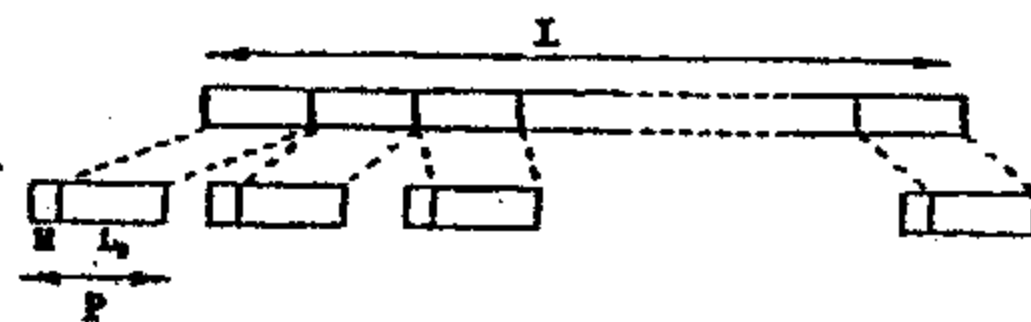
**RED TELEFÓNICA** Tiene una estructura jerárquica parcialmente mallada

**SEÑALIZACIÓN EN LA RED TELEFÓNICA** La señalización en la red telefónica se acostumbra a dividir en los siguientes tramos:

- tramo entre el abonado y la central (señalización de abonado). En este tipo de señalización se aprovecha al máximo las posibilidades ofrecidas por las variaciones de nivel de corriente y los tonos audibles.
- tramo interno de una central (señalización interna).
- tramo entre centrales (señalización entre centrales). Las centrales pueden intercambiar los mensajes de señalización de dos maneras:
  - .por **CANAL ASOCIADO** se utiliza el propio canal.
  - .por **CANAL COMUN** se envían por un único canal de señalización las señales correspondientes a varios canales de conversación.

~~**CONMUTACIÓN DE CIRCUITOS** La idea básica es que antes de empezar a transmitir se establece una ruta desde el origen al destino, reservando los recursos necesarios para garantizar una correcta transmisión (sin retardo ni pérdidas). Toda la información es transmitida por el mismo camino. Todos los nodos transmiten a la misma velocidad. Existen tres fases distintas: ESTABLECIMIENTO, TRANSMISIÓN, SEPARACIÓN. Utiliza multiplexación determinista. El tiempo de proceso en los nodos es cero.~~

~~**CONMUTACIÓN DE PAQUETES** La información transmitida se agrupa en trozos más pequeños (paquetes). Utiliza multiplexación estadística.~~



Existen dos grandes categorías:

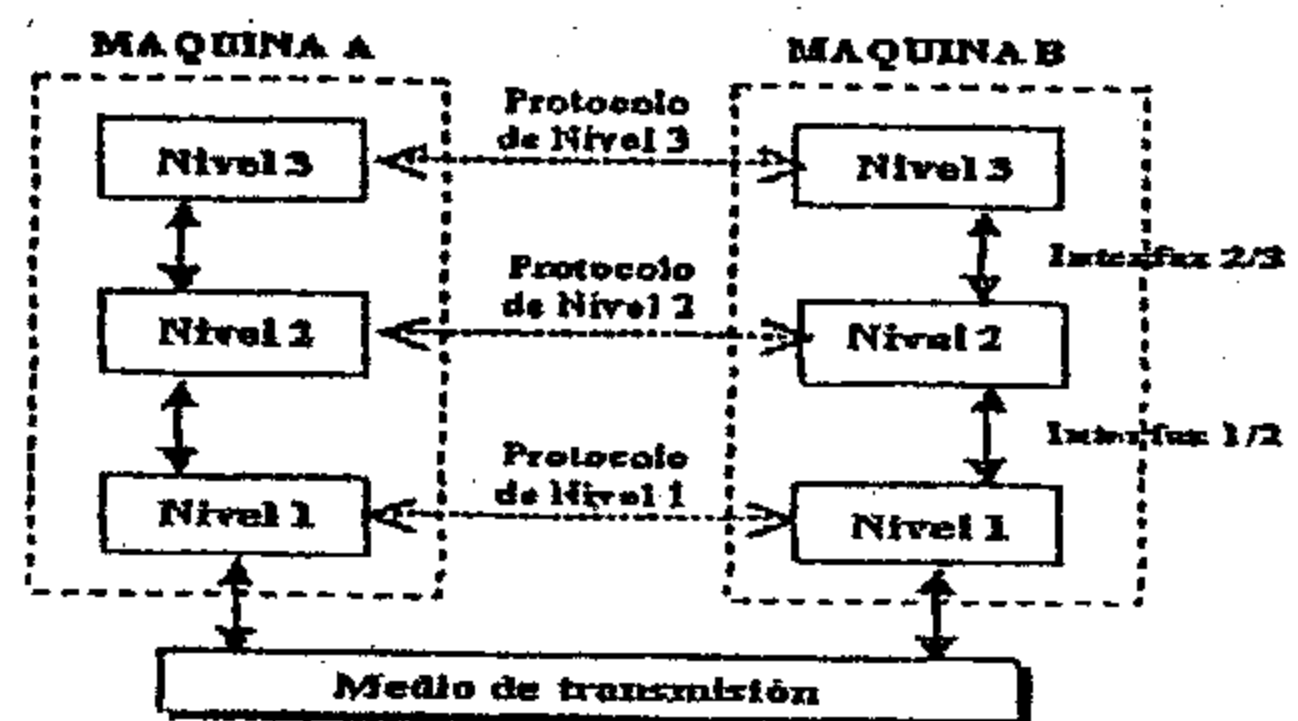
~~**MODO CIRCUITO VIRTUAL** Existen las tres fases de establecimiento, transmisión y liberación. Todos los paquetes se envían por el mismo camino. Cada paquete lleva un identificador de circuito virtual (id\_cv) en su cabecera. El tiempo de proceso requerido en cada nodo es bastante pequeño.~~

~~**MODO DATAGRAMA** Cada paquete se transmite independientemente de los otros. El hecho de tener que enviar cada paquete provoca que al tiempo de proceso en los nodos sea grande. Los paquetes pueden llegar desordenados al destino (fuera de secuencia). Los mecanismos de control de errores son bastante más complejos en modo datagrama que en modo circuito virtual.~~

**PROTOCOLO** Es el conjunto de reglas que conocen los elementos que intercambian datos a través de una red de comunicaciones.

**JERARQUÍA DE PROTOCOLOS** La mayoría de las redes se organizan en niveles (capas), contruidos cada uno sobre el inmediatamente inferior. Una jerarquía se construye en base a:

- Cada nivel ofrece servicios al nivel superior, ocultándole la forma en que realmente se implementan dichos servicios.



- Cada nivel dentro de una máquina conversa con su gemelo en otra.
- No hay transferencia física de datos entre niveles gemelos
- La transferencia se hace en cada máquina entre un nivel, su superior y su inferior a través de los interfaces entre niveles

## MODELO DE REFERENCIA OSI

### NIVEL FÍSICO

Se ocupa de enviar y recibir bits sobre un medio físico de transmisión:

- Debe asegurarse que un bit a 1 enviado es recibido como un bit a 1.
- La transmisión puede ser eléctrica, óptica, ...
- Aborda los problemas eléctricos, electrónicos, de establecimiento de conexiones físicas, ...

### NIVEL DE ENLACE

Transforma un sistema de transmisión crudo (lo que le ofrece el nivel físico) en una línea libre de errores de transmisión (lo que ofrece para el nivel de red) entre máquinas conectadas al mismo medio físico.

- Un emisor trocea los datos del nivel de red y envía tramas de datos secuencialmente, recibiendo tramas de asentimiento del receptor.
- Crea y reconoce separadores de tramas.
- Retransmite tramas perdidas y elimina tramas duplicadas.
- Controla el flujo de la transmisión.
- Se ocupa de la transmisión bidireccional, si la hubiere.

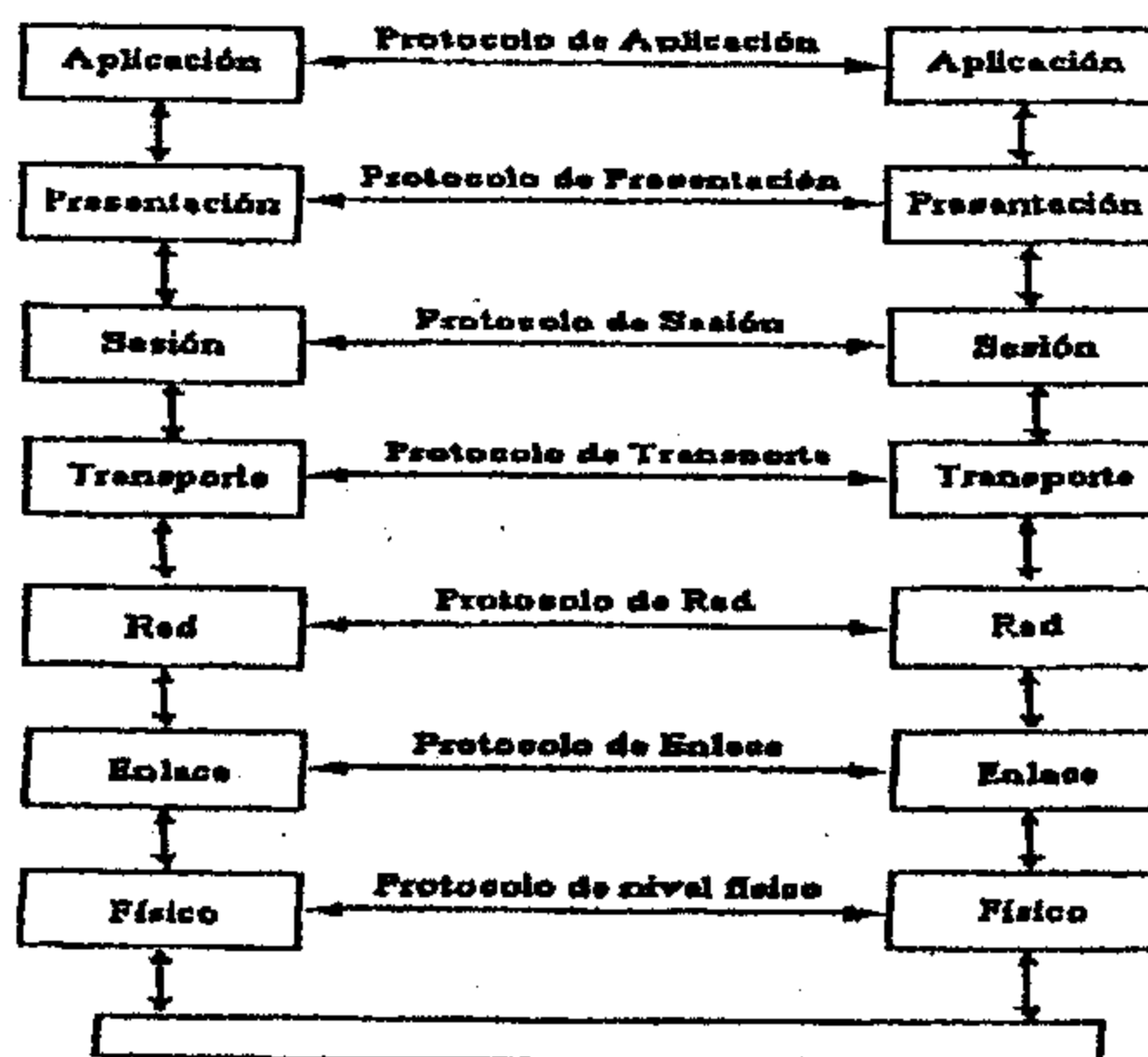
### NIVEL DE RED

Controla el funcionamiento de la subred, para interconectar máquinas que no están en el mismo medio físico:

- Se ocupa del encaminamiento de paquetes desde el origen al destino.
- El encaminamiento puede ser estático o dinámico.
- Gestiona las congestiones y cuellos de botella.
- Puede incluir funciones de tarificación.
- Se encarga de los problemas de direccionamiento entre redes heterogéneas.

### NIVEL DE TRANSPORTE

- Puede proporcionar distintos tipos de servicio: fiable (asegura que toda la información será recibida correctamente; si se pierde un paquete se le comunica al origen que lo vuelva a transmitir), no fiable.
- Realiza comunicación extremo a extremo (entre origen y destino), sin intermediarios.
- Realiza control de flujo extremo a extremo.



### NIVEL DE SESIÓN

Permite a usuarios en distintas máquinas establecer sesiones entre ellos:

- Proporciona mecanismos para controlar el diálogo.
- Gestiona la sincronización entre máquinas.

### NIVEL DE PRESENTACIÓN

- Orden de bytes de enteros.
- Representación de caracteres alfabéticos.
- Tamaño de los tipos de datos.
- Compresión y cifrado de datos.

### NIVEL DE APLICACIÓN

Contiene protocolos que son de utilidad directa para aplicaciones que usan la red:

- Para transmisión de ficheros
- Para correo electrónico
- Para control de trabajos remotos

**INTERCONEXIÓN DE REDES** Es posible gracias a los distintos tipos de nodos. Esta interconexión puede ser:

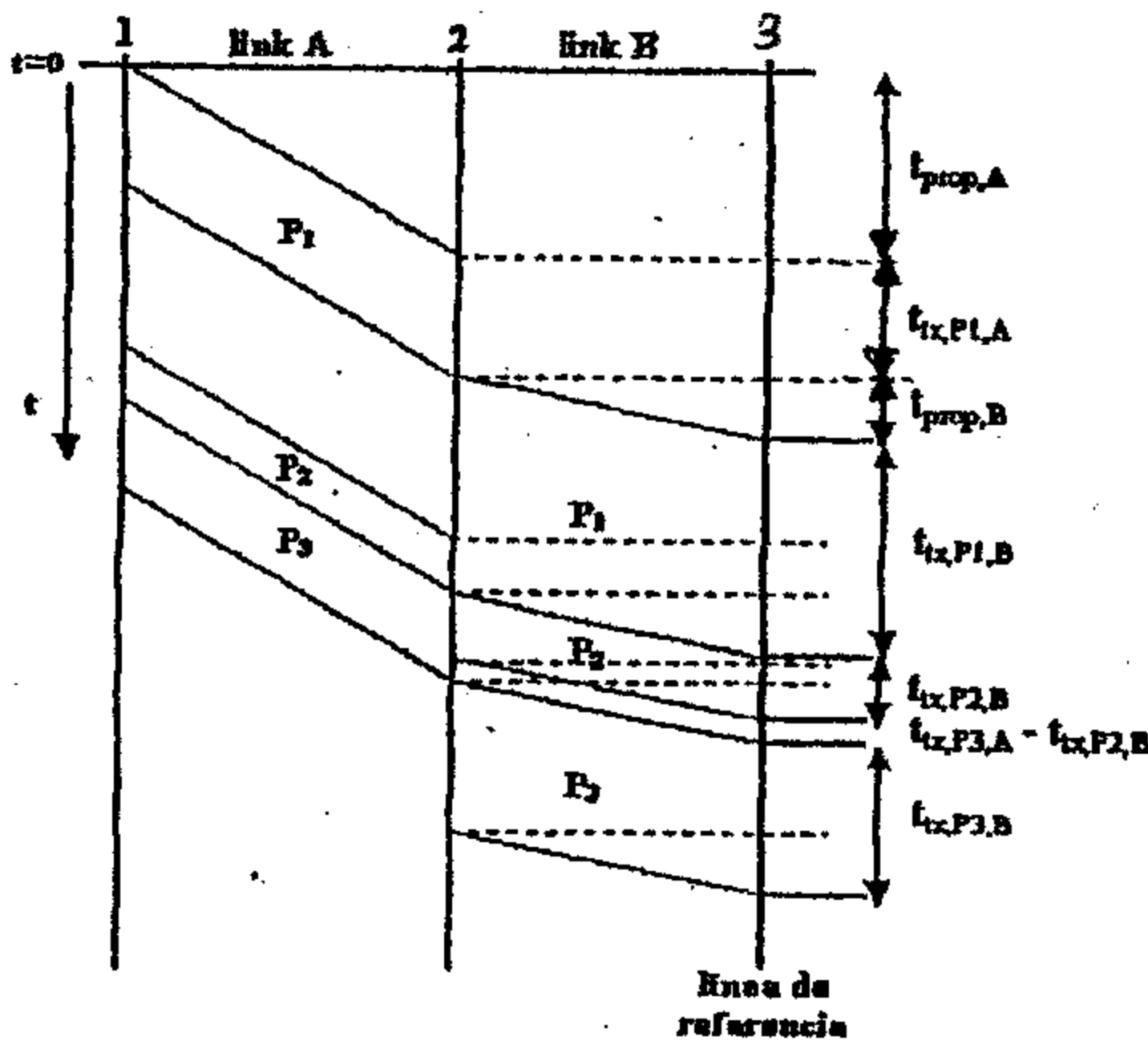
- ~~nivel físico: repetidores (repeater), concentradores (hubs)~~
  - ~~nivel de enlace: puentes (bridges), conmutadores (switches)~~ lee y filtra direcciones.
  - ~~nivel de red: enrutadores (routers)~~ permite conectar distintos tipos de redes.
- por encima del nivel de red se suelen llamar pasarelas de nivel N (gateways)

### CRONOGRAMAS

- sirven para calcular el tiempo de transmisión de información entre un origen y un destino.
- se pueden utilizar tanto en conmutación de circuitos como en conmutación de paquetes
- el tiempo viene representado en el eje vertical.

El siguiente cronograma corresponde a:

- la transferencia de tres paquetes (P1, P2 y P3) de distinto tamaño (longitud)
- transferencia a través de tres nodos (1, 2, 3) (o a través de dos enlaces - linkA y linkB).
- los nodos transmiten la información a distinta velocidad
- la velocidad de propagación en cada enlace es distinta ( $\Rightarrow$  distintas pendientes)



$\triangleright$   $t_{tx}$ , tiempo de transmisión: es la relación entre la longitud (L, en bits) a transmitir y la velocidad de transmisión (capacidad, C en bits/seg) del enlace.

$$t_{tx} = \frac{L}{C} \left( \frac{\text{bits}}{\text{bits/seg}} \right)$$

$\triangleright$   $t_{prop}$ , tiempo de propagación: es la relación entre la distancia (D, en Km) del enlace y la velocidad de propagación ( $v_{prop}$ , en Km/seg) en el enlace. No influye en absoluto la cantidad de bits a transmitir: el  $t_{prop}$  será el mismo para 1 que para 1000 bits.

$$t_{prop} = \frac{D}{v_{prop}} \left( \frac{\text{Km}}{\text{Km/h}} \right)$$

En el caso de conmutación de paquetes (en el cronograma anterior no se ha puesto por simplificación) se debe tener en cuenta:

- $\triangleright$  ~~TIEMPO DE PROCESO~~ tiempo que tardan los nodos en extraer información de la cabecera de los paquetes para enrutarlos, hacer control de errores, etc... Nosotros haremos las siguientes consideraciones:
  - para procesar un paquete, el nodo tiene que haber recibido el paquete entero
  - 1 nodo no puede procesar más de un paquete a la vez
  - mientras procesa un paquete, el nodo puede estar recibiendo o transmitiendo otros paquetes

### LIMITACIONES

La IDEA que SIEMPRE debemos tener clara es que no se le puede hacer llegar información a ningún nodo de la red a una velocidad que provoque que el nodo se "llene".

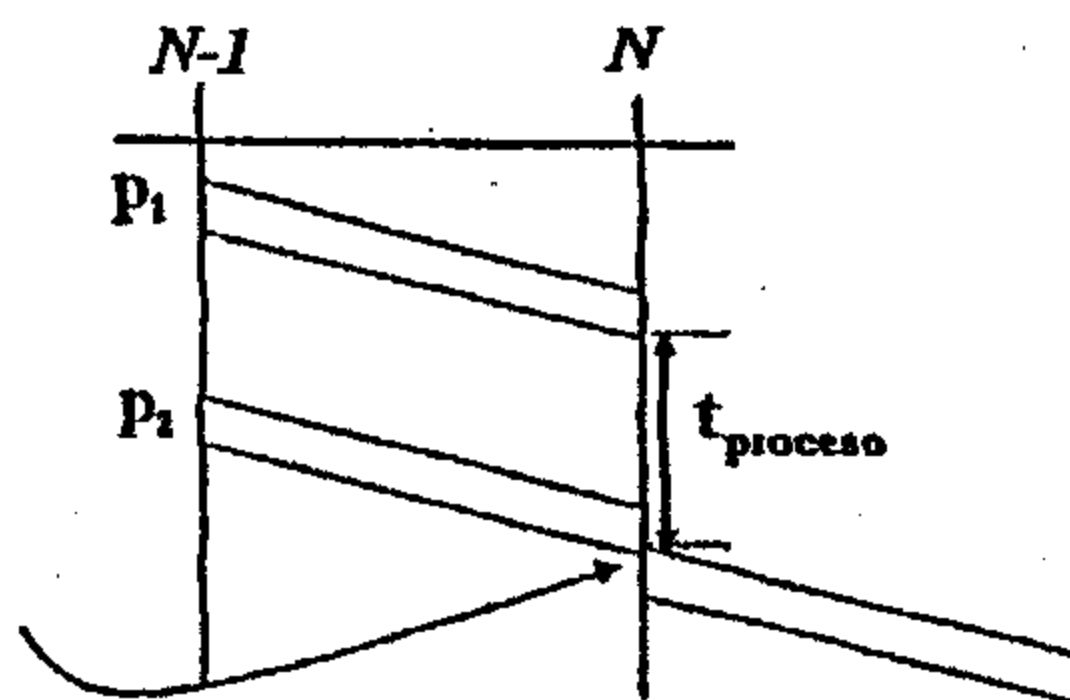
~~La cadencia (paq/seg) a la que una fuente puede mandar paquetes a través de una ruta viene limitada por el mayor tiempo a lo largo de la ruta.~~

Los 2 tiempos que normalmente limitan la cadencia son el tiempo de proceso y el tiempo de transmisión, con lo que podemos distinguir los siguientes casos :

CASO A : si  $t_{proc} > t_{tx}$  la limitación viene dada por el tiempo de proceso en el nodo

