# Estructura de redes de computadores

# Estructura de redes de computadores

Jordi Íñigo Griera
José María Barceló Ordinas
Llorenç Cerdà Alabern
Enric Peig Olivé
Jaume Abella i Fuentes
Guiomar Corral i Torruella

EDITORIAL UOC

Diseño de la colección: Editorial UOC

Primera edición en lengua castellana: Diciembre 2008

© Jordi Íñigo Griera, José María Barceló Orinas, Llorenç Cerdà Alabern
Enric Peig Olivé, Jaume Abella i Fuentes, Guiomar Corral i Torruella, del texto
© Imagen de la cubierta: Istockphoto
© Editorial UOC, de esta edición
Rambla del Poblenou 156, 08018 Barcelona
www.editorialuoc.com

Realización editorial: Carrera Edició, S.L.
Impresión:

ISBN: 978-84-9029-433-8
Depósito legal B.

*La UOC queda facultada expresamente por el/la autor/a para digitalizar y publicar la Obra en un repositorio en línea que será accesible al público bajo licencias Creative Commons, incluyendo la licencia* ***R****econocimientoNoComercial-SinObraDerivada* ***(B****Y****-****NC-ND****)****, v.****4.0*** *Internacional* ***(****jurisdicción internacional****)****, que permitirá copiar, distribuir y transmitir públicamente la Obra siempre citando la autoría y la fuente, sin hacer un uso comercial y sin hacer obra derivada. Si la Obra es transformada, la obra generada estará sometida a una licencia Creative Commons similar o compatible con la licencia mencionada.*

**Jordi Íñigo Griera**
Ingeniero de Telecomunicación por la Universitat Politècnica de Catalunya. Actualmente es director de desarrollo de software en Safelayer Secure Communications, S. A. y profesor en la Universitat Politècnica de Catalunya.

**José María Barceló Orinas**
Doctor ingeniero de Telecomunicación por la Universitat Politècnica de Catalunya. Es profesor titular en el Departamento de Arquitectura de Computadores en la UPC y miembro del Área de Redes de Computadores en la Facultad de Informática de la UPC.

**Llorenç Cerdà Alabern**
Doctor ingeniero de Telecomunicación e ingeniero técnico industrial por la Universitat Politècnica de Catalunya. Es profesor titular en el Departamento de Arquitectura de Computadores en la UPC y miembro del Área de Redes de Computadores en la Facultad de Informática de la UPC.

**Enric Peig Olivé**
Ingeniero de Telecomunicación por la Universitat Politècnica de Catalunya y doctor por la Universitat Pompeu Fabra. Es profesor de la Escuela Superior Politécnica de la Universitat Pompeu Fabra.

**Jaume Abella i Fuentes**
Ingeniero electrónico e ingeniero técnico de Telecomunicaciones por Ingeniería y Arquitectura de La Salle-Univeridad Ramon Llull. Actualmente trabaja en sistemas de seguridad en redes informátricas. Profesor de Redes de comunicaciones y de CCNA y CCNP de CISCO en el *Networking Academy Program* de CISCO Systems. Director del máster en Seguridad de ingeniería y arquitectura de La Salle-Universidad Ramon Llull.

**Guiomar Corral i Torruella**
Ingeniera electrónica e ingeniera técnica de Telecomunicaciones por Ingeniería y Arquitectura de La Salle-Universidad Ramon Llull. Máster en Seguridad en tecnologías de la Información de Ingeniería y Arquitectura de La Salle-Universidad Ramon Llull. Actualmente trabaja en sistemas de seguridad en redes informáticas. Profesora de Redes de ordenadores, Redes de comunicaciones y de CCNA y CCNP en el Netmorking Academy Program de CISCO Systems.

# Índice

**Capítulo I. Conceptos básicos de redes de computadores**
*Jordi Íñigo Griera, Enric Peig Olivé*

**Introducción** ..... 13
**1. Breve historia de las comunicaciones** ..... 14
1.1. El teléfono ..... 14
1.2. Aparecen los primeros ordenadores ..... 20
1.3. Arquitecturas de protocolos ..... 24
1.4. La digitalización de la red telefónica ..... 27
1.5. La banda ancha ..... 29
1.6. La telefonía móvil ..... 30
**2. Las redes de gran alcance** ..... 31
2.1. Conmutación de circuitos ..... 31
2.2. Datagramas y circuitos virtuales ..... 37
2.3. Comparación entre conmutación de circuitos y conmutación de paquetes ..... 39
2.4. Encaminamiento ..... 41
**3. Arquitecturas de protocolos: el modelo OSI** ..... 50
3.1. Definición ..... 50
3.2. Los protocolos ..... 51
3.3. Los servicios ..... 53
3.4. Los siete niveles del modelo OSI ..... 53

**Capítulo II. Transmisión de datos**
*Llorenç Cerdà Alabern, Jordi Íñigo Griera*

**Introducción** ..... 59
**1. Conceptos básicos de transmisión de datos** ..... 59
1.1. Modelo sencillo de comunicaciones ..... 62

**2. Interfaz DTE-DCE** ..... 65
2.1. RS-232 ..... 65
**3. Circuito de datos** ..... 68
3.1. Circuitos unidireccionales y bidireccionales ..... 68
3.2. Velocidad de transmisión y velocidad de modulación ..... 69
3.3. Tipos de codificaciones de canal ..... 73
3.4. Cálculo de la probabilidad de error ..... 86
3.5. Multiplexación ..... 88
**4. Medio de transmisión** ..... 92
4.1. Perturbaciones de la señal ..... 93
4.2. Tipo de medios de transmisión ..... 111

## Capítulo III. Enlace de datos

*José M. Barceló Ordinas, Llorenç Cerdà Alabern*

**Introducción** ..... 125
**1. Descripción del nivel de enlace** ..... 126
1.1. Montaje de las tramas ..... 127
1.2. Sincronismo de trama ..... 127
**2. Detección de errores** ..... 131
2.1. Distancia de Hamming ..... 133
2.2. Capacidad de detección de una ráfaga de error ..... 134
2.3. Probabilidad de que una combinación arbitraria de bits sea aceptada como palabra válida ..... 134
2.4. Control de paridad ..... 135
2.5. Códigos de paridad longitudinal y transversal ..... 137
2.6. Códigos detectores de errores CRC ..... 140
2.7. Códigos correctores de errores ..... 142
**3. Control de errores** ..... 144
3.1. Diagramas de tiempo ..... 145
3.2. Evaluación de los protocolos: eficiencia ..... 148
3.3. Protocolo *Stop & Wai*t ..... 149
3.4. Protocolos de transmisión continua ..... 157
**4. Control de flujo** ..... 162
4.1. Protocolos de ventana ..... 163

4.2. Dimensionado de la lista de transmisión ... 164
4.3. Ventana de recepción ... 165
4.4. Ventana óptima ... 166
4.5. Dimensionamiento del campo de los identificadores de trama ... 167
**5. Otras consideraciones sobre el nivel de enlace** ... 169
5.1. *Piggybacking* ... 169
5.2. Sincronismo de protocolo ... 170
5.3. Importancia del nivel de enlace según el contexto ... 172
5.4. Multiplexación en el nivel de enlace ... 174
**6. Ejemplo: HDLC y ATM** ... 176
6.1. HDLC ... 176
6.2. ATM ... 179

**Capítulo IV. Tecnologías de red de área local**

*José M. Barceló Ordinas, Jordi Íñigo Griera, Enric Peig Olivé, Jaume Abella i Fuentes, Guiomar Corral i Torruella*

**Introducción** ... 183
**1. Conceptos básicos** ... 184
1.1. Arquitectura de protocolos en las LAN ... 186
**2. Topologías** ... 188
2.1. Topología en estrella ... 188
2.2. Topología en bus ... 189
2.3. Topología en anillo ... 190
2.4. Topologías lógicas y topologías físicas ... 191
2.5. Pseudotopología sin hilo ... 194
**3. Control de acceso al medio** ... 196
3.1. Control de acceso al medio estático ... 197
3.2. Control de acceso al medio dinámico ... 203
3.3. Control de acceso al medio aleatorio ... 205
3.4. Acceso múltiple por escucha de portadora con eliminación de colisiones ... 220
**4. Commutación en las LAN: dominios de colisión** ... 224
**5. Estándares para redes de área local** ... 230
5.1. Ethernet y IEEE802.3 ... 232

5.2. Fast Ethernet y Gigabit Ethernet ..... 237
5.3. Cableado EIA/TIA 568 ..... 246
5.4. Redes inalámbricas: norma IEEE802.11 ..... 251
**6. Nuevas prestaciones de las redes Ethernet** ..... 256
6.1. Ethernet full-duplex ..... 256
6.2. Control de flujo ..... 257
6.3. Sistemas multiestándar: autonegociación ..... 258
**7. Redes de área local virtuales: VLAN** ..... 259
7.1. Características de las VLAN ..... 260
7.2. Tipos de VLAN ..... 263
7.3. IEEE 802.1Q ..... 265