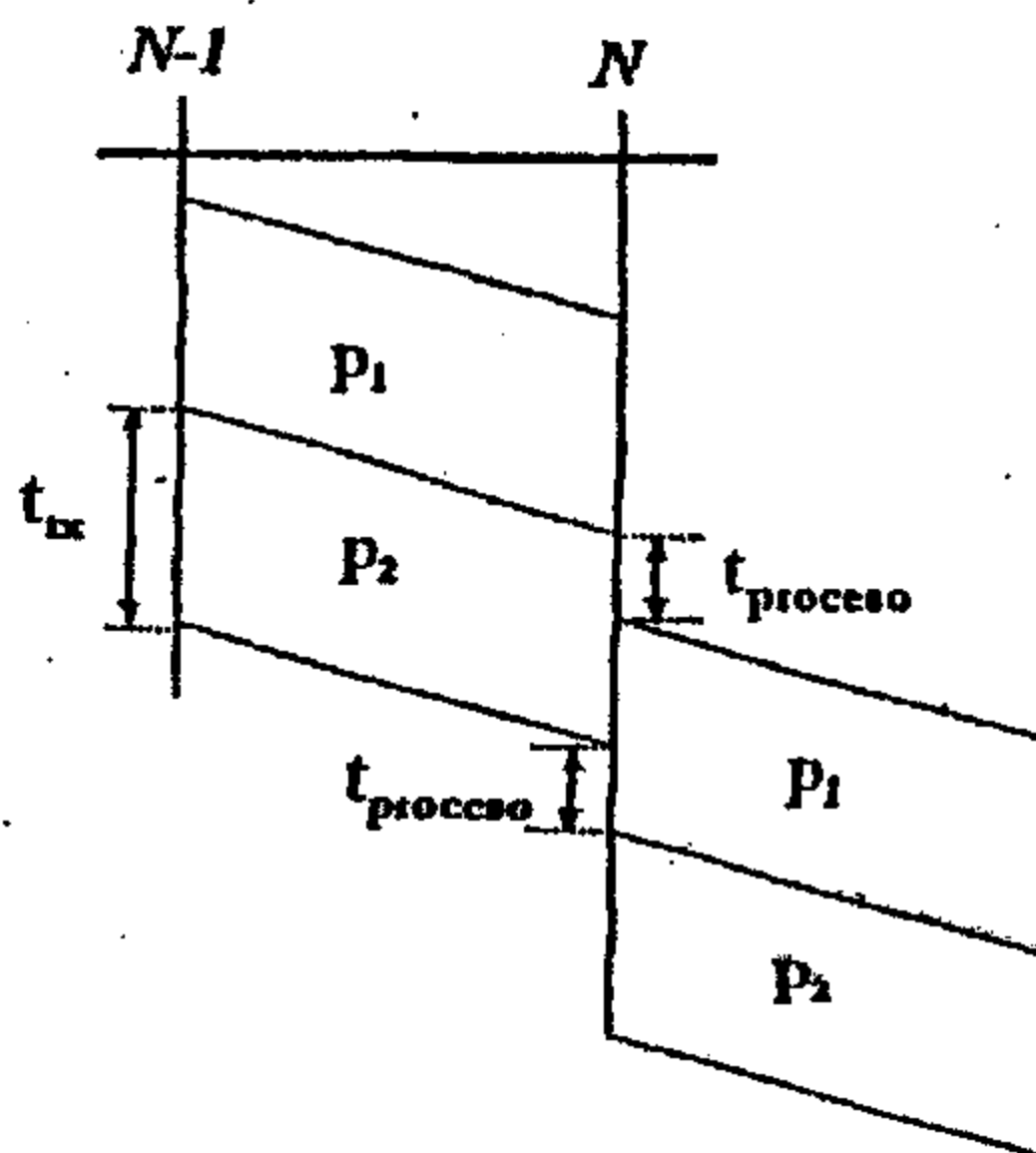
Se hacen coincidir:

- el instante de tiempo en el que se procesa el último bit de p1
- el instante en el que llega el último bit de p2



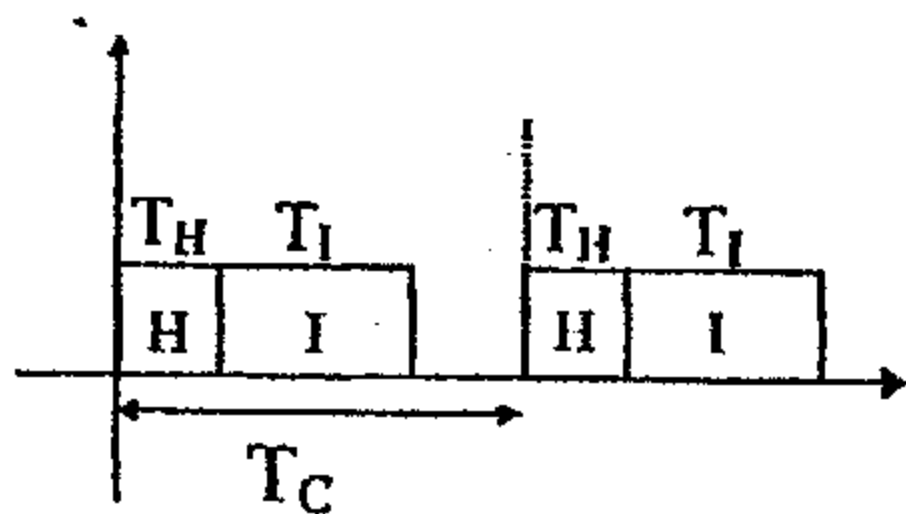
CASO B : si  $t_{proc} < t_{tx}$  la limitación viene dada por el tiempo de transmisión del paquete

La fuente envía un paquete cada  $t_{tx}$



### DEFINICIONES

Finalmente vamos a ver algunas definiciones que nos pueden ser útiles de cara a examen:



UTILIZACIÓN DEL ENLACE

$$U \triangleq \frac{t_H + t_I}{T_C}$$

EFICIENCIA

$$\eta \triangleq \frac{t_I}{T_C}$$

RENDIMIENTO DEL PROTOCOLO

$$\rho \triangleq \frac{I}{I+H}$$

H : Header  
I : Info

$$PER = 1 - (1 - BER)^n$$

n : # bits per packet

## TEMA 2. REDES DE CONMUTACIÓN DE PAQUETES

### REDES DE AREA LOCAL (LAN)

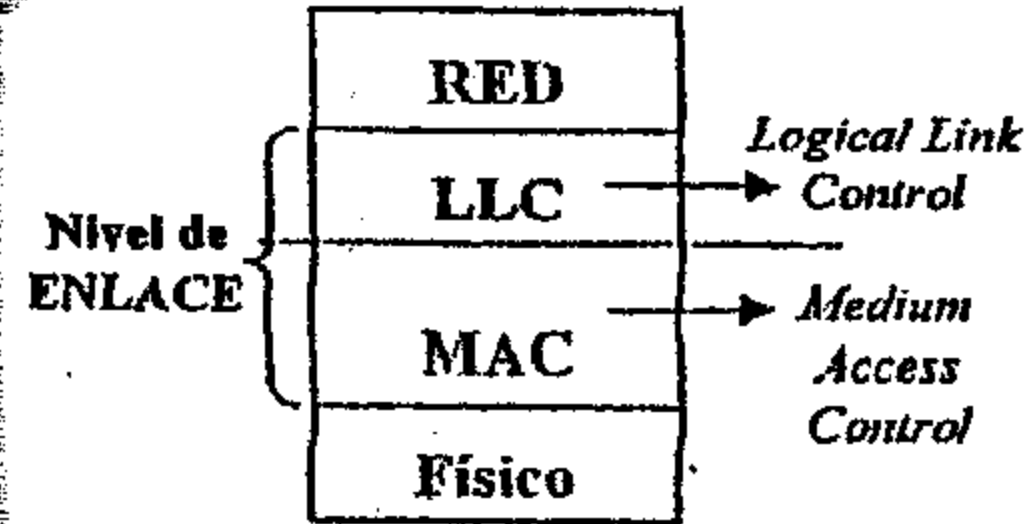
#### MECANISMOS DE ACCESO AL MEDIO

Cuando diversas estaciones quieren acceder a un mismo medio de transmisión es necesario establecer mecanismos de control de acceso al medio.

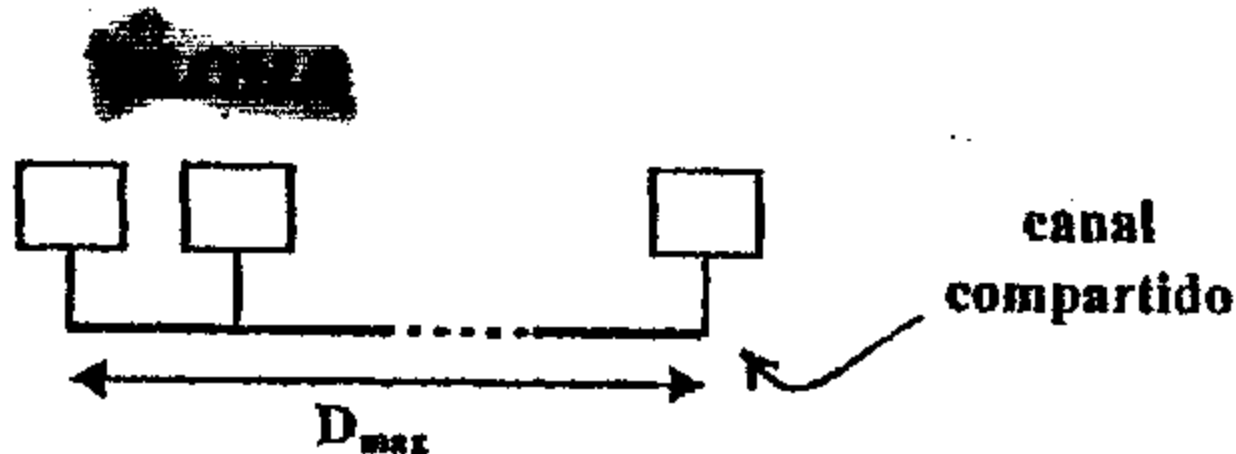
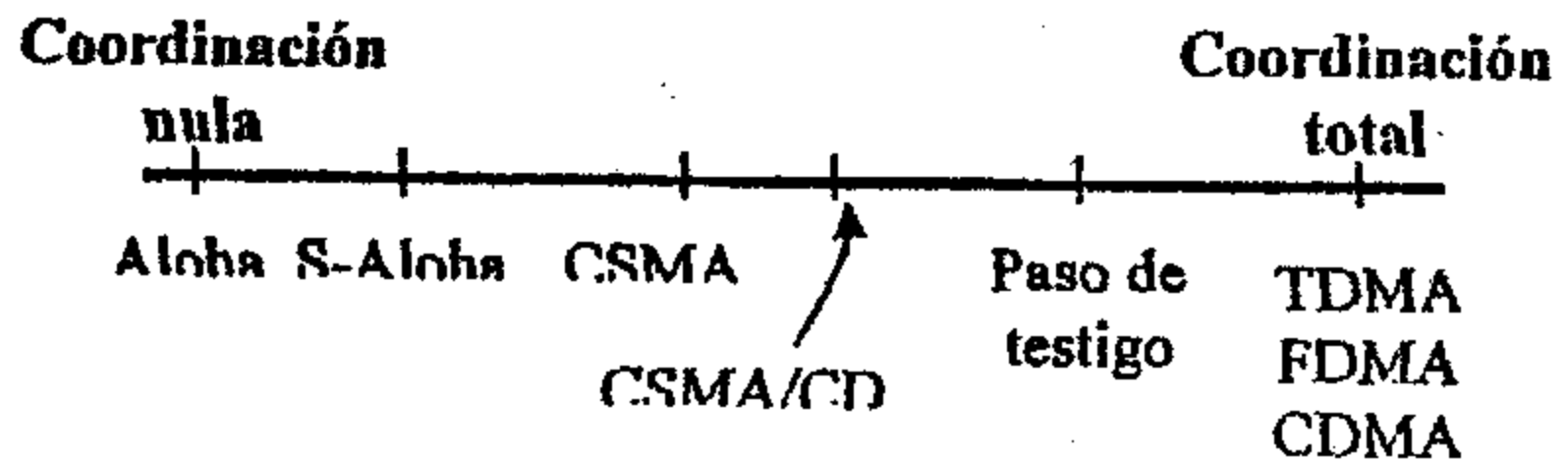
Estos mecanismos se sitúan en la capa MAC (Medium Access Control) dentro del NIVEL DE ENLACE

Estas técnicas de acceso al medio se pueden clasificar en :

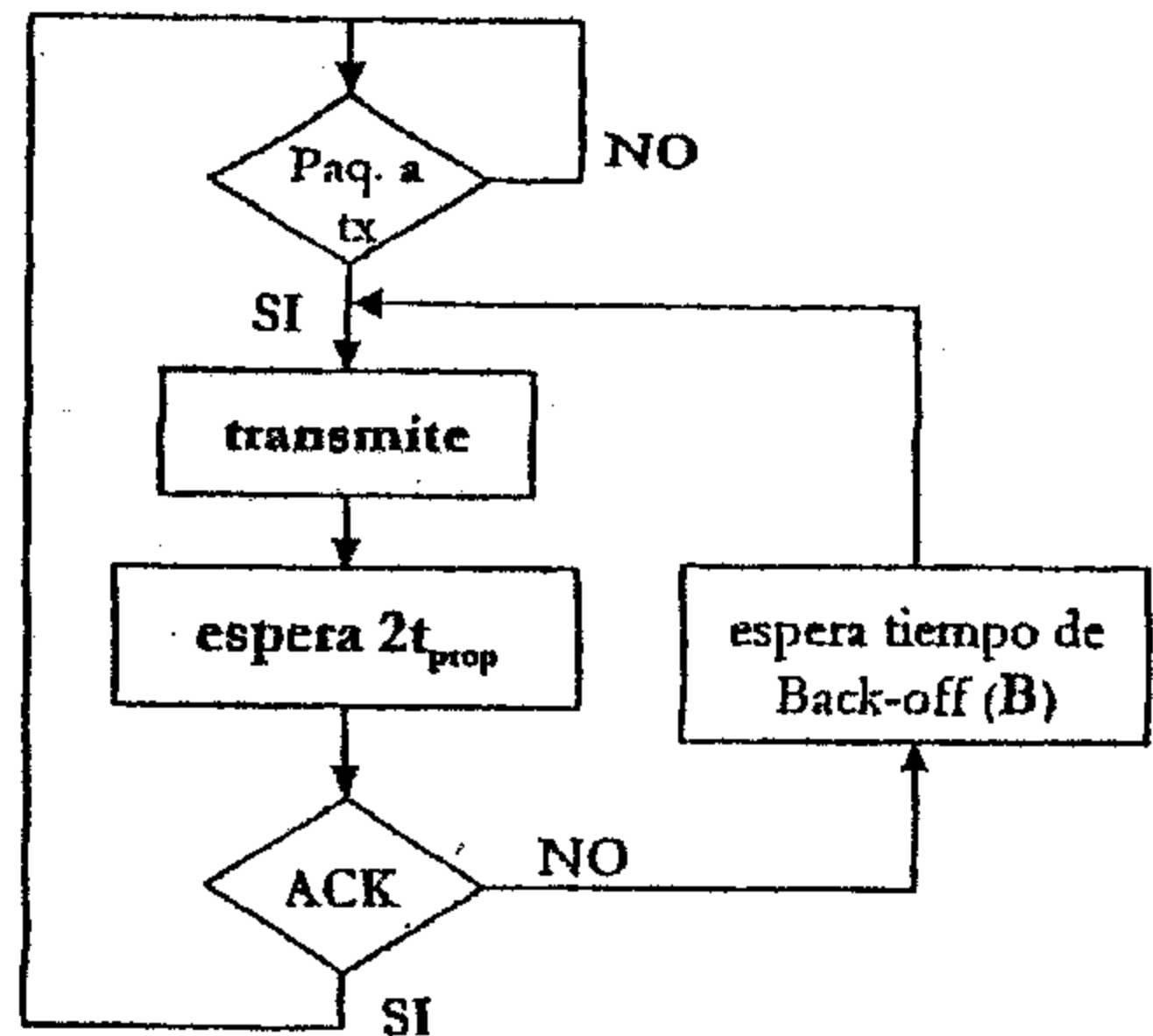
- ✓ **DETERMINISTAS:** las estaciones obtienen el medio de acuerdo a un esquema predeterminado. Inconveniente -> desaprovechan el medio cuando una estación no transmite nada
  - ~~Asignación fija (FDMA - visto anteriormente)~~
  - ~~Asignación por demanda (Polling, Paso de testigo)~~
- ✓ **ALEATORIAS:** las estaciones intentan acceder al medio cuando lo necesitan. Inconveniente -> los paquetes pueden colisionar, con lo que las estaciones tienen que retransmitir la información.
  - ~~ALOHA, ALOHA controlado (S-ALOHA)~~
  - ~~CSMA (Carrier Sense Multiple Access), CSMA/CD (Collision Detect)~~



Bajo qué condiciones es preferible utilizar una técnica u otra? En general, en condiciones de poco tráfico se comportan mejor las aleatorias y para alto tráfico las deterministas



- Si solo transmite una estación -> el paquete se recibe OK en la estación destino; en este caso la estación destino devuelve el ACK a la estación origen.
- Si 2 estaciones transmiten a la vez -> se produce una colisión; en este caso las estaciones destino no devuelven el ACK.
- Una estación origen espera dos veces el retardo máximo de propagación ( $2 \cdot t_{prop}$ ), y si después de este tiempo no ha recibido ACK supone que ha habido colisión -> espera un tiempo aleatorio de backoff y vuelve a transmitir el paquete.



**EFICIENCIA DEL ALOHA**

- Se considera que el tráfico ofrecido por las estaciones sigue una distribución de Poisson (es decir, se considera un número suficientemente elevado de estaciones)
- ~~El tiempo necesario para transmitir un paquete~~ (incluyendo el máximo retardo de propagación)  
(a menudo el retardo de propagación no se considera)

$$t_{tx} = \frac{L}{R_{tx}} \quad t_{prop} = \frac{D_{max}}{v_{prop}}$$

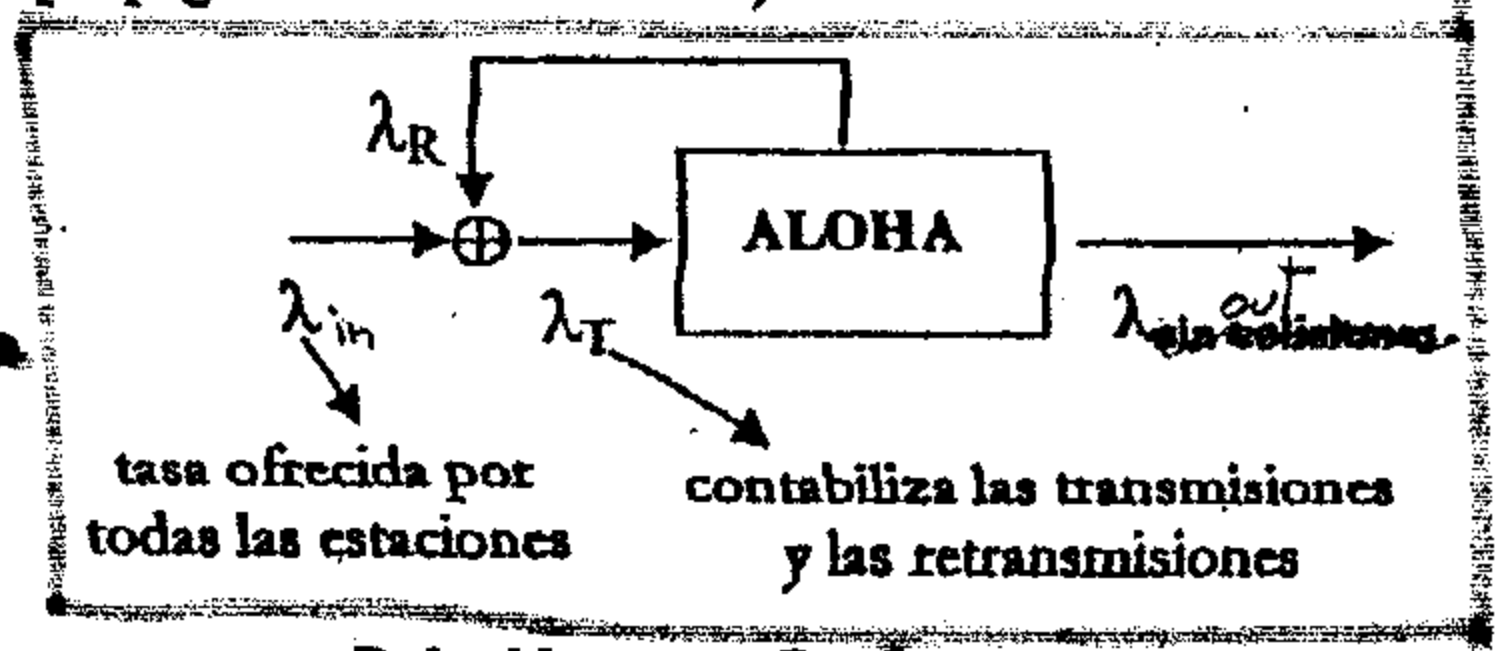
- ~~S (caudal o throughput): número medio de paquetes que se transmiten por unidad de tiempo P. Se cumple  $0 < G < \infty$~~

$$S = G \cdot P \{tx\ ok\}$$

- ~~S (tráfico ofrecido): número medio de intentos de tx. (transmisiones + retransmisiones) por unidad de tiempo P. Se cumple  $0 < G < \infty$~~

$$S_{max} = \frac{1}{2e} \quad \text{cuando } G = \frac{1}{2}$$

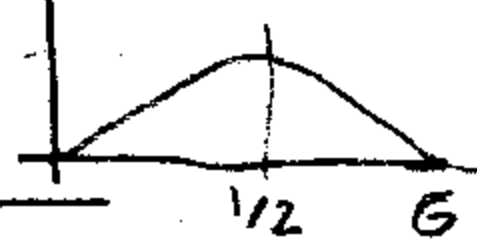
$\lambda$  : taxa paquets =  $\frac{\# \text{packets}}{\text{temps}}$



Relación entre G y S:

$$S = G \cdot P \{tx\ ok\} = G \cdot e^{-2\lambda \cdot P}$$

$$S_{max} = \frac{1}{2e} \quad \text{cuando } G = \frac{1}{2}$$



- Refinamiento del Aloha puro.
- ~~El tiempo se divide en slots (recursos) de longitud fija P.~~
- ~~Una estación sólo puede transmitir al inicio de un slot~~

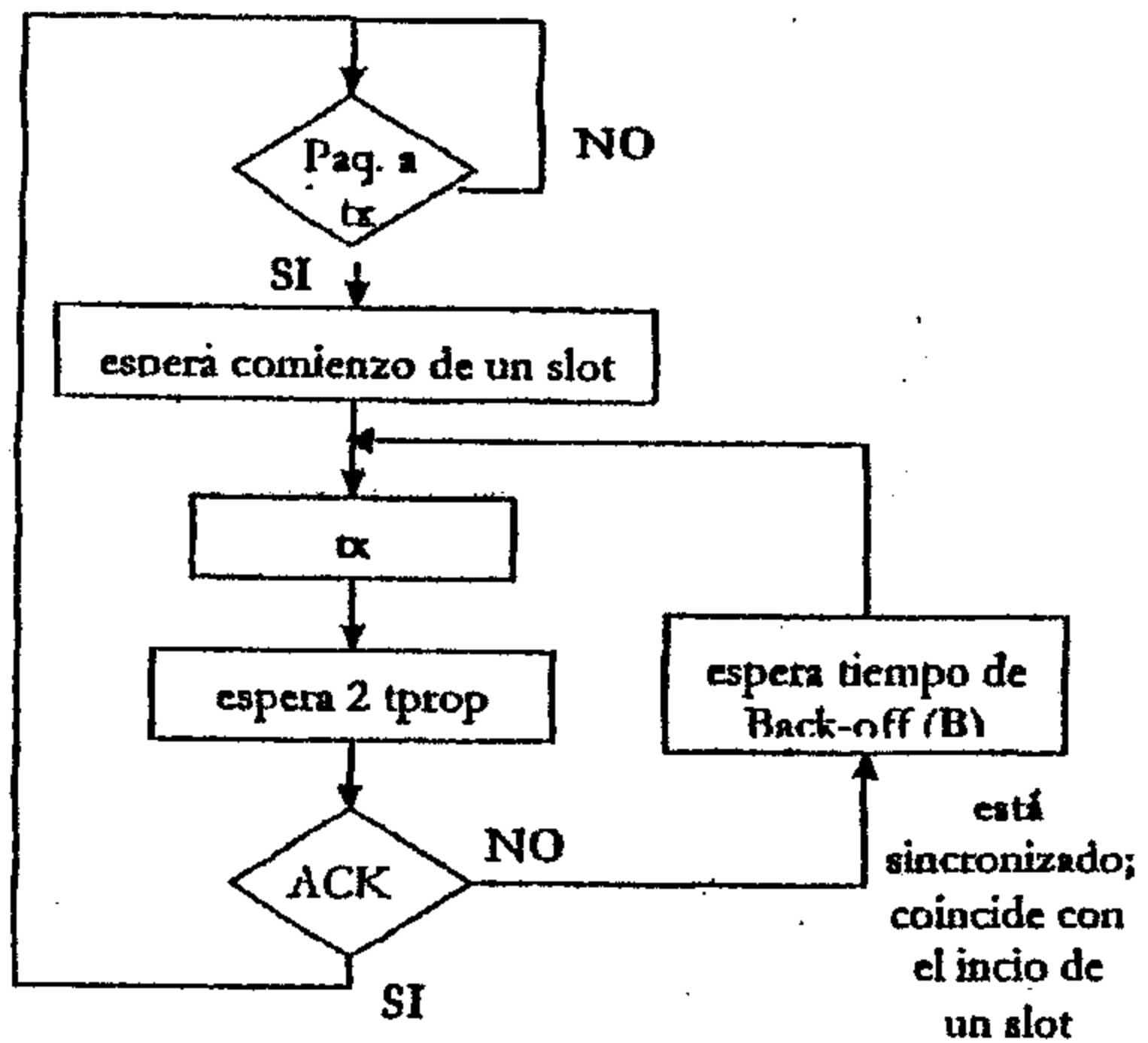
**EFICIENCIA DEL ALOHA RANURADO**

Se utilizan las mismas definiciones de P, S y G que para el Aloha puro.