**Capítulo V. Redes de gran alcance: Internet**

*José M. Barceló Ordinas, Jordi Íñigo Griera, Enric Peig Olivé*

**Introducción** ..... 269
**1. Estructura de protocolos en Internet** ..... 270
1.1. Protocolos de Internet ..... 272
1.2. Encapsulamiento ..... 272
**2. El IP (*Internet protocol*)** ..... 274
2.1. Direcciones IP ..... 275
2.2. Subredes ..... 277
2.3. Direcciones de propósito especial ..... 279
2.4. CIDR y supernetting ..... 281
2.5. El formato del paquete IP ..... 282
2.6. Encaminamiento y encaminadores ..... 286
**3. El ARP (*address resolution protocol*)** ..... 290
**4. El ICMP (*Internet control message protocol*)** ..... 292
4.1. Mensajes ICMP ..... 292
4.2. El programa ping ..... 294
4.3. El programa traceroute ..... 297
4.4. Mensaje de reencaminamiento ..... 298
**5. Protocolos de encaminamiento** ..... 300
5.1. RIP ..... 301
5.2. IGRP ..... 302

5.3. OSPF ..... 302
5.4. IS-IS ..... 303
**6. Mecanismos de acceso a Internet** ..... 304
6.1. Acceso telefónico: el PPP ..... 305
6.2. Acceso ADSL ..... 309
6.3. Acceso LAN: el protocolo Ethernet ..... 311
6.4. El DHCP (*Dynamic Host Configuration Protocol*) ..... 312
**7. NAT** ..... 314
7.1. NAT estática ..... 315
7.2. NAT dinámica ..... 317
7.3. PAT (*Port Address Translation*) ..... 319
7.4. LSNAT (*Load Sharing NAT*) ..... 323
**8. IPv6** ..... 325

**Glosario** ..... 327
**Bibliografía** ..... 335

Capítulo I

# Conceptos básicos de redes de computadores

*Jordi Íñigo Griera*
*Enric Peig Olivé*

## Introducción

Las redes de ordenadores actuales constituyen una amalgama de dispositivos, técnicas y sistemas de comunicación que han ido apareciendo desde finales del siglo XIX o, lo que es lo mismo, desde la invención del teléfono. Este último se desarrolló exclusivamente para transmitir voz, pero actualmente, en muchos casos se utiliza también para conectar ordenadores entre sí. Desde entonces, han aparecido las redes locales, las conexiones de datos a larga distancia con enlaces transoceánicos o satélites, Internet, la telefonía móvil, etc. Son una gran cantidad de tecnologías que configuran las redes de ordenadores que utilizamos como usuarios y que nos proponemos estudiar en este libro.

Dedicaremos este capítulo a introducir las ideas y los conceptos básicos de las redes de ordenadores que trataremos exhaustivamente a lo largo del libro.

Para tener una buena perspectiva de estas tecnologías y entender por qué se han creado, es recomendable considerar la evolución histórica de las telecomunicaciones, que a partir de mediados del siglo XX va íntimamente ligada a la historia de la informática. Este breve repaso histórico debe permitirnos contextualizar dichas tecnologías.

Después introduciremos brevemente el concepto de red de paquetes, en contraposición a las redes de conmutación de circuitos, que nos servirá de punto de partida tanto para las redes de área local como para la red Internet, que veremos en los otros capítulos.

Por último, se presentará un concepto fundamental en sistemas distribuidos: las arquitecturas de protocolos en general y el modelo de referencia OSI en particular. Aunque hoy día este modelo no disfruta de una gran popularidad, sus vir-

tudes pedagógicas están más que demostradas. A partir de este modelo es fácil estudiar y entender otras arquitecturas, como la arquitectura Internet.

# 1. Breve historia de las comunicaciones

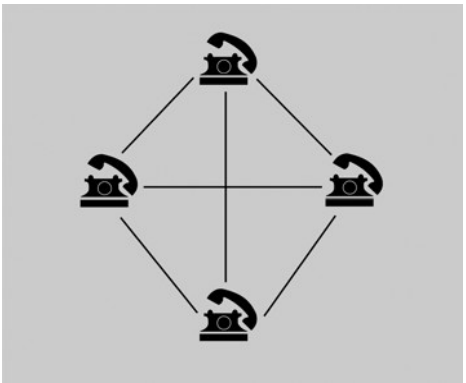
## 1.1. El teléfono

En 1878, Alexander Graham Bell mostró su "máquina eléctrica parlante" y cómo podía mantener una conversación a distancia entre dos de estos aparatos unidos por un hilo eléctrico.

> El teléfono no fue el primer sistema de telecomunicación. Mucho antes se habían utilizado sistemas ópticos que, con la luz del sol y juegos de espejos, permitían comunicarse desde distancias considerables: con un obturador que se abría y se cerraba a voluntad con diferentes periodos, se podían componer mensajes. Con posterioridad, a mediados del siglo XIX, se inventó el telégrafo. El mismo concepto, pero con un circuito eléctrico, un pulsador y un timbre.
>
> Todos estos sistemas requerían conocer un código para interpretar el mensaje. En cambio, para utilizar el teléfono no es necesario conocer ningún código: se escucha exactamente lo mismo que si se tuviera al interlocutor al lado. Ésta ha sido la razón principal de su éxito y de que los otros dos sistemas casi no se utilicen.

Al principio, había muy pocos teléfonos y, para utilizarlos la gente, por cuenta propia, cableaba un aparato con otro: si alguien tenía un aparato en casa y otro en el trabajo, los conectaba directamente. A medida que el número de teléfonos instalados crecía, el interés por mantener múltiples comunicaciones también lo hacía: era preciso pensar en la manera de interconectarlos. Nacía la idea de red de comunicaciones.

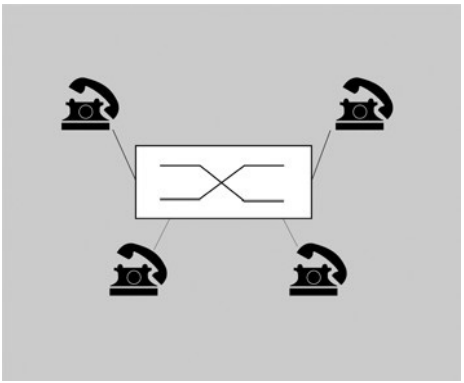
Una posible manera, bastante inmediata, de interconectar todos los aparatos sería lo que se puede observar en la figura siguiente:



Todos con todos.

Es evidente que este modelo de conexión, "todos con todos", es completamente inviable: para cada aparato nuevo que se incorpora a la red, se precisa un gran número de conexiones nuevas. Para hacernos una idea, una red "todos con todos" de cincuenta teléfonos necesita 1.225 líneas de conexión y, en cada teléfono, un dispositivo que permita cuarenta y nueve conexiones.

Para solucionar este problema, aparecieron compañías que ofrecían un servicio de conmutación: hacían llegar un cable hasta cada teléfono y conectaban los cables de los teléfonos que deseaban establecer una comunicación. De este modo, cada aparato disponía de una sola conexión y no era necesario establecer ninguna variación en la misma para incorporar nuevos aparatos a la red.



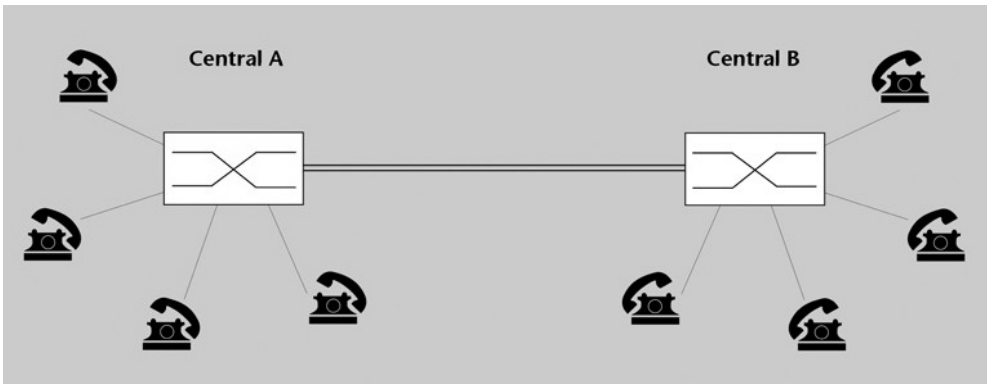
Central de conmutación.

De aquí provienen términos hoy tan comunes como abonado (el usuario que se abona a una central), bucle de abonado (el cable que une al abonado con la central) o central de conmutación.

Los primeros sistemas de conmutación no eran dispositivos muy sofisticados: eran personas, por lo general mujeres. Cuando alguien quería realizar una llamada, descolgaba y pedía a la operadora que le conectara con quien deseaba hablar. Una vez finalizada la comunicación, la operadora desconectaba los cables y, así, las líneas quedaban preparadas para recibir otras llamadas.