Relación entre G y S:

$$S = G \cdot P \{tx\ ok\} = G \cdot e^{-\lambda \cdot P}$$

$$S_{max} = \frac{1}{e} \quad \text{cuando } G = 1$$



Antes de empezar con el CSMA vamos a ver un parámetro importante:

$$a = \frac{t_{prop}}{t_{tx}} = \frac{D_{max}}{d_{tx}}$$

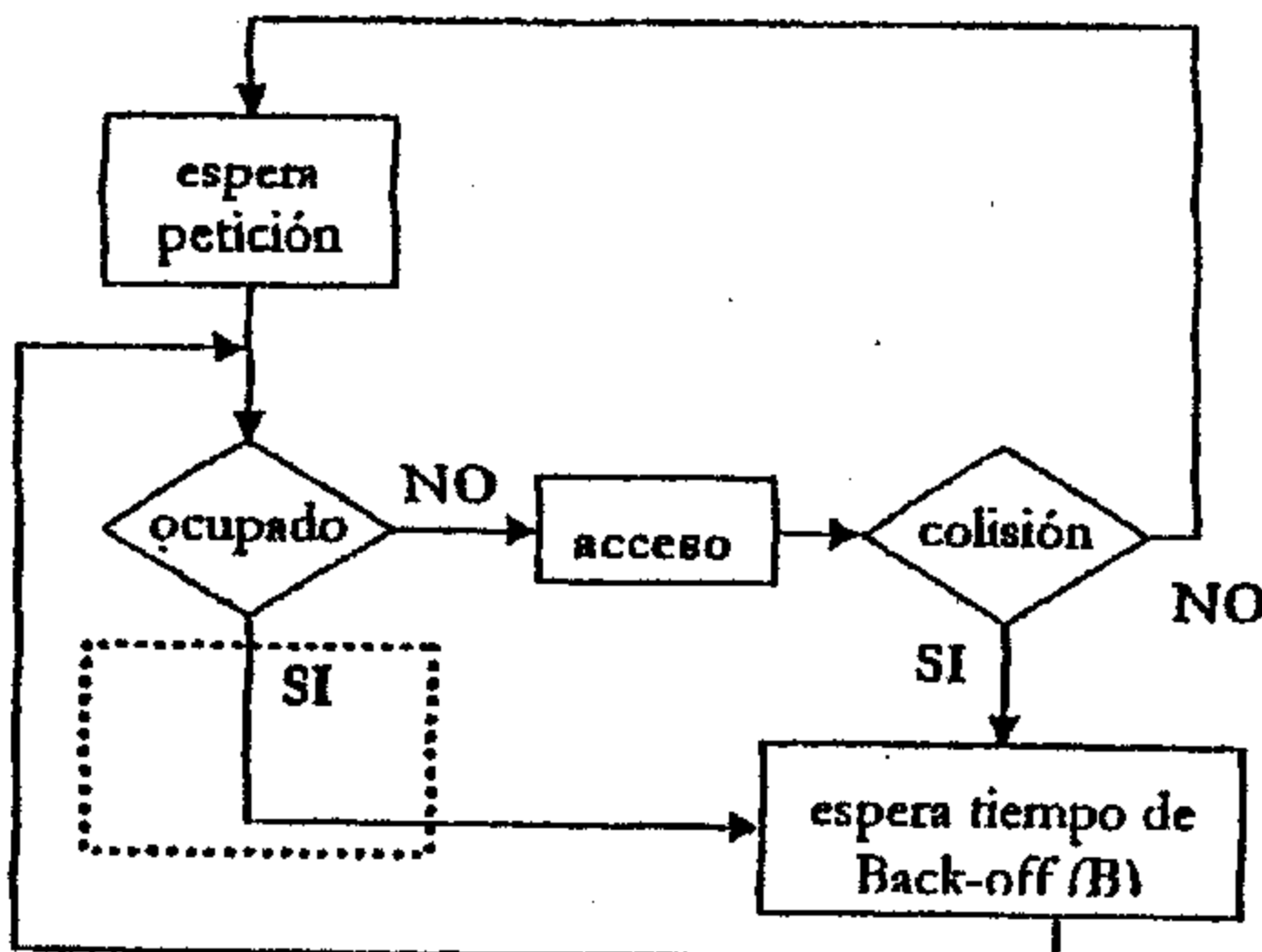
El rendimiento del CSMA depende del valor de a; para valores de a pequeños, CSMA funciona mejor, puesto que las estaciones detectan de forma más eficaz si el canal está ocupado; para valores grandes de a el rendimiento de CSMA puede ser peor que el del Aloha

**CSMA (Carrier Sense Multiple Access)**

- Refinamiento del Aloha Ranurado: antes de tx 1 paquete las estaciones escuchan el canal:
  - ✓ si está libre → transmiten
  - ✓ si está ocupado → esperan
- La "forma de esperar" depende del tipo de protocolo; existen 2 clases de protocolos:
  - ✓ ranurada (slots de duración  $t_{prop}$ )
  - ✓ no-ranurada
- Y cada uno puede trabajar de forma:
  - ✓ no persistente
  - ✓ p-persistente

Nosotros solo trabajaremos con protocolos no-ranurados.

**CSMA (no persistente)**



**FUNCIONAMIENTO** -> cuando la estación quiere tx un paquete, escucha el canal:

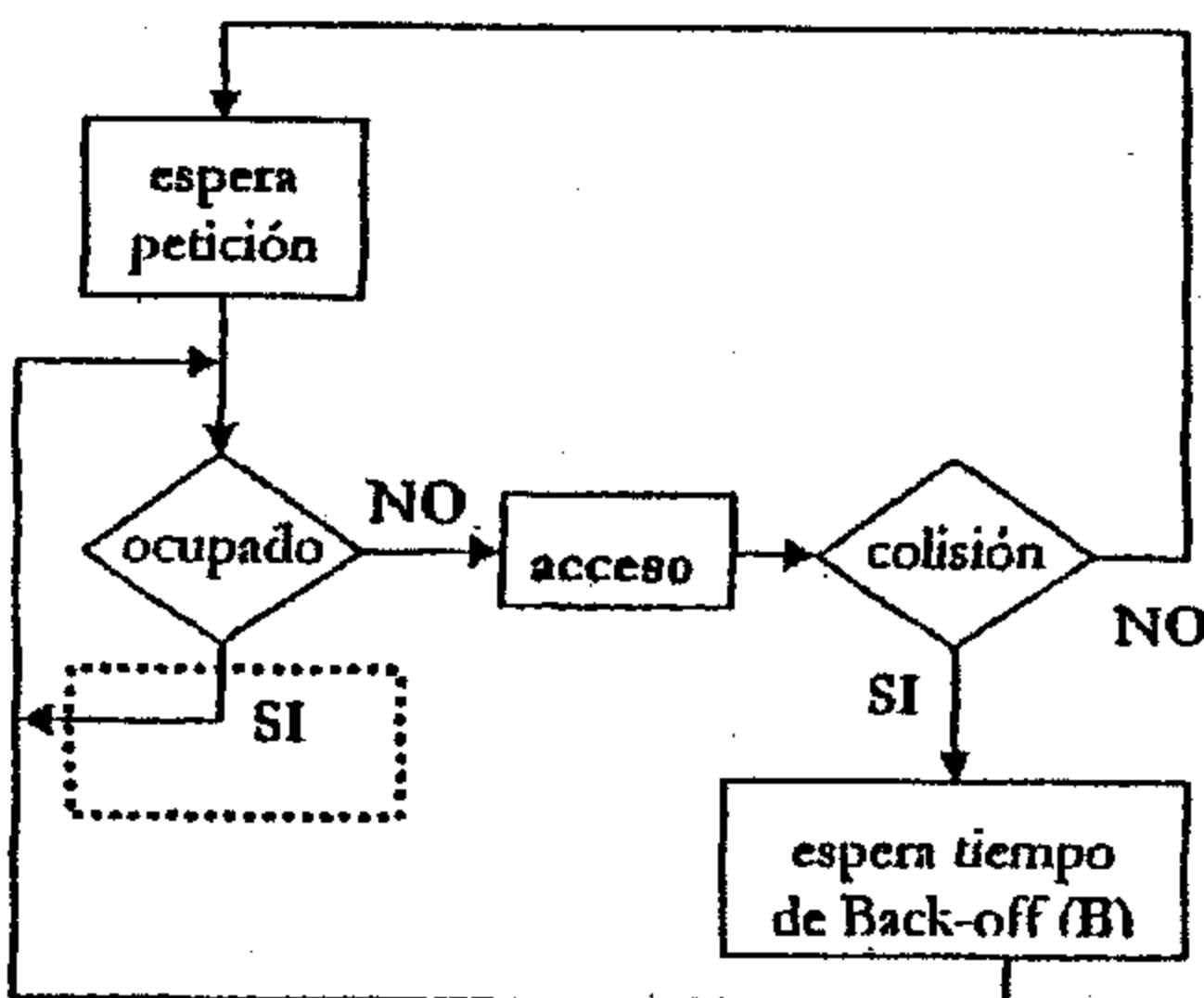
- 1- si medio libre → accede
- 2- si medio ocupado → espera B y vuelve a 1
- 3- si colisión → espera B y vuelve a 1

**CAUDAL (THROUGHPUT)**

$$S = \frac{G \cdot e^{-aG}}{G(1+2a) + e^{-aG}}$$

$$\lim_{a \rightarrow 0} S = \frac{G}{G+1}$$

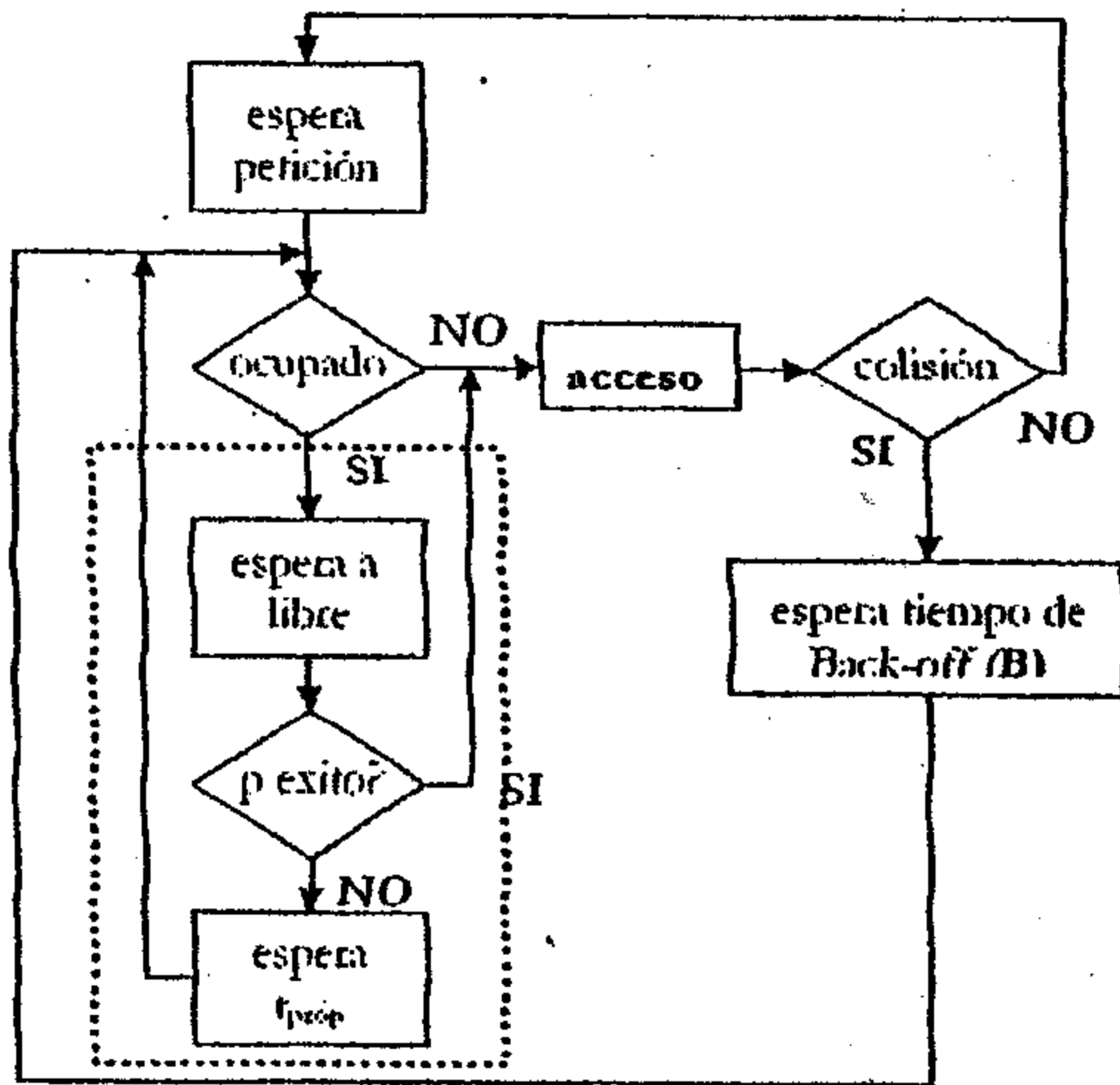
**CSMA (1 persistente)**



**FUNCIONAMIENTO** -> cuando la estación quiere tx un paquete, escucha el canal:

- 1- si medio libre → accede
- 2- si medio ocupado → espera hasta libre y accede
- 3- si colisión → espera B y vuelve a 1

**CSMA (p. persistente)**

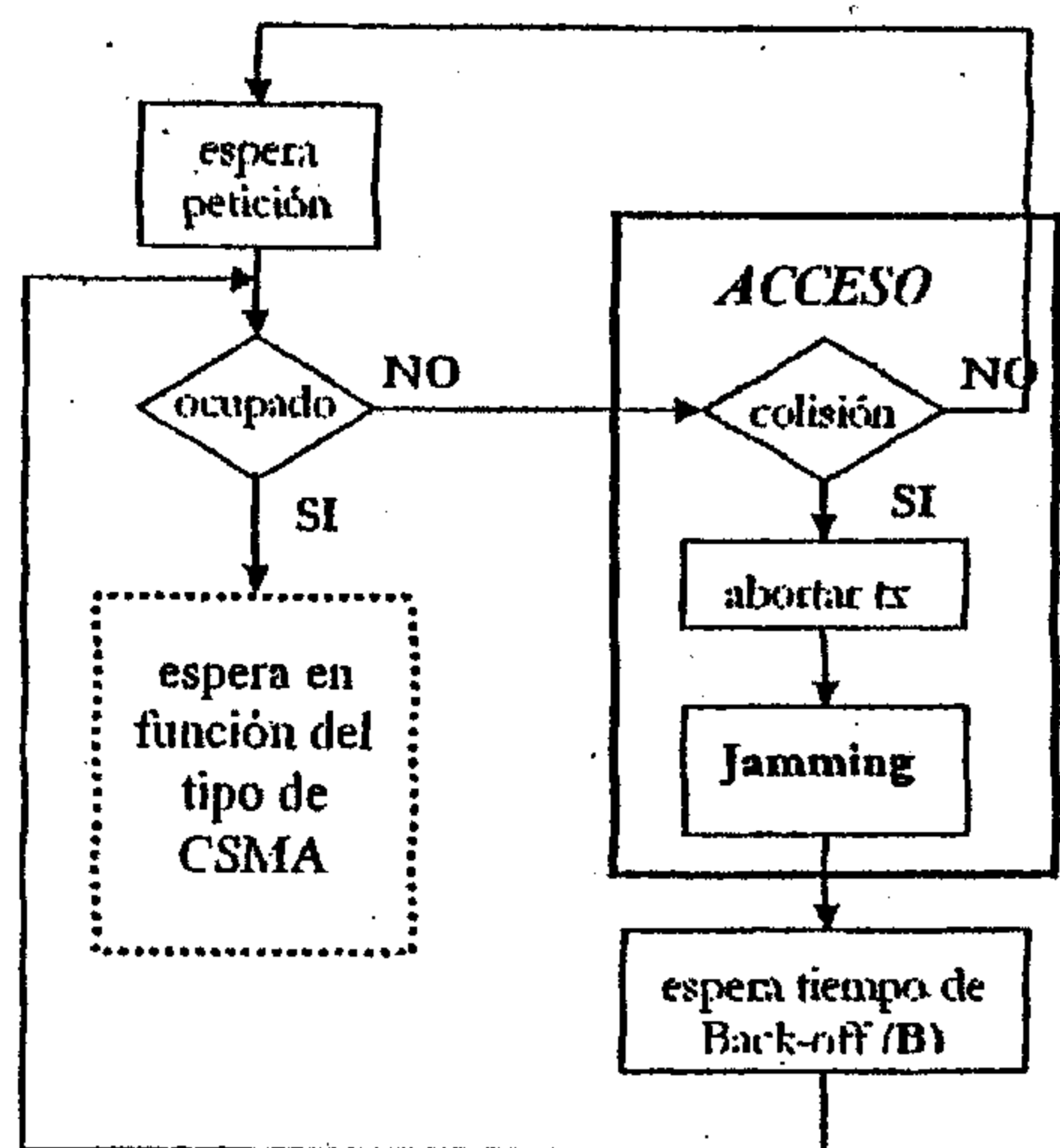


**FUNCIONAMIENTO** -> cuando la estación quiere tx un paquete, escucha el canal:

- 1.- si medio libre -> accede
- 2.- si medio ocupado -> espera hasta libre y accede con probabilidad p
- 3.- si no accede espera tprop y vuelve a 1
- 4.- si colisión -> espera B y vuelve a 1

**CSMA/CD (Carrier Sense Multiple Access / Collision Detect)**

- Mismo funcionamiento que el CSMA, con la diferencia que las estaciones, ~~después de transmitir el paquete,~~ <sup>antes de transmitir el paquete,</sup> siguen escuchando el medio para ver si se produce colisión.
- Si una estación detecta colisión:
  - 1.- Aborta la tx en curso y espera un tiempo B, después del cual intenta acceder de nuevo al medio
  - 2.- Lo notifica al resto de estaciones: esto se hace mediante lo que se llama señal de Jamming
- Algunas implementaciones tienen un algoritmo de backoff dinámico:



CSMA/CA (Collision Avoided)

## CONMUTACIÓN DE PAQUETES

### LAN'S. ASPECTOS BÁSICOS

Las LAN's son redes que conectan directamente entre sí equipos situados en un ámbito geográfico local; el medio de transmisión es compartido por todas las estaciones

#### ► TOPOLOGÍAS

Hay tres topologías físicas básicas:

- **Bus:** típicas de las redes Ethernet antiguas; cada dispositivo está conectado a un cable común, con lo que todos pueden ver todas las señales de todos los demás dispositivos
- **Anillo:** típicas en las redes Token Ring y FDDI; para que la información pueda circular, cada estación debe transferir la información a la estación adyacente
- **Estrella:** utilizadas en todas las LAN actuales en sustitución de los buses y los anillos; hay un nodo central que permite que todos los demás nodos se comuniquen entre sí de manera conveniente; la desventaja principal es que si el nodo central falla, toda la red se desconecta. Según el tipo de nodo utilizado (hub o switch) hacemos una subdivisión.

Detectar colisiones  $\Rightarrow T_{TXmin} \geq 2T_{prop}$

~~IEEE 802.3~~ (típicamente conocido como Ethernet) es actualmente el protocolo más utilizado. Utiliza topología en estrella (hub o switch).