Las operadoras humanas fueron sustituidas progresivamente por ingenios electromecánicos: las centralitas. En los teléfonos se incorporó un disco con números para "marcar" el número del destinatario de la llamada. La centralita descodificaba este número para saber entre qué dos cables era preciso establecer la comunicación.

Este servicio de conmutación empezó en el ámbito local: un pueblo, un barrio, una ciudad. El paso siguiente consistió en ofrecer conexiones a larga distancia, conectando centrales locales entre sí directamente, o por medio de centrales de tráfico.




Comunicación entre dos centrales de conmutación

Entre las dos centrales locales se establece un enlace con diferentes cables independientes, de manera que los abonados de una de estas últimas pueden, además de conectarse entre sí, conectar con los abonados de la otra: se elige un cable de los que forman el enlace, se conecta con el abonado local y se pide a la otra central que conecte el enlace con el abonado destino si no está ocupado con otra llamada.

La conexión entre las dos centrales comporta un primer escollo importante: es preciso decidir con cuántas líneas diferentes se llevará a cabo.

Supongamos que la central A de la figura anterior proporciona servicio a cien abonados y la B, a doscientos cincuenta. Parece que, si se pretende dar el mejor servicio posible, se necesitan cien líneas para que todos los abonados de la central A puedan hablar de manera simultánea con otros de la central B.

No obstante, la probabilidad de que todos los abonados de una central realicen una llamada en el mismo momento es muy baja, puesto que las llamadas telefónicas son, en general, cortas y esporádicas. Por tanto, es completamente innecesario que la conexión entre las dos centrales incluya todas las llamadas posibles: esta situación no se dará nunca y tiene un coste exagerado.

Unos modelos matemáticos bastante complejos permiten calcular el número concreto de enlaces que se precisan a partir de la estadística de las llamadas que sirven las centrales (la frecuencia de aparición y su duración).

Supongamos que en el ejemplo anterior estos modelos nos dan veinticinco enlaces. Si en un momento dado hay veinticinco llamadas en curso entre A y B y llega otra llamada, no tendrá ningún camino disponible y, por consiguiente, no se podrá establecer. Esta situación se denomina *bloqueo*: el abonado a quien se quiere llamar no está ocupado; sin embargo, no se puede encontrar un camino libre por la red para establecer la comunicación.

De esta situación se desprenden dos ideas fundamentales en relación con la red telefónica:

- La conmutación de circuitos requiere pasar por tres fases para cada comunicación:
  - Establecimiento de llamada. Cuando se solicita iniciar una conversación, es preciso averiguar si el destinatario está disponible y, en caso afirmativo, debe buscarse un camino libre en la red, que incluya conmutadores dentro de las centrales y enlaces entre las mismas.
  - Comunicación. Una vez establecido el circuito, los interlocutores se intercambian información.
  - Liberación de recursos. Acabada la comunicación, se liberan los recursos utilizados (enlaces entre centrales y conmutadores dentro de las centrales).
- El hecho de que los recursos estén ocupados en exclusiva mientras dura la comunicación hace que las compañías que ofrecen el servicio cobren según la duración de la llamada: cuanto más tiempo están los recursos ocupados,

más se paga. De este modo, el usuario se apresura en acabar la comunicación y dejar los enlaces libres, de manera que disminuye la probabilidad de bloqueo.

La red telefónica constituye una red de conmutación de circuitos. Para llevar a cabo una comunicación es preciso establecer un circuito entre los dos extremos por medio de la red. Mientras dura la comunicación, se ocupan unos recursos en exclusiva, aunque no haya intercambio de información. Las compañías cobran el uso de los recursos por tiempo de ocupación.

Pronto el sistema telefónico pasó a ser una cuestión nacional. Los estados desarrollaban sus redes según sus criterios y gustos. Se creó un organismo, el CCITT (Comité Consultivo Internacional de Telegrafía y Telefonía, Comité Consultatif International Télégraphique et Téléphonique), para armonizar los sistemas nacionales y permitir, con centrales de tráfico internacionales, las comunicaciones entre países.

**El CCITT o ITU-T**

El CCITT es un organismo internacional patrocinado por las operadoras de telefonía, dedicado a tareas de normalización en el ámbito de las telecomunicaciones. El 1 de marzo de 1993 pasó a llamarse ITU-T (*International Telecommunication Union Standardisation Sector*).

Hemos comentado que entre las centrales existe una serie de líneas que permiten la conexión entre abonados de diferentes centrales. Al principio era realmente así: si se decidía que entre dos centrales era preciso disponer de cincuenta enlaces, se instalaban cincuenta cables entre una y otra. Sin embargo, con el progresivo aumento de enlaces necesarios, este sistema pronto fue totalmente inviable y fue preciso recurrir a una técnica ya conocida en radiodifusión: la multiplexación[1].