~~El acceso al medio es utilizado el método CSMA/CD~~

A nivel físico existen las siguientes implementaciones: Ethernet (10 Mbps), Fast Ethernet (100 Mbps), Gigabit Ethernet (1 Gbps, 10 Gbps), Wireless LAN (802.11)

- El medio físico de tx puede ser:
- cables de cobre STP o UTP
  - cable coaxial
  - fibra óptica multimodo y monomodo

- El método de codificación es Manchester; el tiempo de bit se divide en dos mitades y:
- el 1 corresponde a una transición baja-alta
  - el 0 a una transición alta-baja

Ethernet y IEEE 802.3 son en realidad dos estándares distintos; aunque sus tramas son parecidas tienen alguna diferencia: IEEE 802.3 normalizó, además de las funciones de capa física y de subcapa MAC, una serie de funciones de control de enlace de datos que incluyó en la denominada LLC (Logical Link Control)

#### ► FORMATO DE LA TRAMA ETHERNET:

preambulo	SFD control	MAC dest.	MAC orig.	TIPO	DATOS	CRC
62 bits	26 bits	48 bits	48 bits	16 bits	46-1500 bytes	32 bits

#### ► FORMATO DE LA TRAMA IEEE 802.3

preambulo	control	MAC dest.	MAC orig.	LONGI	DATOS	<del>MAC</del> PAD	CRC
7 bytes	3 byte	6 bytes	6 bytes	2 bytes	0-1500 bytes	0-46 bytes	4 bytes

- preámbulo: 7 bytes.
- sincronización (SFD-Start Frame Delimiter): 1 byte.
- MAC destino: 48 bits; tres tipos de direcciones destino son posibles: individual, de multicast y de broadcast
- MAC origen: 48 bits. Su significado es el mismo que la dirección destino.
- tipo (Ethernet): 16 bits; identifica el tipo de protocolo que hay en el campo de datos.
- longitud (802.3): 16 bits: longitud total del mensaje
- datos: este campo contiene los datos a transmitir
- CRC: 32 bits; control de errores
- ~~El tamaño de trama es como mínimo de 64 bytes y como máximo 1518 bytes.~~

➤ **IMPLEMENTACIONES** (solo vamos a ver la ethernet a 10 Mbps)

**ETHERNET 10 MBPS:** se definen 4 configuraciones básicas: 10Base2, 10Base5, 10BaseT y 10BaseF

**10BASE2 - Coaxial Ethernet fino**

- cable coaxial apantallado y blindado.
- topología en bus, con conectores en T o mediante un transceiver.
- longitud máxima del segmento=185m.
- número máximo de segmentos = 5 conectados por repetidores.
- longitud máxima de una red 10Base2 = 925m.
- distancia mínima entre dos estaciones = 0,5 m
- número máximo de conexiones por segmento=30

**10BASE5 - Coaxial Ethernet grueso**

- cable coaxial apantallado y blindado.
- topología en bus, mediante un transceiver.
- longitud máxima del segmento=500m.
- número máximo de segmentos = 5 conectados por repetidores.
- longitud máxima de una red 10Base5 = 2500m.
- distancia mínima entre dos estaciones = 2,5m.
- número máximo de conexiones por segmento=100

**10BASET - Ethernet sobre pares trenzados**

- cable con pares trenzados, UTP-3 o UTP-5
- estaciones conectadas a un hub; topología en estrella o cascada de estrellas conectadas entre ellas mediante hubs.
- distancia máxima entre un hub y una estación (o entre dos hubs)= 100 m

**10BASEF - Ethernet sobre fibra óptica**

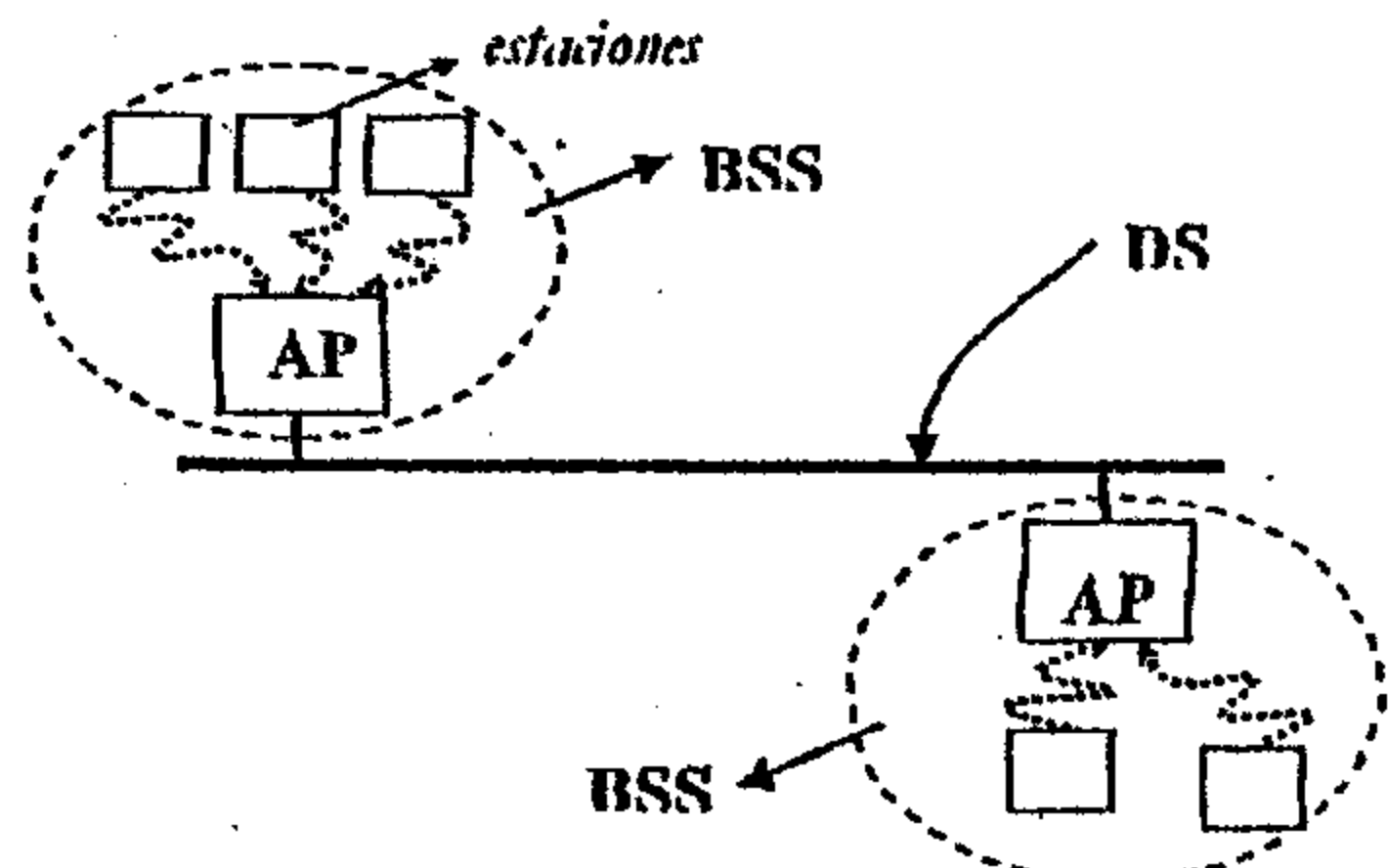
- conectores del tipo SC o ST.
- 10BaseF define 3 variantes:
  - ✓ 10BaseFP (*passive star*): repetidor óptico pasivo, máximo de 33 nodos y 1Km/segmento.
  - ✓ 10BaseFL (*link*): interconecta nodos o repetidores con un límite de 2 Km.
  - ✓ 10BaseFB (*backbone*): interconecta repetidores hasta 2 Km entre ellos con transmisión síncrona.

**WIRELESS LAN. 802.11**

- Una red de área local **no cableada** o WLAN (Wireless LAN) utiliza ondas electromagnéticas (radio o infrarrojo) para enlazar los equipos conectados a la red.
- Existen diversas especificaciones de WLAN's; nosotros veremos la 802.11.
- ~~Las CONFIGURACIONES se pueden dividir en 2 tipos:~~
  - ✓ ~~redes peer-to-peer (o ad-hoc):~~ es la configuración más sencilla, puesto que los únicos elementos necesarios son terminales móviles equipados con los correspondientes adaptadores (no es necesario ningún punto de acceso). Se utiliza muy poco
  - ✓ ~~redes que utilizan puntos de acceso (BSS):~~ un esquema general de este tipo de configuraciones podría ser:

El bloque fundamental de esta arquitectura es la celda, conocida como BSS (*Basic Service Set*). Una celda BSS consta de una o más estaciones inalámbricas y una estación central base, conocida como AP (*Access Point*). Los AP realizan a la vez funciones de:

- hub: actúan como un concentrador
- bridge: conectar las estaciones a otras redes





Múltiples AP se pueden conectar juntos mediante redes cableadas, conformando lo que se llama un DS (*Distribution System*). A todo el conjunto se le denomina una ESS (*Extended Service Set*)

► **CAPA FÍSICA**

IEEE 802.11 define dos opciones para la elección de la capa física:

- radiofrecuencia: ofrece mejor cobertura y puede atravesar objetos sólidos; utiliza tecnología de espectro ensanchado; distinguimos entre (ambas en la banda de frecuencia 2.4 GHz ISM):
  - ✓ espectro ensanchado por secuencia directa o DSSS (*Direct Sequence Spread Spectrum*)
  - ✓ espectro ensanchado por salto de frecuencias o FHSS (*Frequency Hopping Spread Spectrum*)
- luz infrarroja en banda base (o sea, sin modular); cobertura menor

► **CAPA MAC**

- Igual que en una red Ethernet cableada, en una IEEE 802.3 las estaciones deben coordinar sus accesos y el uso del medio de transmisión compartido.
- Existen dos métodos que se pueden utilizar de forma alternada:

- 1.- ~~DCF (Distributed Coordination Function)~~: Se puede utilizar con las dos configuraciones. Método aleatorio en el que las estaciones compiten para acceder al medio (si existe, el AP también tiene que competir para acceder al medio). ~~No hay ningún coordinador~~. Asegura un retardo relativamente pequeño.
- 2.- ~~PCF (Point Coordination Function)~~: Solo se puede utilizar en redes con infraestructura. Método determinista en el que el AP coordina el acceso de las estaciones (por lo tanto puede asegurar un ancho de banda determinado para cada estación). Es útil para aplicaciones que requieren un gran ancho de banda.

▪ **Funcionamiento básico DCF**

SIFS → Control  
PIFS → Gestión  
DIFS → Datos

~~1.- si canal libre durante un tiempo DIFS (Distributed Inter Frame Space) → transmite~~

~~2.- si canal ocupado o se ocupa durante la espera:~~

- espera a que se libere
- vuelve a esperar un tiempo DIFS

~~2.1.- si se vuelve a ocupar durante este tiempo DIFS, vuelve al inicio del punto (2)~~

~~2.2.- si permanece libre durante DIFS → empieza un timeout de timeout:~~

- ~~2.2.1.- si rebota el timeout y nadie ha transmitido → transmite~~
- ~~2.2.2.- si durante el timeout alguien transmite → PARA el timeout, espera a que acabe la transmisión, espera DIFS (otra vez) y después SIGUE con el timeout. Si durante DIFS se vuelve a ocupar el canal, espera a que acabe la transmisión y luego sigue con el timeout → poner un DIF~~

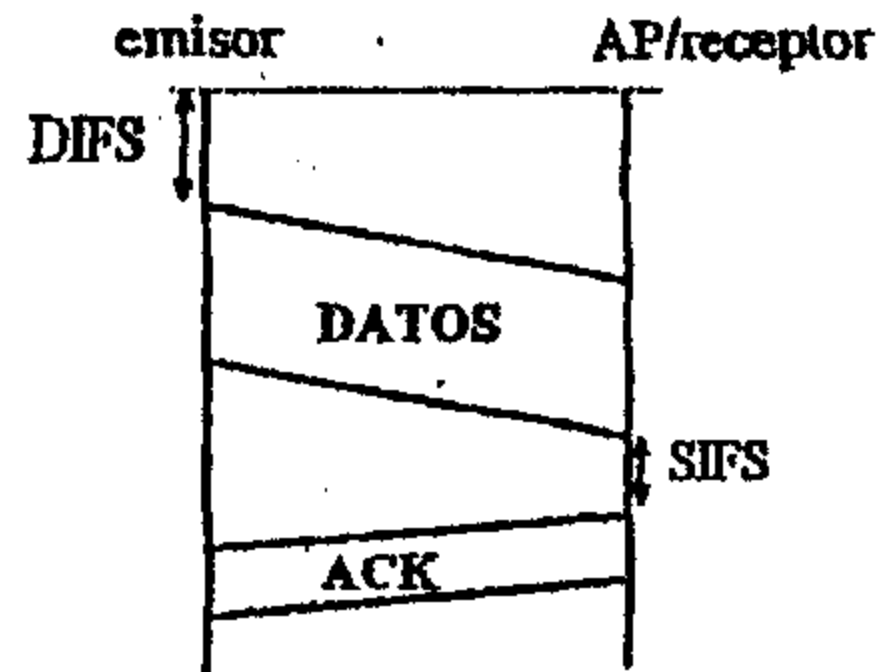
~~3.- cuando la estación receptora ha recibido correctamente la trama, espera un corto período de tiempo SIFS (Short Inter Frame Space), después del cual envía un ACK al emisor.~~

► **FORMAS DE ACCESO AL MEDIO**

- IEEE 802.11 utiliza CSMA/CA (*Carrier Sense Multiple Access*) con <sup>Collision avoided</sup> Prevención de Colisión
- Existen dos formas de acceder al medio, que se pueden utilizar tanto en redes ad-hoc como en redes con infraestructura
- puesto que el canal es poco fiable en ambas se utilizan ACKs

a) **FORMA A: sin reserva del medio**

- ✓ cuando una estación quiere transmitir espera DIFS y si obtiene el medio, tx los datos (hacia el receptor si es una red ad-hoc o hacia el AP si es una red con infraestructura)
- ✓ cuando el AP/receptor ha recibido correctamente la trama de los datos, espera SIFS y devuelve un ACK notificando al transmisor que ha recibido correctamente la información.



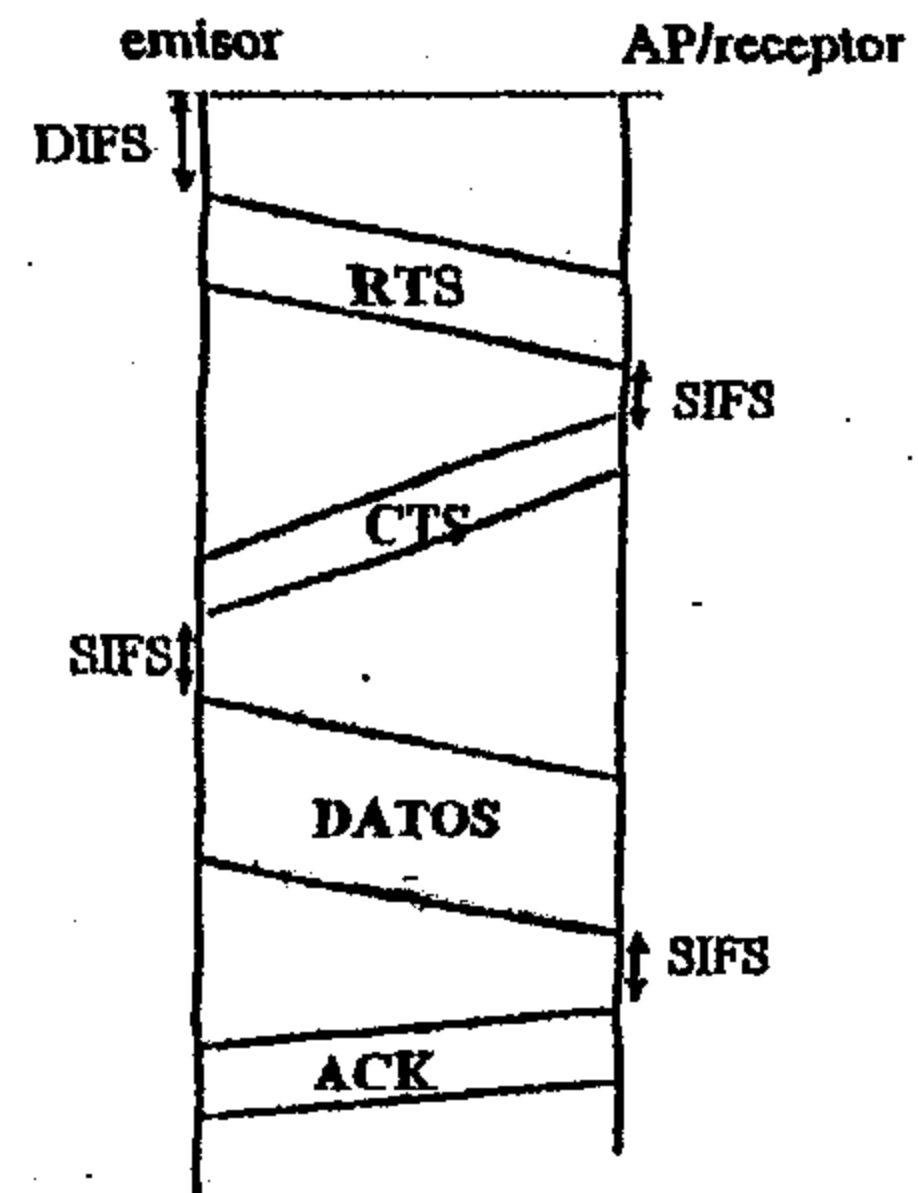
b) **FORMA B: con reserva del medio**

Se utiliza debido a algunos de los siguientes motivos:

- ✓ Existe terminal oculto
- ✓ Hay mucho tráfico en la red
- ✓ Los paquetes de datos son grandes comparados con los paquetes RTS/CTS

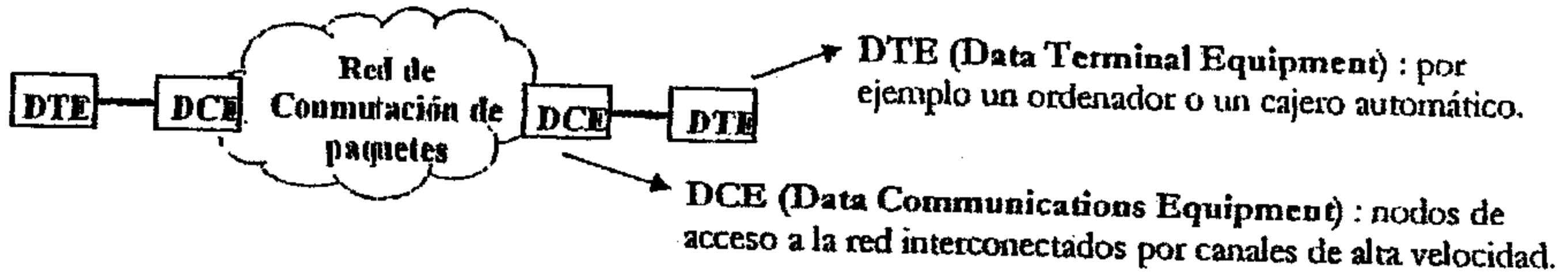
El funcionamiento es el siguiente:

- 1.- cuando una estación está lista para transmitir (i.e, ha detectado medio libre y ya ha esperado DIFS), ~~envía una solicitud RTS~~ (request to send - destino y longitud del mensaje) ~~al punto de acceso AP/receptor, quien difunde un NAV (Network Allocation Vector - señal que indica el tiempo que la estación emisora estará ocupando el medio)~~ a todos los demás nodos que la pueden escuchar, para que queden informados de que se va a transmitir (y que por lo tanto no transmitan). Estos nodos dejarán de transmitir durante el tiempo indicado por NAV más un intervalo backoff.
- 2.- ~~Si el AP/receptor no sufre problemas espera SIFS y luego responde con una autorización (CTS - clear to send). El emisor después de recibir el CTS espera SIFS y envía su trama de datos.~~ Si el emisor no recibe la trama CTS, supone que ocurrió una colisión y los procesos RTS empiezan de nuevo (i.e, vuelve a esperar DIFS etc...)
- 3.- Cuando el AP/receptor ha recibido correctamente la trama de los datos, espera SIFS y devuelve un ACK notificando al transmisor que ha recibido correctamente la información.



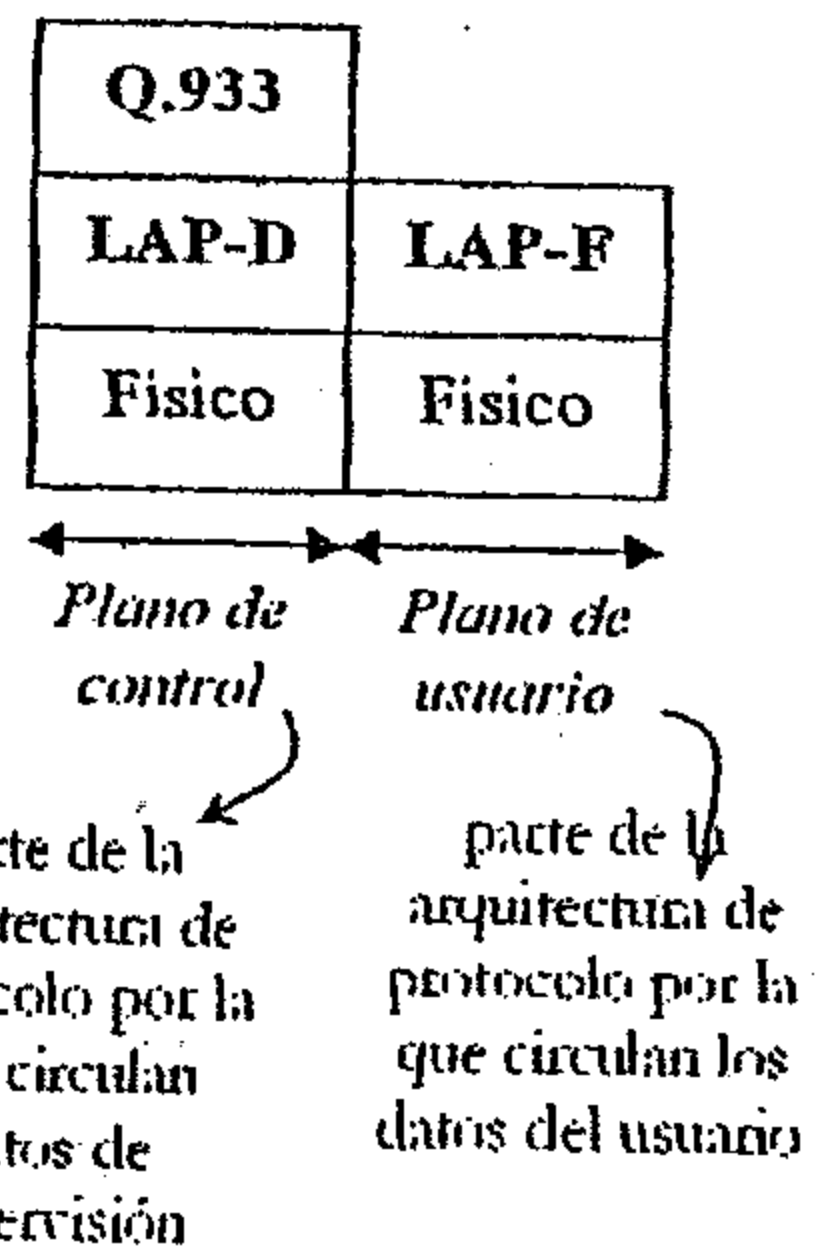
**IMPORTANTE!!!** Con reserva de canal no se producen colisiones de los paquetes de datos, aún así permanece el problema de que las tramas RTS sean enviadas por varias estaciones a la vez, sin embargo estas colisiones son menos dañinas ya que el tiempo de duración de estas tramas es relativamente corto.

**REDES DE AREA EXTENDIDA (WAN- WIDE AREA NETWORK)**



- REDES X.25**
- ~~Red de conmutación de paquetes por circuito virtual~~ Tamaño de paquete fijo: 128 bytes
  - ~~Ofrecen un servicio orientado a conexión y fiable~~ (no duplica ni pierde ni desordena paquetes)
  - ~~Realiza control de flujo y control de errores en cada enlace de la red~~ (i.e, entre nodos adyacentes) -> hay una sobrecarga de procesamiento muy grande ->  $v_m$  bajas (aprox. 64kbps)
  - Útil si los enlaces tienen una probabilidad de error ( $P_e$ ) elevada
  - Soporta dos tipos de circuitos virtuales:
    - ✓ ~~SVC (Switched Virtual Circuit)~~: hay que realizar un diálogo con el nodo de acceso previo a la transmisión para establecerlos
    - ✓ ~~PVC (Permanent Virtual Circuit)~~: están establecidos de antemano (por contrato), así que no requieren de fase de establecimiento
  - Este tipo de redes se estructuran en base a 3 niveles:
    - ✓ NIVEL 1: Físico -> Normas v.28, v.24, x.21
    - ✓ NIVEL 2: Enlace -> (basado en HDLC) Protocolo LAP-B. Incluye capacidades para la detección de pérdida de secuencia o extravío de paquetes
    - ✓ NIVEL 3: Packet Level Protocol (PLP). Se encarga de:
      - Encaminamiento de paquetes (puede establecer hasta 4095 circuitos virtuales), control de flujo a nivel de red (i.e, extremo a extremo), establecimiento y liberación de SVC, secuenciamiento de los paquetes extremo a extremo, multiplexación etc...

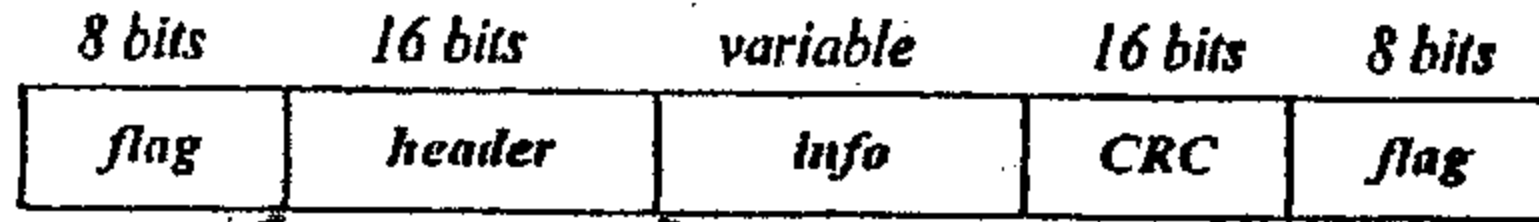
- REDES FRAME RELAY**
- ~~Red de conmutación de paquetes por circuito virtual~~
  - ~~Ofrecen un servicio orientado a conexión y no fiable~~
  - ~~Sopresión de control de errores en los nodos intermedios~~ -> esto es posible ya que los enlaces tienen una  $P_e$  baja -> ofrece  $v_m$  más elevadas que x.25 (2 Mbps aprox.)
  - ~~La configuración se realiza fuera de banda~~
  - Al igual que x.25, soporta SVC y PVC
  - Realiza una separación del Plano de Usuario y Plano de Control
  - *Simplificaciones en el Plano de Usuario.*
    - ✓ Suprime el Nivel 3 del plano de usuario. Pero como FR ofrece un servicio orientado a conexión, nos surge la siguiente pregunta: ¿Qué ocurre con el establecimiento y liberación de las llamadas? Pues que se lleva al plano de control del nivel 3 (Q.933). ¿Y con la función de multiplexión de conexiones? La función de multiplexión se pasa al nivel 2 en FR



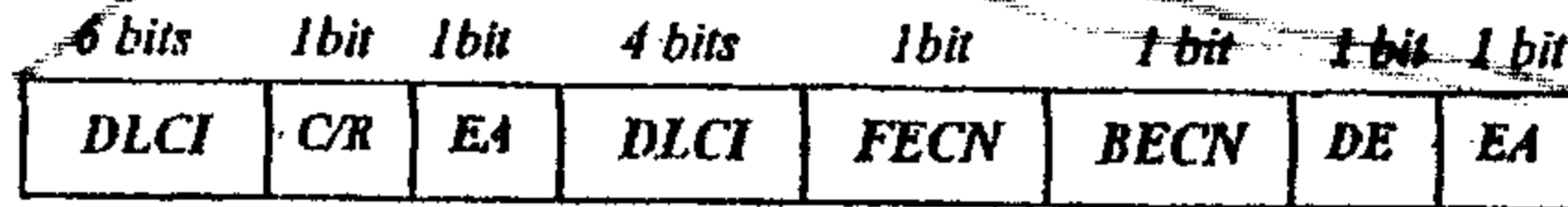
➤ **REDES FRAME RELAY (cont.)**

Formato de Trama (nos referimos al formato existente en el plano de usuario)

- no se establece una longitud máxima de trama, pero debe ser un múltiplo entero de bytes



Header: puede tener varios formatos (como en X.25), pero normalmente suele ser de 2 bytes



- **DLCI: Data Link Circuit Identifier.** Estos diez bits son el identificador de conexión de enlace de datos. Permite definir hasta 1024 circuitos virtuales; con el DLCI se identifica al canal lógico al que pertenece cada trama. Estos números de canal lógico se asignan por contratación
- **EA: Extensión de dirección.** Puesto que se permiten más de dos bytes de cabecera, estos bits indican si detrás siguen más bytes.
- **CONGESTIÓN:** en X.25 el control de congestión se realizaba mediante el *Control de Flujo* (se detienen fuentes cuando se detecta tráfico excesivo en algún punto del circuito virtual); en FR se usa el mecanismo de *notificación y descarte*.

*"Cuando se detecta una zona congestionada, se notifica al usuario que envía los datos que pasan por esa parte de la red, el cual disminuye la tasa de tráfico inyectado. Si el usuario no lo hace, la red descartará los datos que considere oportuno (aceptable, ya que F-R es un servicio no fiable)".*

La implementación de la técnica de *notificación y descarte* se realiza mediante los campos:

- ✓ **FECN (Forward Explicit Congestion Notification):** Notificación de congestión en el sentido de la transmisión.
- ✓ **BECN (Backward Explicit Congestion Notification):** Notificación de congestión en el sentido contrario a la transmisión.
- ✓ **DE (Discard Eligibility):** Las tramas que tienen este bit a "1" son susceptibles de descarte en situaciones de congestión.

**PROTOCOLOS TCP/IP** La familia de protocolos TCP/IP constituyen el armazón sobre el que se sitúa Internet. Está estructurado en 4 niveles. A la derecha vemos la equivalencia con el modelo OSI.

**CAPA DE ENLACE:** la base de este modelo de capas, es la capa de interface de red. Esta capa es la responsable de enviar y recibir frames los cuales son los paquetes de información que viajan en una red como una *'unidad simple'*. Algunos de los protocolos que podemos encontrar en este nivel son: PPP (*Point to Point Protocol*), LLC (*Logical Link Control*), HDLC (*High Level Data Link Control*) ...

**CAPA DE RED:** encapsula paquetes en datagramas y además ejecuta todos los algoritmos de enrutamiento (*routing*) de paquetes. Los protocolos de esta capa son:

TCP/IP	MODELO OSI
APLICACIÓN (protocolos que usan directamente aplicaciones de red)	APLICACIÓN PRESENT. SESIÓN
TRANSPORTE (transporte de datos extremo a extremo)	TRANSPORTE
RED (encaminamiento y paquetes)	RED
ENLACE (acceso a la red física)	ENLACE FÍSICO



- *Internet Protocol (IP)*: es el responsable del envío y enrutamiento de paquetes entre maquinas y redes.
- *Address Resolution Protocol (ARP)*: obtiene las direcciones de hardware de las maquinas situadas en la misma red física.
- *Internet Control Message Protocol (ICMP)*: envía mensajes e informa de errores en el envío de paquetes.
- *Internet Group Management Protocol (IGMP)*: se utiliza para la comunicación entre routers

**CAPA DE TRANSPORTE:** da el nivel de "sesión" en la comunicación; los dos protocolos posibles son:

- *Transmission Control Protocol (TCP)*: ofrece un servicio orientado a conexión; se utiliza en aplicaciones que requieren un "reconocimiento" (ACK) de los datos recibidos
- *User Datagram Protocol (UDP)*: proporciona conexión de comunicación y no garantiza la entrega de paquetes; la aplicación que lo utilice, es la responsable de la fiabilidad.

**CAPA DE APLICACIÓN:** es la capa que las aplicaciones utilizan para acceder a la red. Existen muchas utilidades y servicios en la capa de aplicación, por ejemplo: FTP, Telnet, SNMP y DNS.

**IP-INTERNET PROTOCOL**

- protocolo responsable del envío y enrutamiento de paquetes entre hosts.
- ofrece un servicio no orientado a conexión; i.e, no establece una sesión antes de intercambiar datos.
- ~~no fiable~~ a lo largo del camino, un paquete puede perderse, cambiarse de secuencia, duplicarse, retrasarse, o incluso trocearse.
- ~~no requiere un ACK~~ cuando se reciben los datos: esto es responsabilidad de una capa de un nivel más alto, como por ejemplo el TCP.
- reenvío de paquetes:

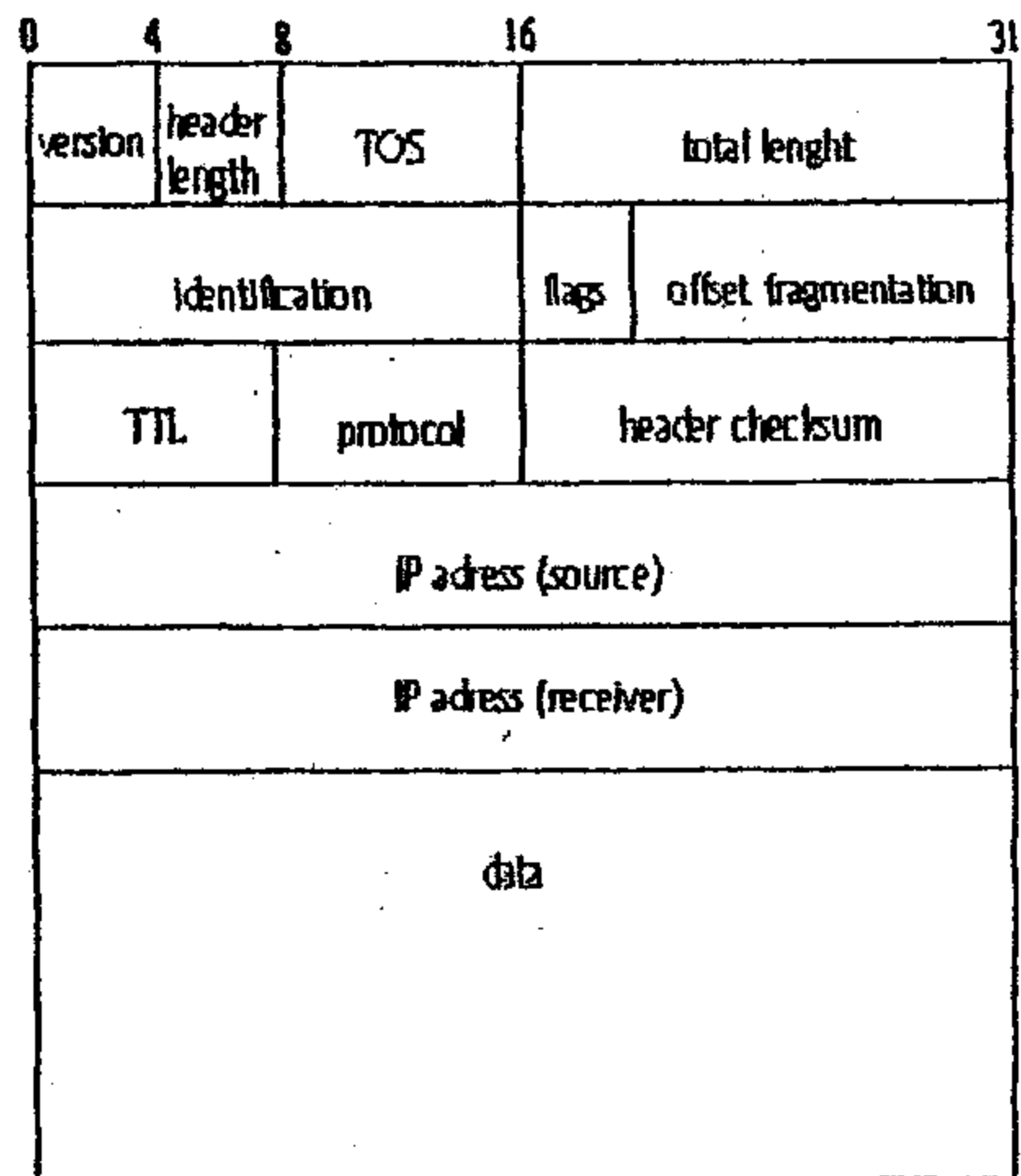
✓ ~~si el IP identifica una dirección de destino remoto, el IP envía el paquete directamente a la maquina.~~

✓ ~~si el destino es identificado como un destino 'remoto', el IP chequea sus tablas de rutas:~~

- si encuentra una ruta, el IP envía el paquete utilizando esa ruta
- si no encuentra una ruta, el IP envía el paquete al router por defecto

▪ formato de trama:

- .version: 4 bits para indicar la versión del IP.
- .header length: 4 bits que indican el numero de palabras de 32-bits; *min=20 bytes; max=30 bytes.*
- .TOS (*Type of Service*): 8 bits para indicar la calidad del servicio esperado por este datagrama.
- .total length: 16 bits, incluida la cabecera.
- .identification, flags y offset: relativos a la fragmentación de paquetes.
- .TTL(*Time To Live*): 8 bits, indican la cantidad de 'saltos' que le quedan al paquete IP.
- .protocol: 8 bits, indican el protocolo encapsulado en el campo de datos.
- .header checksum (o CRC): 16 bits, se recalcula en cada router debido a que el TTL varía.
- .IP adress (source,receiver): 32 bits para cada una
- .options: (no aparecen en el dibujo) información poco usual.



**~~ICMP, IGMP~~**

- ~~ICMP informa de errores y mensajes de control en nombre del IP~~, no pretende convertir al IP en un protocolo seguro y fiable, meramente comunica errores e informa de condiciones específicas. Los mensajes ICMP son enviados como datagramas IP y son por tanto inseguros en sí mismos.
- IP usa IGMP para informar a los routers que los hosts de un grupo específico están disponibles en una red; i.e., ~~es el equivalente a los mensajes ICMP pero entre los routers en vez de entre máquinas.~~

**~~TCP (TRANSFER CONTROL PROTOCOL)~~**

- ofrece un servicio de entrega ~~orientado a conexión, es un servicio totalmente fiable.~~
- ofrece un servicio extremo a extremo.
- los datos TCP ~~se transmiten en segmentos y se establece una sesión antes de que las máquinas puedan intercambiar datos.~~
- ~~se consigue la seguridad asignando un número de secuencia a cada segmento transmitido~~ por el TCP.
- la recepción de un ACK confirma la llegada correcta de un segmento a la otra máquina.
- si el emisor no recibe un ACK dentro de un periodo de tiempo ~~especificado~~, entonces el dato se vuelve a transmitir.
- tiene capacidad de multiplexación, ~~distintos entre múltiples destinos (puertos) dentro de una misma computadora~~
- el segmento de datos TCP suele ir encapsulado en el área de datos del datagrama IP
- formato de trama

- .puerto origen: puerto TCP de la máquina emisora
- .puerto destino: puerto TCP de la máquina destino
- .número de secuencia: posición del 1er byte tx
- .número de ACK: el número de secuencia del próximo byte que se espera recibir
- .len: longitud de la cabecera
- .res: reservado para uso futuro
- .bits de código: se trata de una serie de flags
- .ventana: espacio que queda disponible en la ventana
- .checksum: controla tanto la cabecera como los datos
- .puntero de datos urgentes: si se están enviando datos urgentes (especificados así en flags) este campo apunta al final de los datos urgentes en el segmento.

0		15 16		31	
puerto origen			puerto destino		
número de secuencia					
número de reconocimiento					
len	res	código	ventana		
checksum			puntero datos urgentes		
DATOS					

**~~UDP (USER DATAGRAM PROTOCOL)~~**

- ~~servicio de envío de datagramas sin garantía de entrega~~
- ofrece un servicio ~~NO~~ orientado a conexión -> la llegada al destino de un datagrama o la secuencia correcta de entrega no está garantizada.
- ~~se utiliza en aplicaciones que no requieren ACK~~; las aplicaciones que lo utilizan son típicamente las que transmiten pequeñas cantidades de datos a la vez.
- para utilizar UDP, una aplicación debe dar una dirección IP y un número de puerto de la aplicación destino.
- ~~los puertos UDP y los TCP son distintos~~ aún cuando algunos de ellos puedan tener el mismo número

**DIRECCIONAMIENTO IP**

- una dirección IP identifica la localización de un sistema en la red; esta dirección es única.
- una dirección IP está compuesta de dos partes:
  - ✓ ~~net\_id~~ : para identificar a la red
  - ✓ ~~host\_id~~ : para identificar al host dentro de esa red; todos los hosts pertenecientes a una misma red requieren el mismo net\_id

**IDENTIFICACIÓN DE RED E IDENTIFICACIÓN DE HOST**

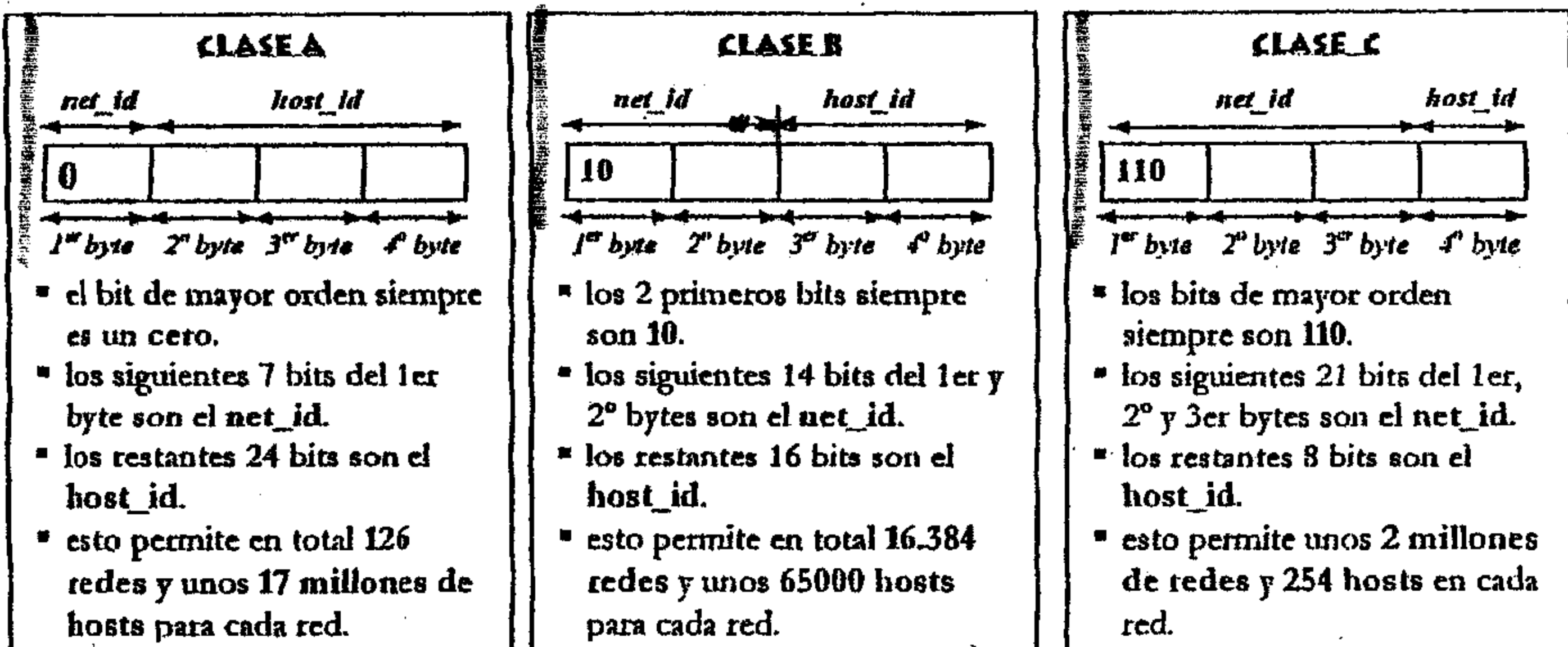
- existen 2 formatos para referirnos a una dirección IP: formato binario y formato decimal con puntos.
- ~~las direcciones IP están formadas por 4 campos de 8 bits -> 32 bits~~
- estos bytes están separados por puntos; cada byte representa un número decimal entre 0 y 255

Ej: 10011001 . 11011100 . 00110101 . 00001111 -> formato binario

153 . 220 . 53 . 15 -> formato decimal con puntos

**CLASES DE DIRECCIONES**

- hay 5 clases de direcciones IP (A, B, C, D, E) para poder acomodar redes de propósito distinto
- cada clase define que bits son usados para la red y cuales son usados para identificar el host
- cada clase se identifica por los primeros bits del primer byte

**PRINCIPIOS DE DIRECCIONAMIENTO**

Los siguientes principios se deben seguir para asignar unos net\_id y net\_host correctos.

- ✓ ~~el net\_id no debe ser 127~~ (esta identificación está reservada para funciones de diagnóstico)
- ✓ ~~host\_id no pueden estar todos a '1'~~ -> todos los bits a '1' es una dirección de broadcast
- ✓ ~~host\_id no pueden estar todos a '0'~~ -> todos los bits a '0' es la dirección de la red

**MASCARA DE RED Y DIRECCION IP**

- ~~una máscara de red (subnet mask) es una dirección de 32 bits usada para distinguir el net\_id del host\_id~~ -> esto es necesario para poder determinar cuando una dirección IP pertenece a la red local o a una red remota.
- cada *host* en una red TCP/IP requiere una máscara de red; distinguimos entre:
  - ✓ máscara de red por defecto : usada cuando una red no está dividida en subredes
  - ✓ máscara de red 'personalizada' : cuando la red está dividida en subredes (o segmentos)

### ► MASCARAS DE RED POR DEFECTO

- las máscaras por defecto que podemos utilizar, dependen de la 'clase' de dirección.
- ~~en la máscara de red, todos los bits que corresponden al net\_id están colocados a 1~~ (es útil recordar que el valor decimal correspondiente a todos unos es 255)
- ~~todos los bits que corresponden al host\_id estarán colocados a cero~~

Clase A ->	binaria:	11111111 00000000 00000000 00000000	decimal:	255.0.0.0
Clase B ->	binaria:	11111111 11111111 00000000 00000000	decimal:	255.255.0.0
Clase C ->	binaria:	11111111 11111111 11111111 00000000	decimal:	255.255.255.0

Ejemplo: Dirección IP :                      147.98.12.200 (de clase B)  
 Máscara :                                      255.255.0.0  
 net\_id :                                        147.98.0.0  
 dirección de broadcast:                  147.98.255.255

### ▪ Determinando el destino de un paquete:

- ✓ una suma binaria (AND) es el proceso interno que el IP utiliza para determinar cuando un paquete está destinado para un host local (en la propia red local) o remoto (en una red remota).
- ✓ cuando se inicializa el TCP/IP, la dirección IP del host es sumada con la máscara de red (1).
- ✓ antes de enviar un paquete, la dirección IP del destino es sumada con la misma máscara (2).
- ✓ si el resultado de (1) y (2) coincide, IP sabe que debe enviar el paquete a la red local.
- ✓ si el resultado no coincide el paquete será enviado a la dirección del router por defecto.

### SUBREDES

- segmento físico de una red que utiliza una dirección IP derivada del net\_id de la red
- al dividir una red en subredes, cada segmento debe tener un id de subred distinto
- beneficios al crear subredes:
  - ✓ mezclar diferentes topologías de red, como por ejemplo Ethernet y Token Ring.
  - ✓ superar limitaciones de las actuales tecnologías (#max terminales en un segmento).
  - ✓ reducir la congestión de red, redireccionando el tráfico y reduciendo el broadcasting.

### ► MASCARA DE SUBRED

- antes de implementar las subredes, básicamente debemos determinar:
  - ✓ número de segmentos físicos en nuestra red (i.e., el número de subredes).
  - ✓ número de direcciones hosts en cada segmento físico de la red.
- debemos tener en cuenta que cuantos más bits utilicemos en las máscaras de subred, más subredes estarán disponibles, pero menos hosts podrá tener cada subred.

### ► DEFINIENDO UNA MASCARA DE SUBRED

- una vez determinado el número de segmentos lo convertimos a formato binario.
- contamos el número de bits necesarios para representar este número.
- convertimos ese número de bits a formato decimal de izquierda a derecha.
- ✓ Ejemplo red de clase B : 147.98.0.0 -> máscara por defecto : 255.255.0.0
  - si necesitásemos, por ejemplo, 6 subredes, el valor binario es 110
  - para representar 6 en binario, requerimos 3 bits.
  - utilizamos los 3 primeros bits del host\_id como el id de subred -> 11100000
  - su valor decimal es 224 -> así, la máscara de subred es: 255.255.224.0

### ► DEFINIENDO IDENTIFICADORES DE SUBRED

Se usan los mismos bits utilizados para definir la máscara de subred:

- listamos todas las posibles combinaciones.
- eliminamos todos los valores que su contenido son todos ceros o unos.
- convertimos a decimal los valores para cada subred -> cada valor decimal representa una única subred; este valor será usado para definir el rango de hosts para esa subred.

Siguiendo con el ejemplo anterior:

- Recordemos que la máscara de subred era :

binario: 255 255 224 0

decimal: 11111111 11111111 11100000 00000000

- Si listamos todas las posibles combinaciones, las 6 subredes quedan definidas por:

3 <sup>er</sup> byte	id de subred
00000000 = 0	No válida
00100000 = 32	-> 147.98.32.0
01000000 = 64	-> 147.98.64.0
01100000 = 96	-> 147.98.96.0
10000000 = 128	-> 147.98.128.0
10100000 = 160	-> 147.98.160.0
11000000 = 192	-> 147.98.192.0
11100000 = 224	No válida

### ► DEFINIENDO ID DE HOSTS EN UNA SUBRED

Para determinar el número de hosts en una subred:

- calculamos el número de bits disponibles para el id de host.
- convertimos este número a decimal.
- le restamos 1

Siguiendo con el ejemplo anterior:

- teníamos una dirección de clase B, que usaba 16 bits para la net\_id y 3 bits para el id de subred; por lo tanto nos quedaban 13 bits para el id de host -> este valor en decimal es 8191; si le restamos 1 nos queda un total de 8.190 hosts por subred.