La técnica de multiplexación que se aplicó a la telefonía fue la multiplexación en frecuencia: se modulan los diferentes canales de entrada a distintas frecuencias portadoras, de manera que puedan viajar por el mismo medio sin interferirse. Se aplican filtros a la recepción que permiten separar los distintos canales multiplexados.

1. *Multiplexar* significa hacer pasar diferentes comunicaciones independientes por el mismo medio de transmisión.

Hacemos lo mismo al escuchar la radio o al ver la televisión. Hasta nuestra antena llegan todos los canales emitidos; con el dial y el selector de canales, respectivamente, seleccionamos el canal (la gama de frecuencias) correspondiente a la emisora que queremos recibir. Es decir, el dial o el selector de canales de la televisión constituyen los filtros que separan, en recepción, los diferentes canales multiplexados.

El número de canales diferentes que pueden viajar por un medio multiplexado depende del ancho de banda de la señal y de la capacidad del medio.

Por lo que respecta a la capacidad del medio, no dispone de la misma un par de hilos que un cable coaxial o que una fibra óptica.

En cuanto al ancho de banda, en el caso de la voz, debería ser de 19.980 Hz (que es un ancho de banda considerable) puesto que el oído humano es capaz de distinguir frecuencias entre los 20 Hz y los 20.000 Hz. No obstante, a raíz de estudios que se llevaron a cabo sobre las características de la voz humana, se llegó a la conclusión de que con mucho menos bastaba, puesto que la inteligibilidad de la voz se concentra en una banda bastante estrecha, entre los 300 Hz y los 3.400 Hz.

A partir de esta conclusión, se tomó una decisión que, a la larga, ha condicionado mucho el uso de la red telefónica: hacer el canal de voz de 4 kHz (entre 300 Hz y 3.400 Hz, más unas bandas laterales de guardia).

Ello explica por qué se escucha tan mal la música por el teléfono: no hay ni graves ni agudos, sólo están las frecuencias del medio.

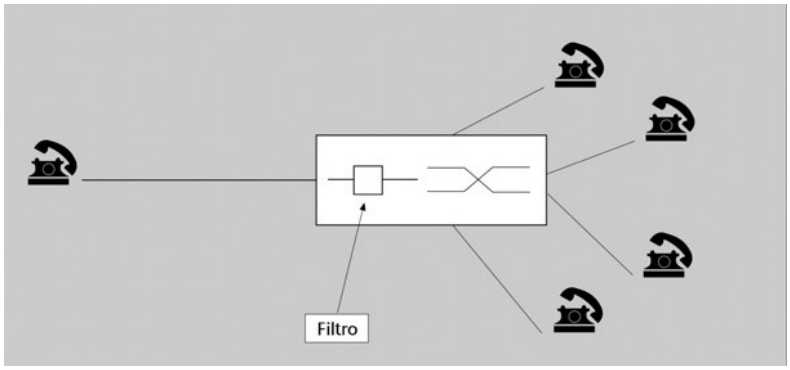
A partir de aquí, se estandarizaron los diferentes niveles de multiplexación. El nivel básico es la agrupación de distintos canales de 4 kHz, el siguiente es una agrupación de multiplexados básicos, etc.

**Jerarquía estándar**

La jerarquía que propuso la compañía americana AT&T es la siguiente:

| Nombre | Rango | Ancho de banda | Canales de voz |
|---|---|---|---|
| Group | 60-108 kHz | 48 kHz | 12 |
| Supergroup | 312-552 kHz | 240 kHz | 60 |
| Mastergroup | 564-3.084 kHz | 2,52 MHz | 600 |
| Jumbogroup | 0,5-17,5 MHz | 17 MHz | 3.600 |

En la entrada de la central local se encuentra un filtro que elimina cualquier frecuencia por encima de los 4 kHz. La señal de salida de este último es la que se multiplexa, conmuta y lleva hasta el destinatario.