- y los rangos para cada subred son:

id de subred	rango de id's de host
00000000 = 0	No válida
00100000 = 32	x.y.32.1 - x.y.63.254
01000000 = 64	x.y.64.1 - x.y.95.254
01100000 = 96	x.y.96.1 - x.y.127.254
10000000 = 128	x.y.128.1 - x.y.159.254
10100000 = 160	x.y.160.1 - x.y.191.254
11000000 = 192	x.y.192.1 - x.y.223.254
11100000 = 224	No válida

## NIVEL DE ENLACE- MECANISMOS DE CONTROL DE FLUJO

- Según el modelo OSI el control de flujo se puede implementar en los niveles de enlace, de red y de transporte; aquí vamos a tratar de los mecanismos de control de flujo a nivel de enlace, por simplicidad y porque su implementación en los otros dos niveles sigue las mismas pautas básicas.
- Estos mecanismos se pueden dividir en dos bloques según su ámbito de aplicación:
  - ✓ control de flujo básico entre dos equipos (por *software* -Xon/Xoff -o por *hardware* -RTS/CTS)
  - ✓ control de flujo bajo protocolo
- A su vez los mecanismos de control de flujo bajo protocolo se pueden clasificar en:
  - FEC (*Forward Error Correction*)
  - ARQ (*Automatic Repeat Request*)

- Esta técnica usa una combinación de dos mecanismos fundamentales:

- ✓ acknowledgments (ACK)
- ✓ timeouts

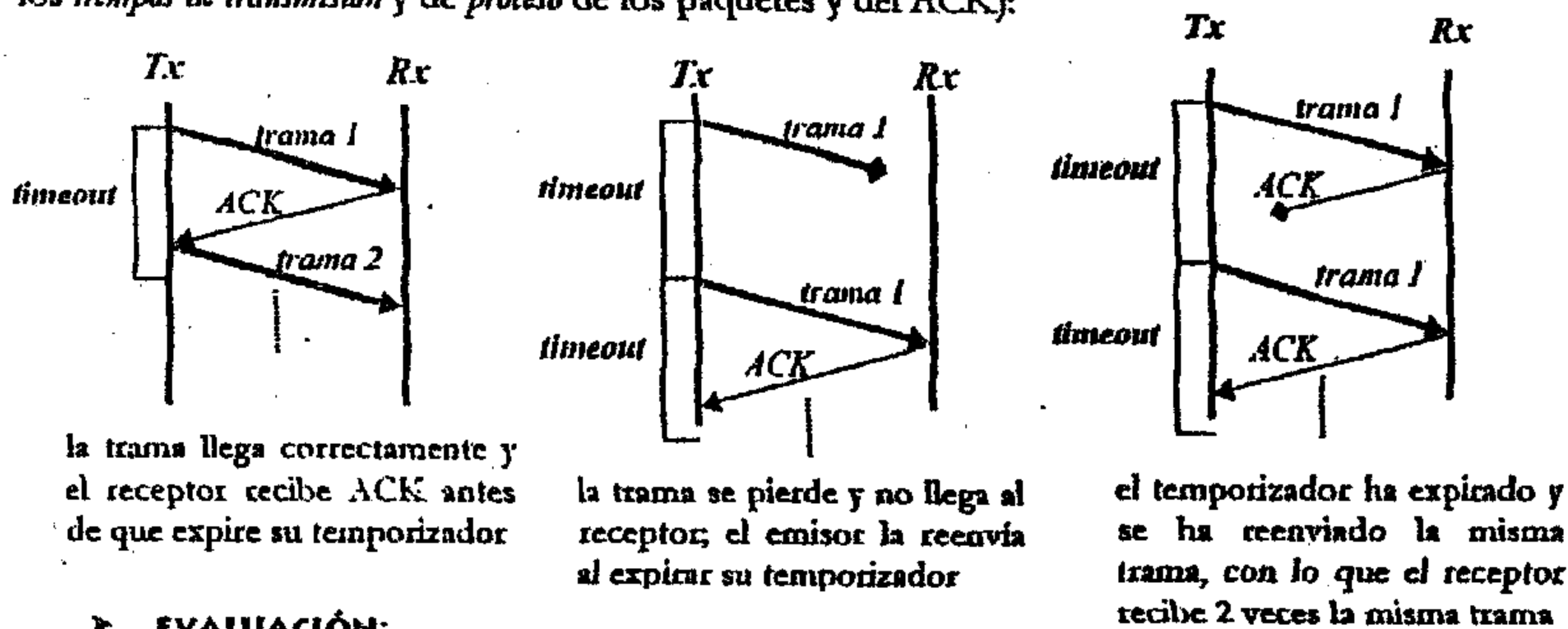
- Un ACK es una pequeña trama de control con que el receptor informa al emisor de que ha recibido la trama correctamente; algunos mecanismos tb utilizan NACKs para informar que la trama se ha recibido con error y para solicitar su reenvío; si el emisor no recibe un ACK/NACK durante un tiempo razonable (timeout) retransmite la trama.

- mediante ARQ se logra implementar tanto el *control de flujo* como la *recuperación de tramas* recibidas con error.

**TIPO 1: DE PARADA Y ESPERA (STOP & WAIT)**

- caso más simple de ARQ -> no consigue una buena eficiencia
- trabaja en modo half-duplex
- funcionamiento: el emisor, al mandar una trama, guarda una copia en un buffer y lanza un temporizador, quedándose a la espera de que le llegue un ACK o un NACK; luego:
  - ✓ si le llega un ACK -> libera el buffer de memoria y envía una nueva trama.
  - ✓ si le llega un NACK o expira el temporizador -> envía la trama que guardaba en el buffer.

En los siguientes esquemas se puede ver este funcionamiento (para simplificar se han supuesto nulos los tiempos de transmisión y de proceso de los paquetes y del ACK):



**EVALUACIÓN:**

- $L$ : longitud del paquete (en bits)
- $p_b$ : probabilidad de error de bit
- $p_e$ : probabilidad de paquete erróneo
- $t_{tx}$ : tiempo de tx de 1 paquete
- $t_{out}$ : 2 veces el  $t_{prop}$  → no es timeout
- $t_T = t_{tx} + t_{out}$
- $t_{proceso} = t_{tx}$  del ACK = 0
- $t_o$ : tiempo entre llegadas de 2 paquetes correctos

$$p_e = 1 - (1 - p_b)^L \approx L \cdot p_b$$

$$\bar{t}_v = \frac{t_T}{1 - p_e} \rightarrow \lambda_{max} = \frac{1}{\bar{t}_v} = \frac{1 - p_e}{t_T}$$

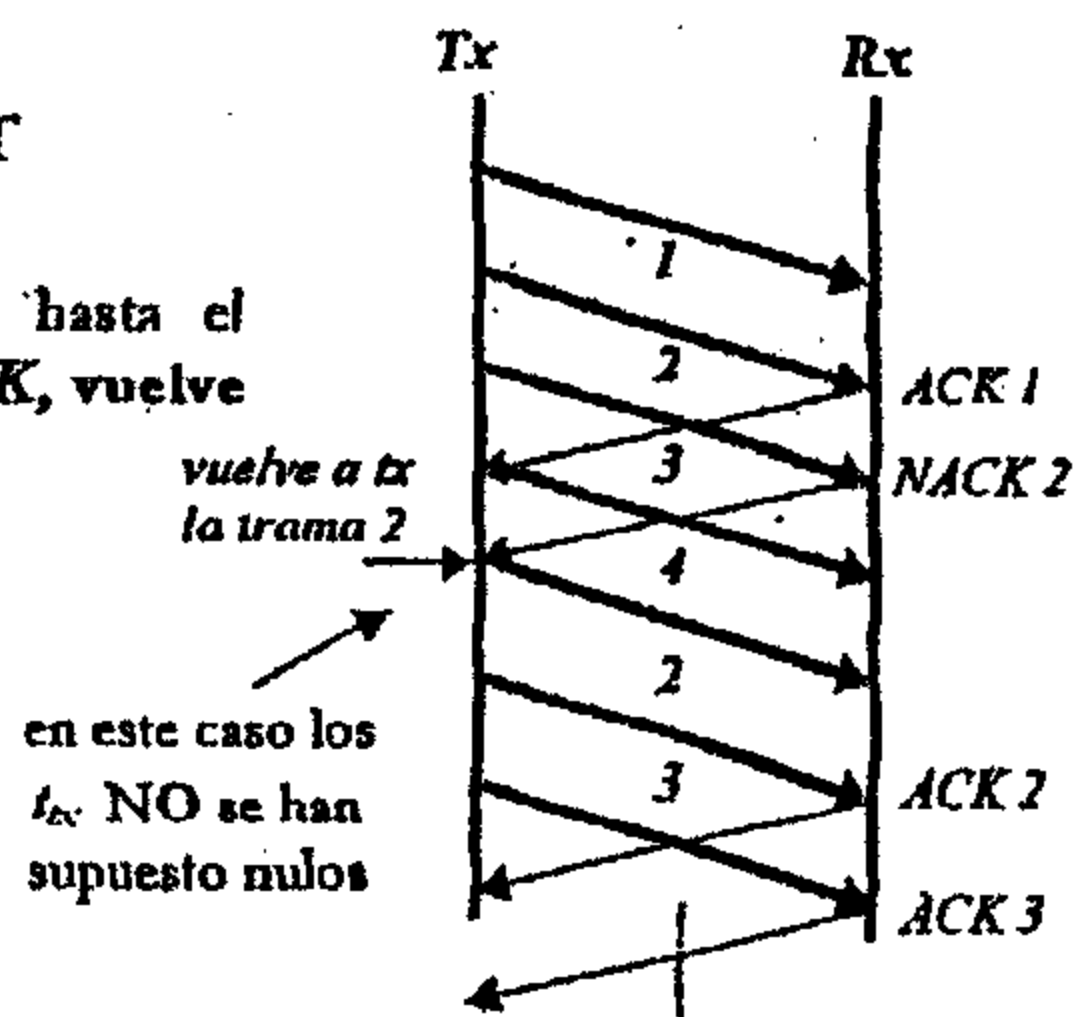
$$t_T = 2t_{prop} + t_{Tx}$$

**TIPO 2: GO-BACK-N**

- ventana deslizante -> mejor eficiencia que STOP&WAIT
- trabaja en modo full-duplex
- funcionamiento: el emisor va transmitiendo tramas hasta el tamaño de la ventana; si le llega un NACK de la trama  $K$ , vuelve a transmitir la trama  $K$  y TODAS las que le siguen
- es necesario información de secuencia
- también utiliza timeouts para los ACKs

✓ **EVALUACIÓN:**

$$\bar{t}_v = t_{tx} + \frac{p_e \cdot t_T}{1 - p_e}$$

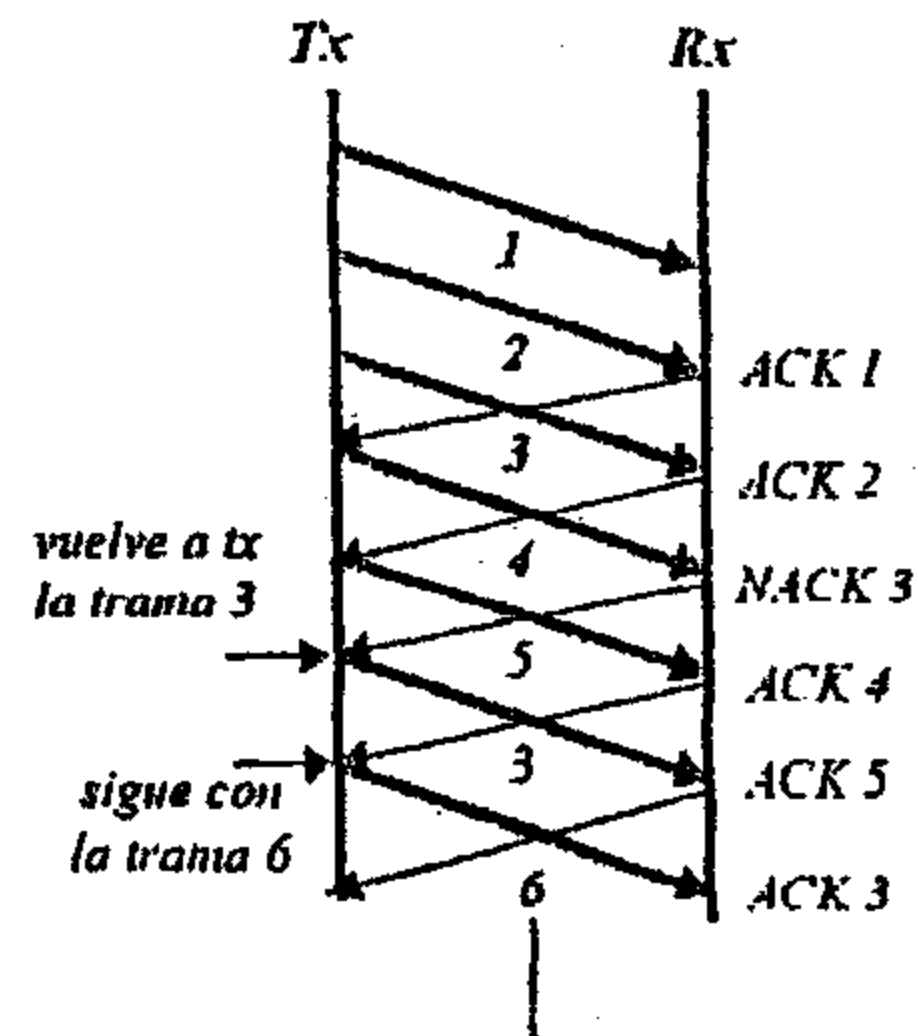


**TIPO 3: ARQ SELECTIVO**

- ventana deslizante, eficiencia similar al GO-BACK-N
- trabaja en modo full-duplex
- funcionamiento: el emisor va transmitiendo tramas hasta el tamaño de la ventana; si le llega un NACK de la trama  $K$ , vuelve a transmitir SOLO la trama  $K$
- es necesario información de secuencia
- también utiliza timeouts para los ACKs (al igual que en GO-BACK-N, si expira el timeout de la trama  $K$ , el emisor la reenvía))

✓ EVALUACIÓN:

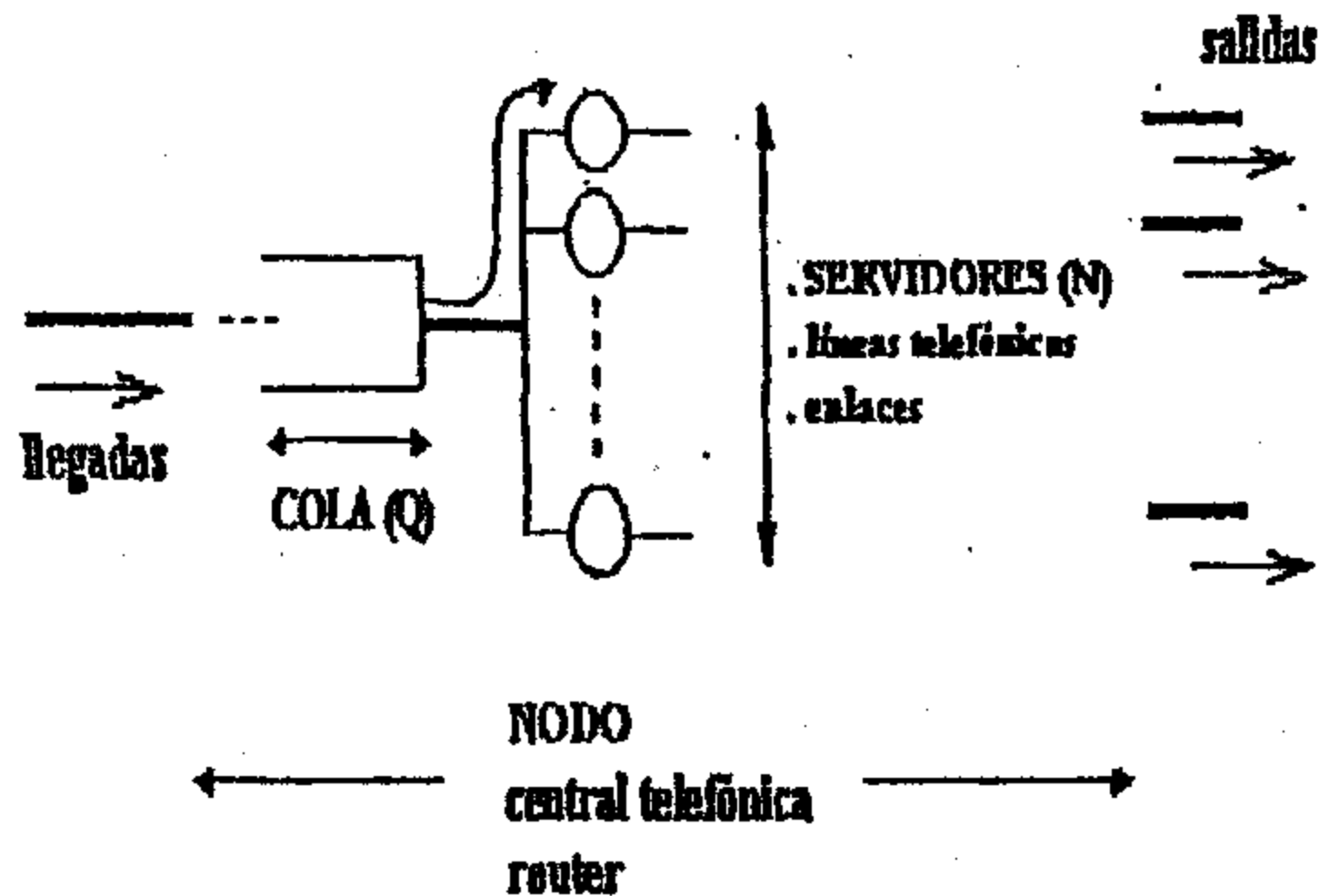
$$\bar{t}_r = \frac{t_{Tx}}{1 - p_e}$$



### TEMA 3. REDES CONMUTACIÓN DE CIRCUITOS

#### DIMENSIONADO DE SISTEMAS

**TEORÍA DE COLAS** Es el estudio matemático del comportamiento de líneas de espera; la figura de la derecha representa el tipo de sistemas con los que trataremos (Q puede ser 0). El principal problema es que el ritmo al que llegan las llamadas (tasa de llegada) y el tiempo que se tarda en servir cada llamada (tiempo de servicio) es algo arbitrario y aleatorio. Se deduce de aquí que los sistemas de líneas de espera son sistemas probabilísticos y que por tanto su estudio se realizará mediante métodos probabilísticos. Debido a esta naturaleza aleatoria se suele trabajar con valores promedio y con probabilidades.



El estudio de las líneas de espera lo clasificaremos en base a:

1. la distribución de las entradas (ley de probabilidad de los intervalos entre llegadas)
2. la distribución del tiempo de servicio (ley de prob. del tiempo que se tarda en servir una llegada)
3. el número de servidores y el tamaño de la cola

**LLEGADAS AL SISTEMA** Los conceptos importantes a considerar son:

- proceso probabilístico: consideraremos solo llegadas en régimen de Poisson de tasa  $\lambda$
- Poisson: se considera una población infinita que genera llegadas a una tasa  $\lambda$  agregada FIJA
- $\lambda$  (llegadas/segundos): ~~tasa media de llegadas~~
- $1/\lambda$  (segundos/llegada): ~~tiempo medio entre llegadas~~

Este proceso de llegadas cumple lo siguiente:

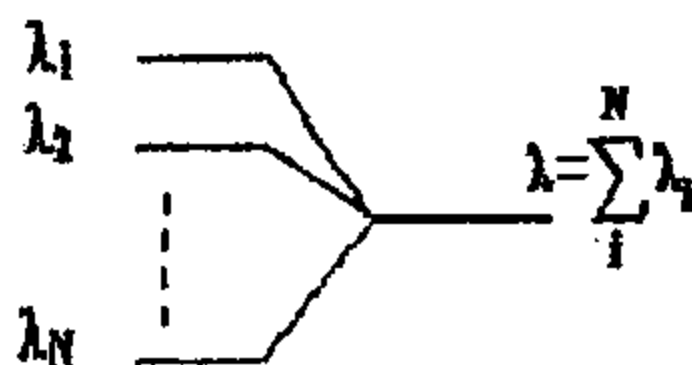
el tiempo entre llegadas está distribuido exponencialmente (i.e., tiene una función de densidad del tipo):  

$$f_r(t) = \lambda \cdot e^{-\lambda \cdot t}$$

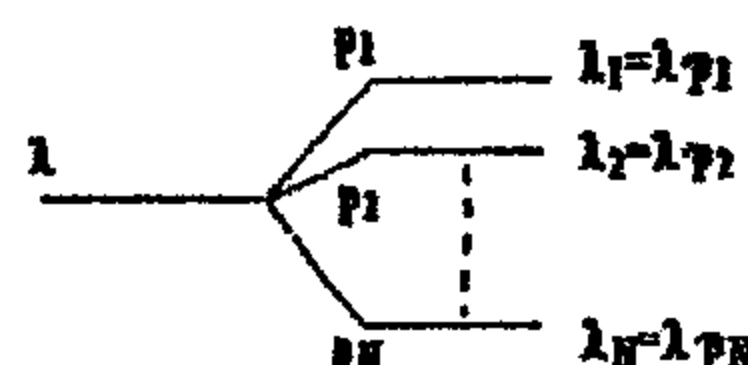
$$P\{X = K \text{ en } T\} = e^{-\lambda T} \frac{(\lambda T)^k}{k!}$$

indica la ~~probabilidad de que en un intervalo de tiempo T hayan llegado al sistema K llamadas~~

#### ADICIÓN DE FUENTES INDEPENDIENTES



#### DISTRIBUCIÓN ALEATORIA



**TIEMPO DE SERVICIO** No todas las llamadas tienen la misma duración o todos los paquetes de datos son del mismo tamaño. Los conceptos importantes a considerar son:

- proceso probabilístico: el tiempo de servicio es una variable aleatoria exponencial de tasa  $\mu$
- $1/\mu$  (seg/llamada): tiempo medio de servicio
- $\mu$  (llamadas/seg): tasa de finalización de llamada

$$f_{servicio}(t) = \mu \cdot e^{-\mu t}$$

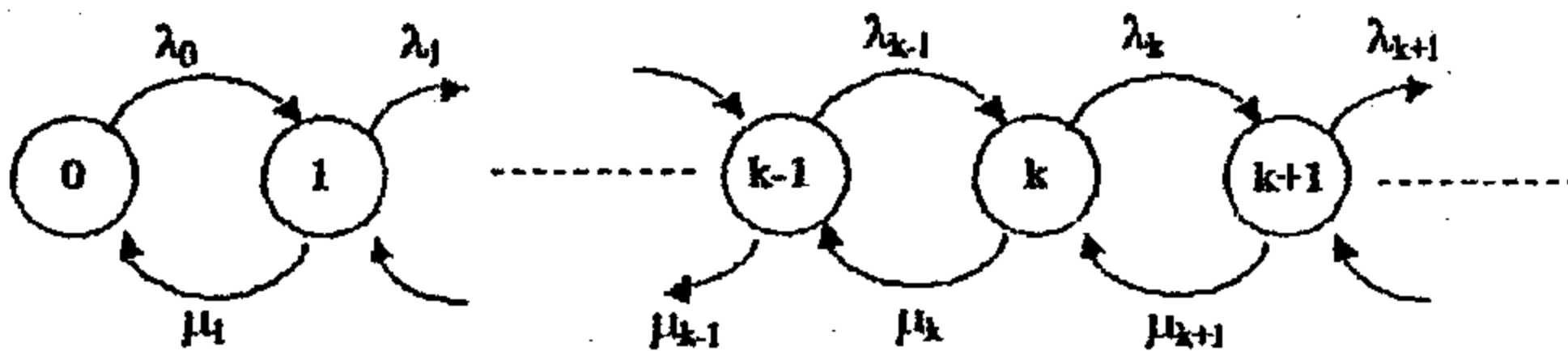
función de densidad de probabilidad del tiempo de servicio de una llamada en 1 servidor

**CADENAS DE MARKOV DE PARÁMETRO CONTINUO**

- procesos estocásticos cuyo desarrollo se puede considerar como una serie de transacciones entre valores determinados, llamados estados del proceso.
- parámetro continuo implica que las transacciones se pueden dar en cualquier instante de tiempo.
- son procesos sin memoria; es decir, el tiempo de permanencia en cada estado es exponencial.

**CADENAS DE MARKOV DE NACIMIENTO Y MUERTE**

- caso particular de cadenas de Markov.
- característica principal : las transacciones sólo pueden darse entre estados adyacentes.
- nosotros trabajaremos siempre en **REGIMEN PERMANENTE (RP)**
- para representar las cadenas se utilizan los Diagramas de Estados



- $k$  : estado en el que se encuentra el sistema; es decir la  $K$  nos estará indicando el número de elementos (llamadas, paquetes, etc.) que hay en el sistema (cola + servidor).
- $p_k$  : probabilidad de estar en el estado  $K$ ; multiplicado por 100 nos da el % de tiempo que el sistema está en el estado  $K$ . Se cumple  $\sum p_k = 1$
- $\lambda_k$  : tasa de llegadas cuando el sistema está en el estado  $K$
- $\mu_k$  : tasa de finalizaciones cuando el sistema está en el estado  $K$ . Es fácil ver que  $\mu_0=0$ , puesto que en el estado 0 no hay ningún elemento siendo atendido.
- $\lambda_k p_k$  : porcentaje de llegadas respecto del total, cuando el sistema está en el estado  $K$
- $\mu_k p_k$  : porcentaje de finalizaciones cuando el sistema está en el estado  $K$

Una cadena de Markov en RP cumple la siguiente ecuación ( IMPORTANTE)

$$p_k \cdot (\lambda_k + \mu_k) = p_{k-1} \cdot \lambda_{k-1} + p_{k+1} \cdot \mu_{k+1}$$

**ECUACIÓN DE EQUILIBRIO GLOBAL**  
(flujo saliente = flujo entrante)

La mayoría de las cadenas de Markov que trataremos se resuelven con esta ecuación y aplicando :

$$\sum_K p_k = 1$$

**TIEMPO DE PERMANENCIA EN UN ESTADO K** Consideremos una variable aleatoria continua ( $\tau$ ) que indique el tiempo que el sistema está en un estado  $K$ . Se cumple que su fd es:

donde  $q_{k,j}$  indica la tasa de transición del estado  $k$  al estado  $j$

$$f_{\tau}(t) = \sum_{j \neq k} q_{k,j} \cdot e^{-\sum_{j \neq k} q_{k,j} \cdot t}$$

tiempo medio de permanencia en un estado  $K \rightarrow \frac{1}{\sum_{j \neq k} q_{k,j}}$

**NOTACIÓN DE KENDALL** Utiliza una notación de la forma  $F_T/F_S/N/K/P$  donde:

- $F_T$ , representa la función de distribución de los tiempos entre llegadas.
- $F_S$ , representa la función de distribución de los tiempos de servicio. Pueden ser:
  - D: determinista, tiempo entre llegadas o de servicio constante
  - M: markoviano, tiempo entre llegadas o de servicio distribuido exponencialmente
  - G: general, para una distribución general de los tiempos de servicio
- $N$ , número de servidores en el sistema
- $K$ , número total de elementos (servidores más cola) en el sistema
- $P$ , población.

Si  $K$  y/o  $P$  no aparecen, significa que son infinitos.

**PASTA (Poisson Arrivals Time Average)**

$$R_k \triangleq \frac{\# \text{ veces que, al llegar al sistema, lo encontramos en el estado } K}{\# \text{ veces total que llegamos al sistema}}$$

Para un proceso de llegadas markovianas  $\rightarrow$

$$R_k = \frac{\lambda_k}{\lambda_{OF}} \cdot P_k$$

con  $\lambda_{OF} \triangleq \sum_k \lambda_k \cdot P_k$   
tasa media ofrecida al sistema

En el caso particular de llegadas de Poisson, tenemos que  $\lambda_k = \lambda \quad \forall K$ , y por lo tanto:

PARA LLEGADAS DE POISSON  $\rightarrow$

$$\lambda_{OF} = \sum_k \lambda_k \cdot P_k = \lambda \sum_k P_k = \lambda$$

$$R_k = \frac{\lambda_k}{\lambda_{OF}} \cdot P_k = P_k$$

**RELACIÓN DE LITTLE**

Primero hay que definir cuál será el sistema sobre el que aplicaremos Little. Normalmente se tratará de el sistema entero (cola + servidores), el subsistema de los servidores o el subsistema cola. Luego se aplica la siguiente relación:

$$\bar{N}_{SIST} = \lambda_{IN} \cdot \bar{W}_{SIST}$$

tiempo medio de permanencia de un elemento en el sistema

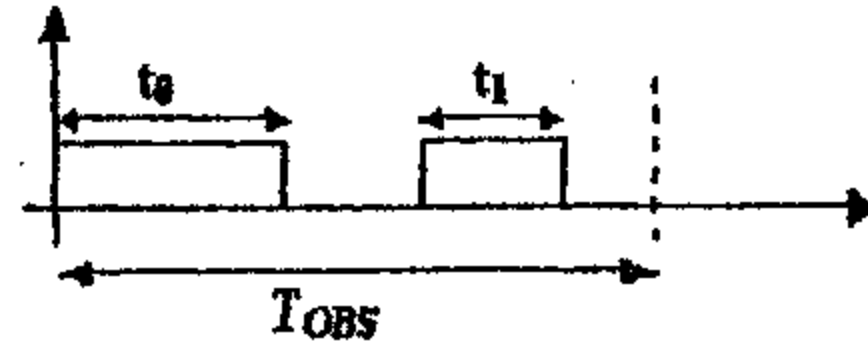
número medio de elementos en el sistema

tasa de entrada al sistema

**IMPORTANTE** : Solo se cuenta el tráfico que entra o pasa por el sistema; el tráfico que no puede entrar (pq el sistema ya está lleno) no se contabiliza en la tasa de entrada

**VOLUMEN DE TRÁFICO (en 1 servidor o circuito) :**

$$V_r = t_0 + t_1 \text{ (suma de los tiempos de ocupación)}$$



**INTENSIDAD DE TRÁFICO (en 1 servidor o circuito) :**

$$I_r = \frac{V_r}{T_{OBS}} \text{ (Erlangs) es la probabilidad de ocupación}$$

$$I_r = \lambda \cdot \frac{1}{\mu} (Er)$$

Estas dos definiciones también son aplicables a un grupo de servidores. De la segunda sacamos varias definiciones importantes:

**TRÁFICO CURSADO (TC):** es el número medio de servidores ocupados ( $\bar{N}_{serv}$ ); es lo que arriba habíamos llamado  $I_{r,TOTAL}$  y es por tanto lo que realmente sirve el sistema.

$$TC = \lambda_{curtido} \cdot \frac{1}{\mu} (Er) = \bar{N}_{serv} = \sum_k k \cdot p_k$$

**TRÁFICO DEMORADO (TD)** lo que tiene que esperar en cola para ser cursado; si el tamaño de la cola es 0 tendremos  $TD=0$  (nota: a lo que se sirve directamente, sin tiempo de espera, se le llama tráfico no demorado,  $TC_{noD}$ )

$$TD = \lambda_{demorado} \cdot \frac{1}{\mu} (Er)$$

**TRÁFICO OFERTADO:** es lo que realmente genera la población; sería el número de recursos (servidores y cola) ocupados si el sistema tuviera infinitos recursos.

$$TO = \lambda_{ofertado} \cdot \frac{1}{\mu} (Er)$$

**TRÁFICO PERDIDO (TP)** el que no puede ser cursado por falta de recursos;  $TP=0$  cuando el número de servidores y/o el tamaño de la cola es infinito.

$$TP = \lambda_{perdido} \cdot \frac{1}{\mu} (Er)$$

**CONSIDERACIONES :**

1. en las expresiones anteriores lo importante es calcular la  $\lambda$  de forma correcta
2.  $TD = TO - TC$  (o  $\lambda_r = \lambda_{er} - \lambda_c$ ) para cualquier sistema
3.  $TC = TD + TC_{noD}$  (o  $\lambda_c = \lambda_r + \lambda_{noD}$ ) para cualquier sistema

**GRADO DE SERVICIO** Se puede ver de dos maneras :

**CONGESTIÓN EN TIEMPO:** se mira desde el punto de vista del sistema; es la probabilidad de que todos los servidores del sistema estén ocupados. Se le llama probabilidad de bloqueo (PB).

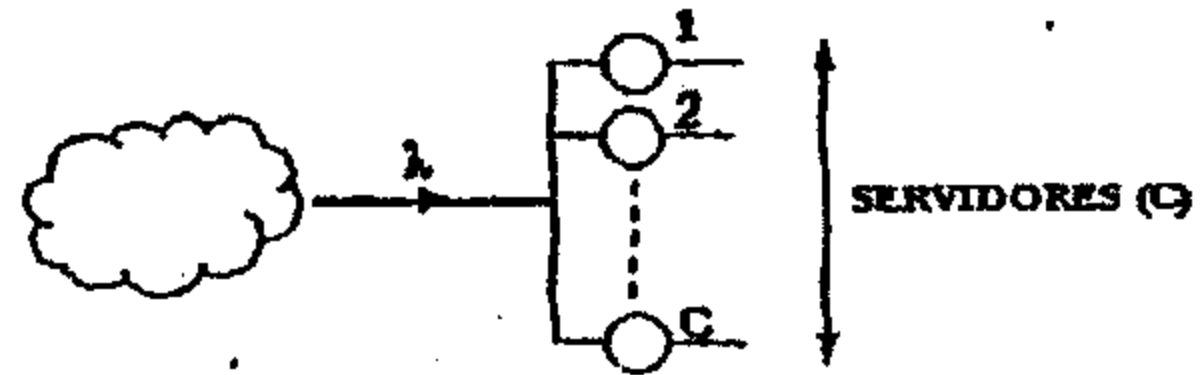
**CONGESTIÓN EN LLAMADAS:** se mira desde el punto de vista del usuario; nos podemos encontrar con la PP y/o PD (Importante: usaremos las  $R_k$  para calcularlas):

- probabilidad de pérdida (PP): es la probabilidad de que cuando una llamada llega al sistema se pierda; es decir, que al llegar encuentre el sistema en un estado que haga que se pierda.
- probabilidad de demora (PD): es la probabilidad de que cuando una llamada llega al sistema se demore; es decir, que al llegar encuentre el sistema en un estado que haga que se tenga que esperar en cola.

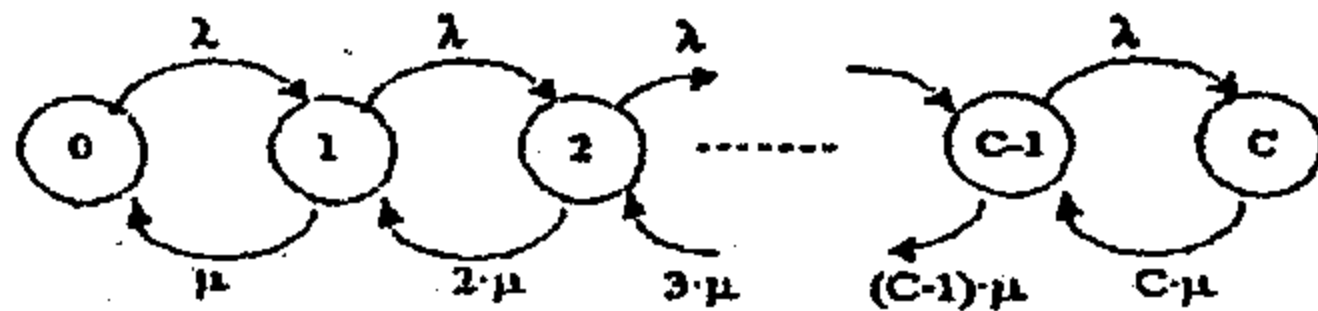
**EJEMPLO,** en un sistema con 3 servidores ( $N=3$ ) y un tamaño de cola de 2 elementos ( $Q=2$ ). la PB sería la probabilidad de que el sistema se encontrara en los estados 3, 4 o 5. la PP sería la probabilidad de que la llegada encontrara al sistema en el estado 5 y la PD sería la probabilidad de que la llegada encontrara al sistema en los estados 3 o 4.

**SISTEMA DE PERDIDAS PURO M/M/C/C (ERLANG-B)**

Llegadas son markovianas, en particular de Poisson (es decir, se supone una población infinita generando llamadas con tasa agregada  $\lambda$  que es fija para cualquier estado), C servidores y cola de tamaño cero.



La cadena de Markov que modela a este sistema será:



$$p_k = \frac{A^k}{k!} \cdot p_0 \quad p_0 = \frac{1}{1 + A + \frac{A^2}{2} + \frac{A^3}{3!} + \dots + \frac{A^C}{C!}} = \frac{1}{\sum_{k=0}^C \frac{A^k}{k!}} \quad \text{con } A = \frac{\lambda}{\mu}$$

$\frac{A^C}{C! \cdot \sum_{k=0}^C \frac{A^k}{k!}}$  está tabulado en tablas de Erlang

$\lambda_{OF} = \sum_k \lambda_k \cdot p_k = \lambda \cdot \sum_{k=0}^C p_k = \lambda$

*se cuenta la  $\lambda$  del estado C porque la población, que no sabe si el sistema está totalmente ocupado, sigue ofreciendo tasa independientemente del estado*

$\lambda_{OF} = \underbrace{\lambda \cdot p_0 + \lambda \cdot p_1 + \lambda \cdot p_2 + \dots + \lambda \cdot p_{C-1}}_{\lambda_{cursada}} + \underbrace{\lambda \cdot p_C}_{\lambda_{perdida}}$

*es decir, la  $\lambda$  total ofrecida al sistema es la  $\lambda$  que llega estando en el estado 0, más la  $\lambda$  que llega estando en el estado 1, ... así hasta el estado C; de todas éstas, las llamadas que lleguen y encuentren el sistema en el estado C se perderán.*

$\lambda_{cursada} = \lambda \cdot p_0 + \lambda \cdot p_1 + \dots + \lambda \cdot p_{C-1} = \mu \cdot p_1 + 2\mu \cdot p_2 + \dots + C\mu \cdot p_C$

puesto que estamos en R.P. todo lo que llega al sistema debe finalizar; esta manera de calcular la  $\lambda_{cursada}$  puede resultar muy útil en determinadas situaciones

$\frac{\lambda_{cursada}}{\mu} = \frac{1}{\mu} (\mu \cdot p_1 + 2\mu \cdot p_2 + \dots + C\mu \cdot p_C) = p_1 + 2 \cdot p_2 + \dots + C \cdot p_C = \bar{N}_{serv} = \bar{N}_{SIST}$

Se cumplen las relaciones

$$\begin{cases} TC = TO - TP \\ TP = TO \cdot p_c \\ TC = TO \cdot (1 - p_c) \end{cases}$$

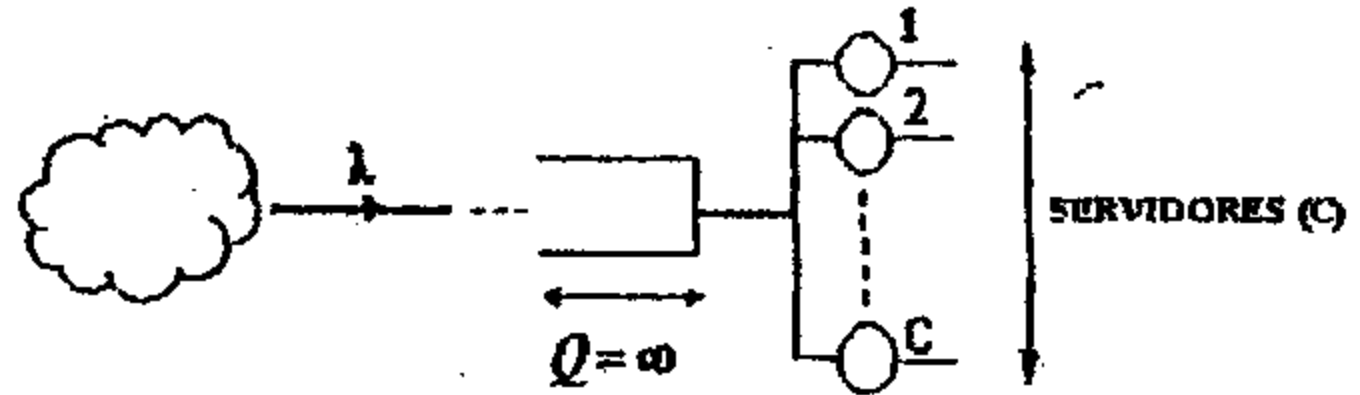
como no hay cola, el número medio de servidores ocupados es el número medio de elementos en el sistema

$PP = R_C = \frac{\lambda_c}{\lambda_{OF}} \cdot p_c = \frac{\lambda}{\lambda} \cdot p_c = p_c$

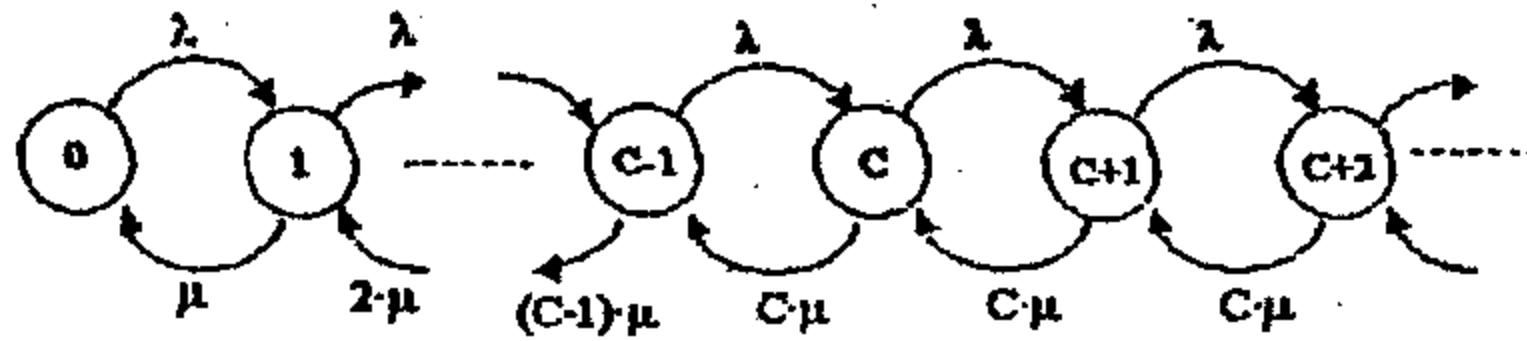
en un sistema de pérdidas puro con llegadas de Poisson se cumple que :  
CONG TIEMPO = CONG. LLAMADAS

**SISTEMA DE ESPERA PURO M/M/C (ERLANG-C)**

Mirando el M/M/C sacamos que las llegadas son markovianas (en particular de Poisson, es decir, se supone una población infinita generando llamadas con tasa agregada  $\lambda$ ), con C servidores y una cola infinita.



La cadena de Markov que modela a este sistema será:



$$p_k = \begin{cases} \frac{A^k}{k!} \cdot p_0 & \text{si } k \leq C \\ \left(\frac{A}{C}\right)^{k-C} \cdot p_C & \text{si } k > C \end{cases} \quad \text{con } A = \frac{\lambda}{\mu}$$

$$p_0 = \frac{1}{\sum_{k=0}^C \frac{A^k}{k!} + \frac{A^C}{C!} \cdot \sum_{k=1}^{\infty} \left(\frac{A}{C}\right)^k}$$

para que esto converga se ha de cumplir:

$$\frac{A}{C} < 1$$

$$\lambda_{OF} = \sum_k \lambda_k \cdot p_k = \lambda$$

$$\lambda_{curada} = \lambda \cdot p_0 + \lambda \cdot p_1 + \dots = \mu \cdot p_1 + 2\mu \cdot p_2 + \dots + C\mu \cdot p_C + C\mu \cdot p_{C+1} + C\mu \cdot p_{C+2} + \dots$$

$$\bar{N}_{serv} = \sum_{k=1}^C k \cdot p_k + C \cdot \sum_{k=C+1}^{\infty} p_k = p_1 + 2 \cdot p_2 + \dots + C \cdot p_C + C \cdot p_{C+1} + C \cdot p_{C+2} + \dots = TC$$

Puesto que la cola es infinita este es un sistema sin pérdidas y por tanto  $TP=0 \Rightarrow \frac{TD - TC}{\mu} = A$  Arriba hemos visto  $A < C$ ; ahora vemos lo que esto significa: el TO debe ser menor que el número de servidores; si esto no se cumple, la cola no pararía de llenarse y el sistema sería inestable.

Lo que sí que hay es espera en la cola; se cumple :

$$PD = R_C + R_{C+1} + \dots = p_C + p_{C+1} + \dots = PB$$

PASTA

en un sistema de espera puro con llegadas de Poisson se cumple que :  
CONG TIEMPO = CONG. LLAMADAS

$$PD = PB = \sum_{k=C}^{\infty} p_k = \sum_{k=C}^{\infty} \left(\frac{A}{C}\right)^{k-C} \cdot p_C = p_C \cdot \sum_{j=0}^{\infty} \left(\frac{A}{C}\right)^j = \frac{p_C}{1 - A/C} = B_0(A, C)$$

$$\bar{W}_q = \frac{\bar{N}_q}{\lambda} = \frac{1}{\mu C} \cdot \frac{1}{1 - A/C} \cdot PD$$

## MULTIPLEXACIÓN. JERARQUÍAS DIGITALES

enlace: es el elemento físico de transmisión que une dos nodos de la red.

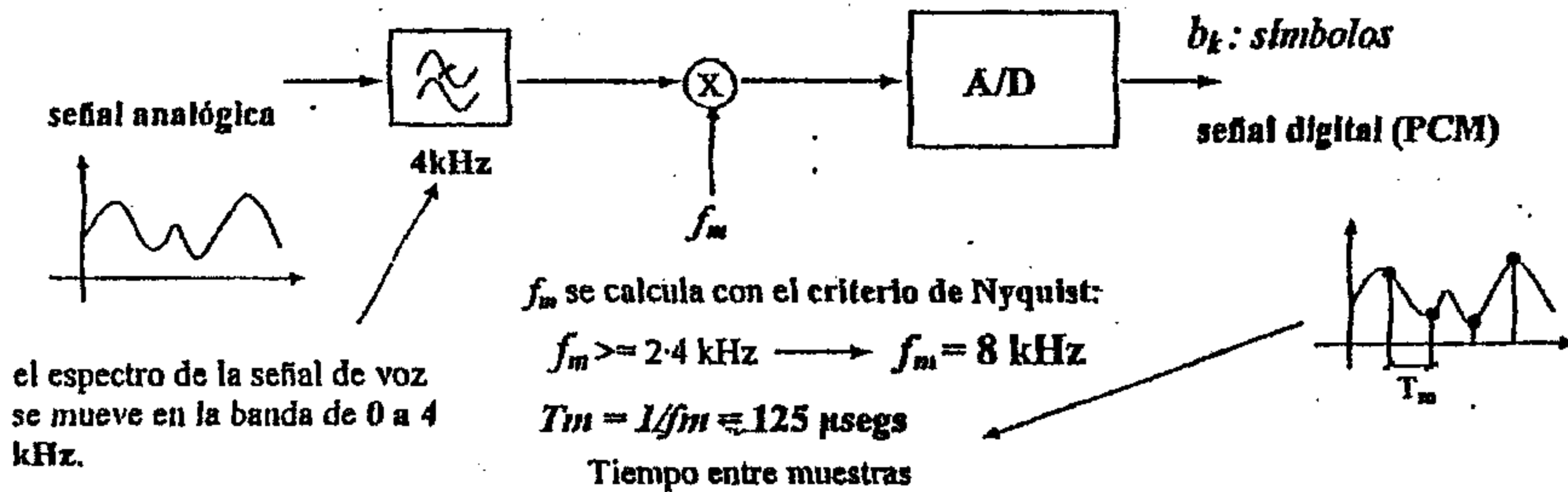
canal: es una fracción de la capacidad del enlace, un enlace tiene pues varios canales.