Con todo ello, ya podemos dibujar un panorama completo de la red telefónica, tal como era hasta los años setenta:

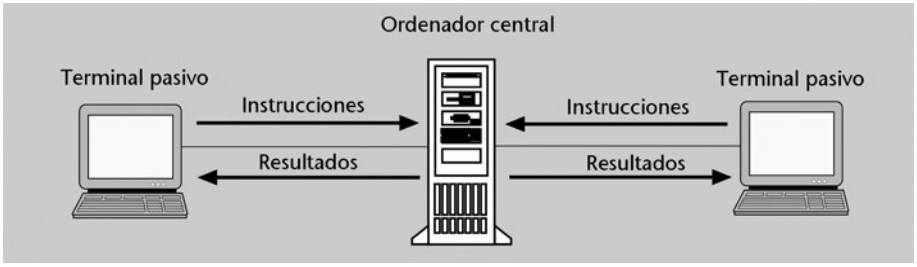
La red telefónica es analógica, ubicua, trabaja con la técnica de conmutación de circuitos, con tarifación por tiempo de ocupación, con enlaces multiplexados en frecuencia y con canales limitados a 4 kHz.

## 1.2. Aparecen los primeros ordenadores

La década de los sesenta vio la aparición de los primeros ordenadores comerciales. Eran grandes, caros y poco potentes[2]. Sólo organismos oficiales, grandes empresas o universidades podían comprarlo, y lo que es más normal es que sólo compraran uno (o algunos, pero no uno para cada usuario, como hoy día estamos acostumbrados a ver). Por ello, estos ordenadores llevaban sistemas operativos multitarea y multiusuario, para que diferentes usuarios, realizando distintos trabajos, pudieran utilizarlos simultáneamente. El acceso a dichos ordenadores

2. Evidentemente, comparados con los que hoy estamos acostumbrados a ver. Para la época, eran unas máquinas fantásticas.

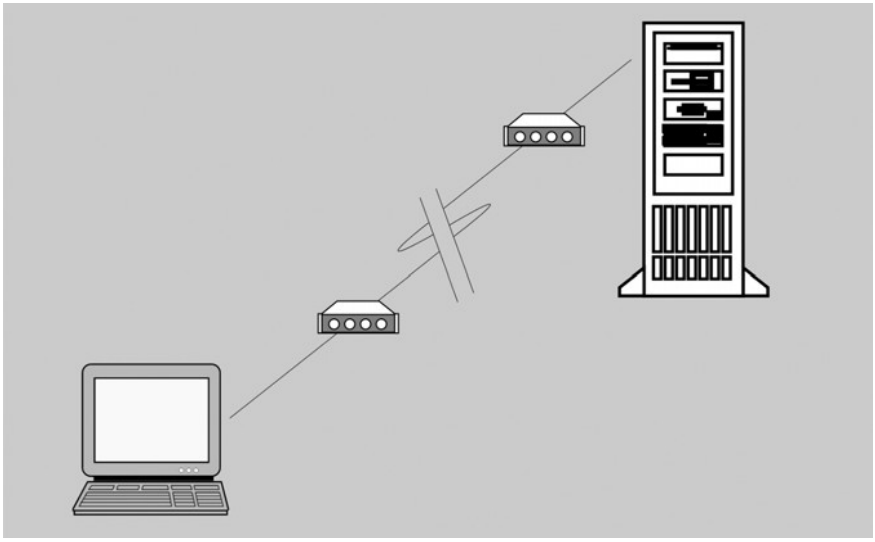
se llevaba a cabo por medio de terminales sin ninguna capacidad de proceso, pasivos[3]:




### *1.2.1. Los módems*

No tardó mucho en aparecer la necesidad de poder alejar los terminales de la unidad central para conectarse, por ejemplo, desde casa o desde una delegación al ordenador central.

Para poder realizar este acceso remoto, la primera solución que aportaron los ingenieros informáticos de la época fue utilizar la red telefónica que, por su ubicuidad, les ahorraba generar infraestructuras nuevas. Sólo se precisaba un apara-