Una de las técnicas para acceder a un medio compartido es la multiplexación; aquí vamos a estudiar la TDM (*Time Division Multiple*). Las características TDM son:

- ✓ es propio de las señales digitales (i.e., trabajaremos con bps)
- ✓ cada usuario puede transmitir por el enlace durante un espacio de tiempo limitado (slot)
- ✓ durante su slot el usuario utiliza todo el ancho de banda del enlace
- ✓ ningún otro usuario puede transmitir sobre un slot que ya esté ocupado
- ✓ la información de los usuarios se transmite de manera secuencial
- ✓ la multiplexación puede hacerse a NIVEL DE BIT o a NIVEL DE MUESTRA

### PCM (PULSE CODE MODULATION). SEÑAL DE VOZ

PCM también conocida como MIC (*Modulación de Impulsos Codificados*), es un esquema digital para transmitir señales analógicas que permite digitalizar cualquier tipo de formato de datos analógico (video, voz, música, etc.). Aquí nos centraremos en la señal de voz. Para obtener una señal PCM a partir de la señal analógica de voz se emplea el siguiente esquema:

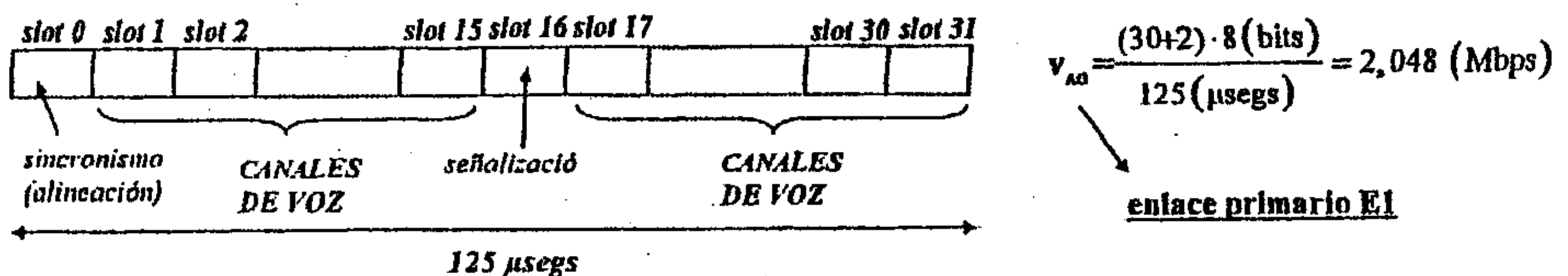


Para determinar cuántos bps salen de conversor analógico/digital (A/D), solo nos queda por saber el número de bits ( $N_q$ ) con el que cuantificamos cada muestra. La señal de voz se cuantifica con 8 bits por muestra. lo que nos da una velocidad de transmisión de:

$$v_{AD} = 8(\text{bits}) / 125(\mu\text{segs}) = 64 \text{ kbps} \rightarrow \text{velocidad de transmisión de voz PCM}$$

### PORTADORA E1

E1 también conocido como MIC 30+2 consiste en una agrupación de 30 canales de voz. La multiplexación es a NIVEL DE BYTE, esto quiere decir que el MUX va cogiendo grupos de 8 bits de cada uno de los 30 canales de entrada. El formato de una trama E1 es el siguiente:



**SEÑALIZACIÓN EN LA TRAMA E1**

Se realiza por **CANAL ASOCIADO (CAS)** fuera del slot. ~~Se utiliza el slot 16 de cada trama, y cada trama transmitida se ocupa de señalar dos canales: de los 8 bits del slot se utilizan 4 para un canal y 4 para otro; puesto que hay 30 canales, se necesitan en total 15 tramas.~~ Esto comporta la aparición de lo que se llama **MULTITRAMA**: una agrupación de 16 tramas, la primera de las cuales (llamada **MFAS**) se ocupa de la ~~señalización~~.



trama n°	slot 16
0	0 0 0 0 * * * *
1	señ. canal 1    señ. canal 17
2	señ. canal 2    señ. canal 18
⋮	⋮
15	señ. canal 15    señ. canal 31

**MFAS** : Multi Frame Alignment Signal  
(el slot 16 de la primera de las 16 tramas, sirve para alinear la multitrama; sus 8 bits se reparten : los 4 primeros son cero y los 4 segundos estan reservados)

Y la velocidad de señalización:

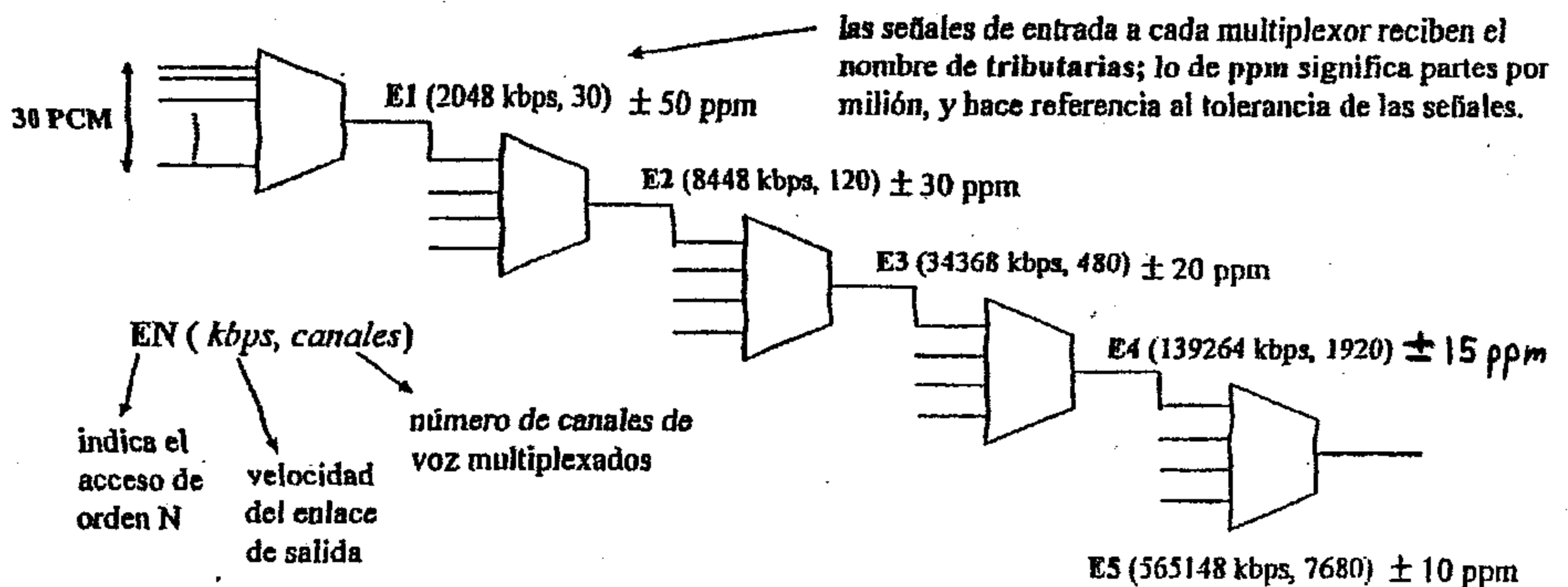
$$\frac{4 \text{ (bits)}}{16 \cdot 125 \text{ (}\mu\text{segs)}} = \dots$$

**JERARQUÍAS DIGITALES**

Aparte de la portadora E1 también existe otra denominada T1. El número y la manera en la que estas portadoras se agrupan dando lugar a velocidades de tx mayores está estandarizado. Esta estandarización da lugar a lo que se conoce con el nombre de jerarquías digitales. Existen dos grandes grupos de jerarquía digital: la *jerarquía digital pleióncrona* y la *jerarquía digital síncrona* (nosotros solo vamos a ver la primera)

> **Jerarquía Digital Pleióncrona (JDP, PDH)**

- tecnología basada en el transporte de canales digitales sobre un mismo enlace.
- los canales a multiplexar se unen formando tramas o módulos de nivel superior



▪ el multiplexado se realiza a nivel de bit (excepto en el primer nivel -del 30 PCM al E1- en el que el multiplexado se realizaba a nivel de byte)

▪ ~~los tramas PDH se estructuran en los siguientes campos:~~

~~FAW (Frame Alignment Word): bits dedicados al sincronismo de tramas~~

~~INFO: bits de las tramas BI (importante: cualquier tipo de bit de la trama E1)~~

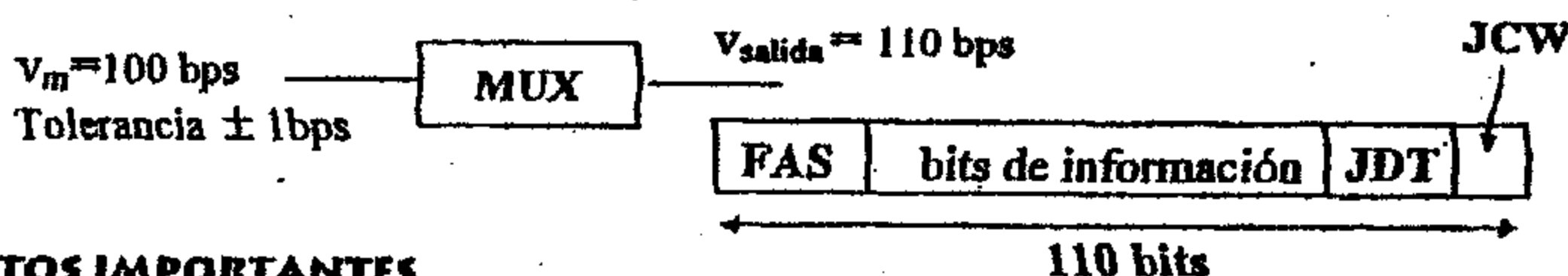
~~JDT: bits de justificación (en ocasiones es necesario rellenar la trama para sincronizar velocidades).~~

~~JCW (Justification Control Word): bits que sirven para indicar si se han utilizado bits de justificación~~

▪ su principal problemática es la falta de sincronismo entre equipos; para solucionar el problema se utilizan los denominados *bits de relleno o de justificación (JDT)*. A continuación vamos a ver como se utilizan

### BITS DE JUSTIFICACIÓN (JDT)

Consideremos la siguiente situación (aunque solo multiplexamos una tributaria, el estudio es análogo si se multiplexan varias tributarias)



#### PUNTOS IMPORTANTES

1. la tolerancia siempre se refiere a velocidades, nunca a tiempos.
2. la tolerancia no significa que la velocidad sea variable; debe quedar claro que la velocidad es fija;
3. los bits de justificación se utilizan en bloque: o se usan todos o no se utiliza ninguno.

Teniendo en cuenta que en la trama de salida tenemos 99 bits en el campo de INFO y 2 bits de JDT, y que el  $t_{AG} = 110(\text{bits}) / 110(\text{bps}) = 1 \text{ segundo}$ , vamos a ver como se utilizan los JDT:

**CASO 1** Velocidad tributaria: 99 bps. 0% de tramas justificadas.

**CASO 2** Velocidad tributaria: 101 bps. 100% de tramas justificadas.

**CASO 3** Velocidad tributaria: 100 bps. 50% de tramas justificadas

#### FORMULAS IMPORTANTES:

1.- el tiempo que tarda en transmitirse la trama por el canal agregado ( $T_{AG}$ ) es:

$$T_{AG} = \frac{(FAW + INFO + JDT + JWC) (\text{bits})}{v_{AG} (\text{bps})}$$

los bits de justificación ocupan un espacio fijo en la trama, y que en ese espacio siempre van a ir 1's o 0's; lo que sucede es que en ocasiones estos JDT llevan información útil y en otras no.

2.- la velocidad a la que se transmiten los bits de la tributaria por el canal agregado ( $v_{trib,AG}$ ); tenemos dos casos :

CASO 1 : no se utilizan los bits de justificación  $\rightarrow v_{trib,AG} = \frac{INFO}{T_{AG}}$

CASO 2 : se utilizan los bits de justificación  $\rightarrow v_{trib,AG} = \frac{INFO + JDT}{T_{AG}}$

3.- se deben cumplir las dos siguientes relaciones

$$V_{trib,AG} \geq V_{trib,max} \rightarrow$$

(utilizando los JDT)

si la desigualdad fuera al revés, estaríamos imponiendo que el canal agregado (aunque utilizara los bits de JDT) no fuera lo suficientemente rápido como para tx todos los bits que la tributaria le está dando

$$V_{trib,AG} \leq V_{trib,min} \rightarrow$$

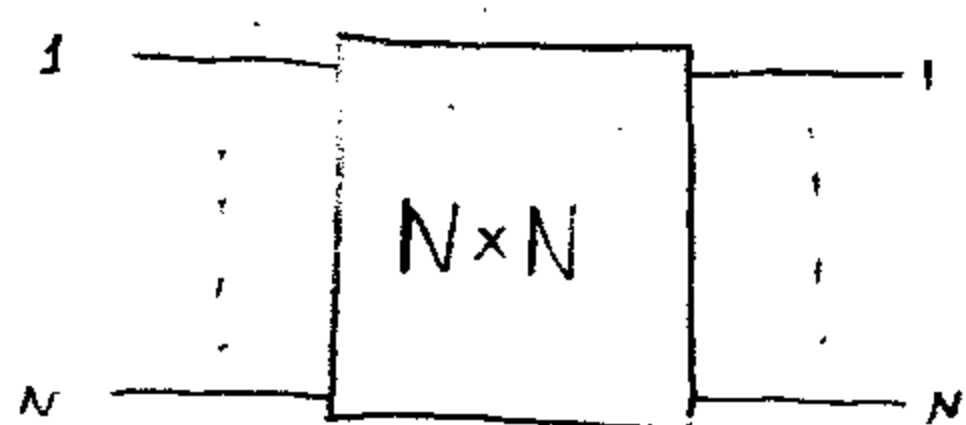
(sin utilizar los JDT)

si la desigualdad fuera al revés, estaríamos imponiendo que el campo INFO en la trama del canal agregado fuera demasiado grande; i.e., cuando el canal agregado fuera a tx la trama, el campo INFO no estaría lleno

4.- siendo f el % (/100) de tramas justificadas, se cumple la siguiente relación:

$$v_{trib} = \frac{(1-f) \cdot INFO + f \cdot (INFO + JDT) \text{ (bits)}}{T_{AG} \text{ (segs)}}$$

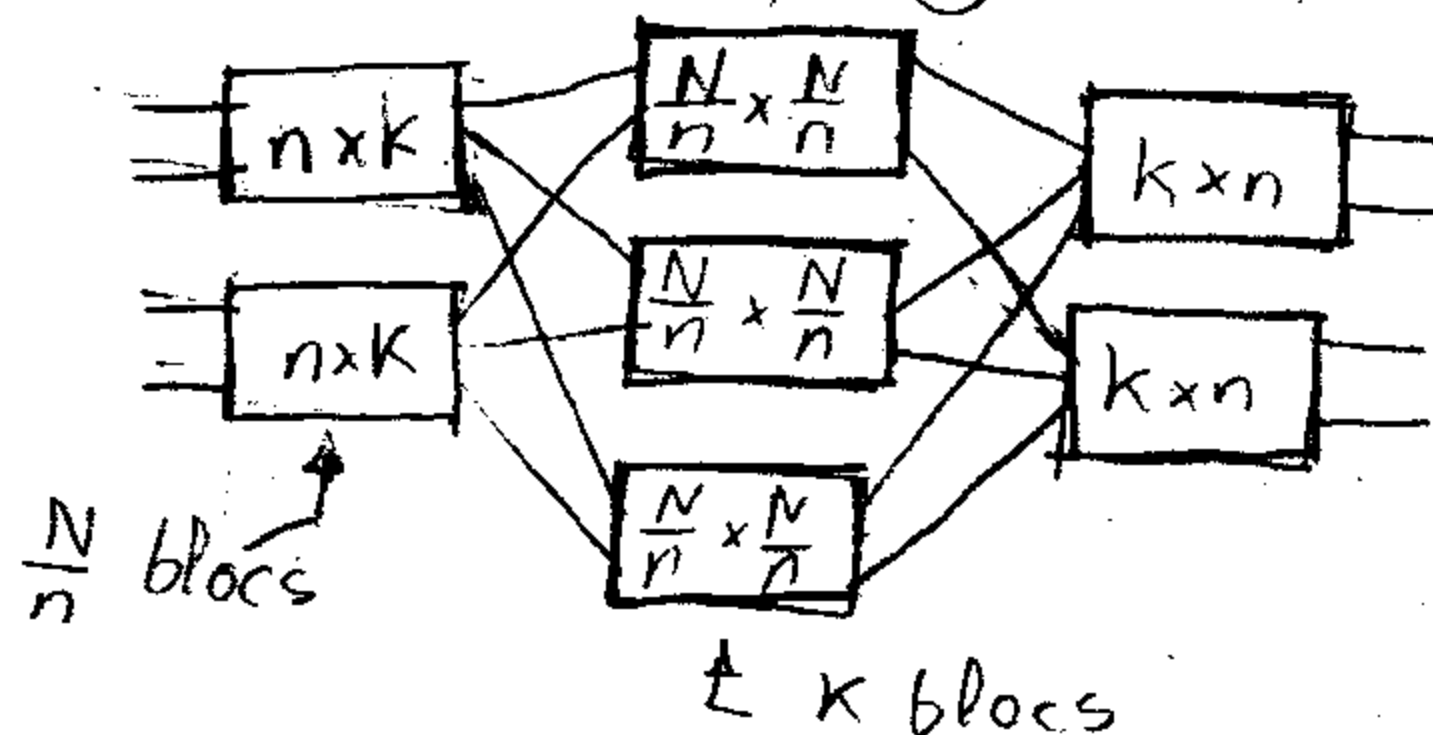
### XARXES D'INTERCONEXIÓ



Totalment malleades  
# Creuaments  $C = N^2$

#### - Xarxes de Clos

• No tenen bloqueig intern



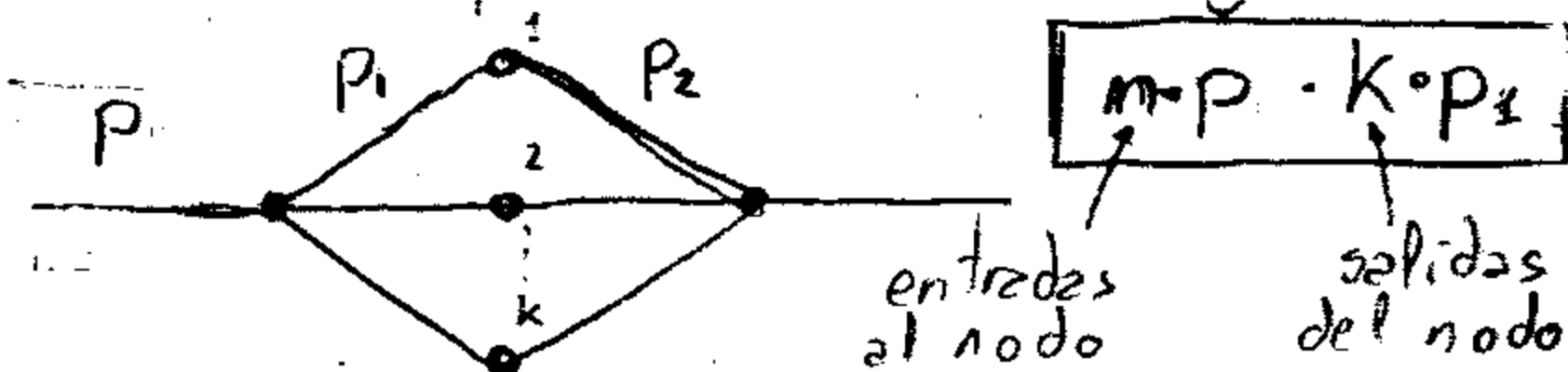
$$k \geq 2n - 1$$

$$C = 4\sqrt{2} N^{3/2}$$

$$n_{opt} = \sqrt{N/2}$$

#### - XARXES AMB BLOQUEIG

Dibuixem la ruta que pot seguir una comunicació si a l'exemple anterior hi haguessin bloqueig:



$$m \cdot p \cdot k \cdot p_2$$

$$2p_1 = 3p_2$$

$$p = \frac{\text{Traf. Oferido}}{N}$$

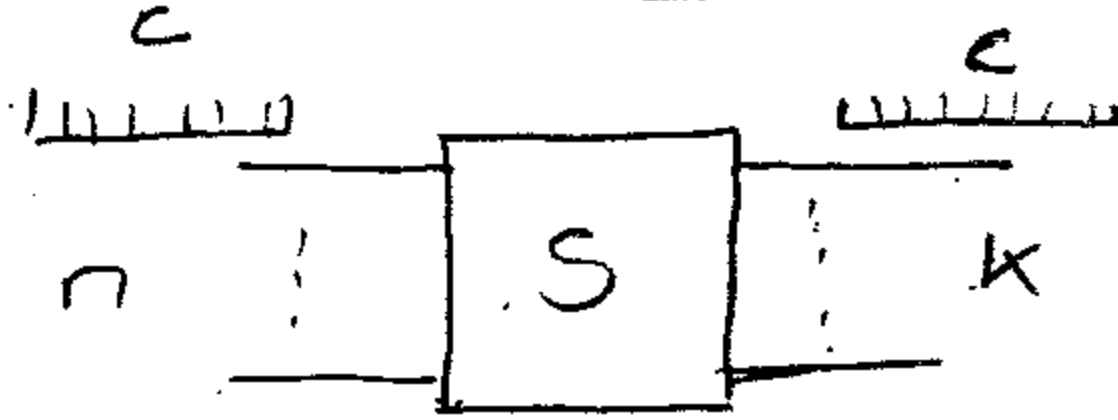
$$PB[\text{diamond}] = PB[\text{line}]^3 = (1 - P \cdot PB[\text{line}])^3 = (1 - (1 - p_1)(1 - p_2))^3$$

# COMUTACIÓ DIGITAL

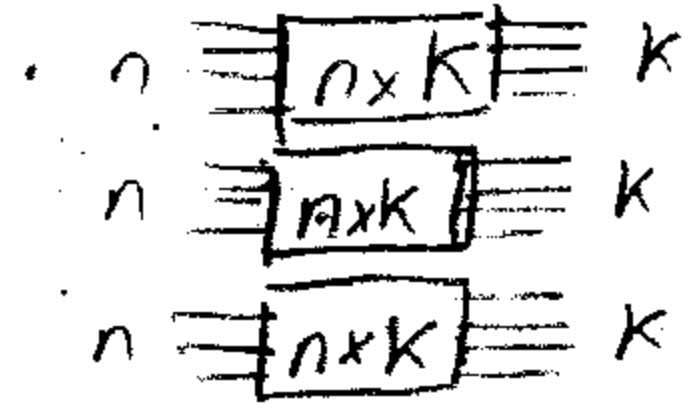
• Equivalent analògic



≡



≡



↑  $c$  blocs