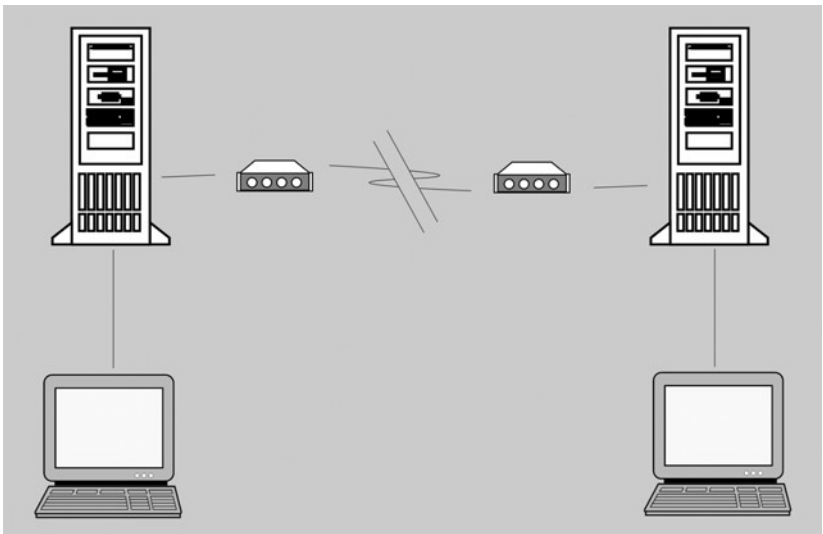


3. En inglés *dumb terminal* ('terminal mudo'). Coloquialmente, estos terminales se llaman *terminales tontos*.

to que adaptara los bits a la red (recordad que la red telefónica sólo deja pasar sonidos entre unos márgenes de frecuencia). Estos aparatos son los módems[4].

Los primeros módems eran de 300 bps y generaban dos tonos diferentes: uno para el 1 lógico y otro para el 0. Los de mayor velocidad van a 56.000 bps, que es el máximo que permite la red telefónica convencional actual.

Los módems no sólo servían para poder alejar los terminales pasivos de los ordenadores centrales, también permitían interconectar ordenadores entre sí, de manera que desde los terminales de uno se podía acceder al otro, y viceversa.



La tecnología de conmutación de circuitos se desarrolló en un origen para las comunicaciones telefónicas y una de sus características fundamentales era la ocupación en exclusiva de los recursos mientras duraba la conexión, lo que (como ya hemos visto) justificaba la tarifación por tiempo. Sin embargo, las comunicaciones informáticas no son cortas, intensas y esporádicas como las de voz. Al conectar un terminal a un ordenador central por medio de dos módems, no están pasando datos todo el tiempo que dura la conexión: puede haber largos periodos de tiempo en los que no pase ningún bit y momentos en los que haya un intercambio de datos intenso, aunque a una velocidad de transmisión mucho más baja

4. Módem es un acrónimo de *modulator-demodulator*, que se refiere a su función: modular (generar señales audibles según los valores de los bits) y demodular (generar bits a partir de las señales que recibe de la red telefónica).

que la que se puede mantener entre el terminal y el ordenador conectados directamente. Las facturas telefónicas empezaron a ser astronómicas y desproporcionadas respecto del uso real de la red.

### *1.2.2. Las redes de datos*

Pronto las grandes empresas presionaron a las compañías telefónicas del momento para que desarrollaran redes pensadas para transportar datos, cuyo sistema de tarifación se ajustara al tráfico de datos real y permitiera más velocidad que los escasos 300 o 1.200 bps de la época, que se lograban utilizando la red telefónica. La respuesta fueron las redes de conmutación de paquetes.

El envío de datos no necesariamente debe llevarse a cabo en tiempo real (las transmisiones de voz, sí). Por tanto, no es preciso establecer el camino entre los dos puntos antes de empezar la transmisión y mantenerlo mientras dura el intercambio de datos. En lugar de ello, se empaquetan los bits que deben transmitirse y se entregan a la central más próxima para que los envíe cuando pueda a la siguiente, y así sucesivamente hasta que lleguen al destino. Si cuando un paquete llega a una central todos los enlaces con la siguiente están ocupados, no pasa nada, lo hace esperar poniéndolo en una cola para enviarlo cuando haya un enlace disponible.

La transmisión por paquetes tiene la ventaja de que sólo ocupa los recursos cuando en realidad se utilizan, no siempre. Sin embargo, como contrapartida, es preciso soportar el retraso que pueda producirse entre que los paquetes salen del origen y llegan a su destino. Retraso que a su vez es variable, puesto que las esperas en las colas son aleatorias, al depender del estado de la red. Sin embargo, como hemos dicho, ello, en comunicación de datos, es hasta cierto punto tolerable. Por lo que respecta a la cuestión económica, no tiene sentido que se cobre por tiempo de conexión: en las redes de datos se paga por bits transmitidos.[5]

Existe otro peligro: los paquetes pueden perderse. Conviene tener presente que las colas son limitadas y, si llega un paquete cuando una ya está llena, no se podrá guardar y se perderá. Es preciso prever mecanismos que eviten dichas pérdidas y regulen el flujo de información entre los nodos de conmutación.

5. En España, esta red de datos se llamaba Iberpac. En la actualidad, para comunicaciones de datos se utiliza el Frame Relay, la evolución natural del X.25, y cada vez más, ATM.

Las compañías telefónicas desarrollaron redes de este tipo, y el CCITT emitió un estándar, el X.25, que es el que se ha adoptado hasta hace poco.

### *1.2.3. Las redes de área local*

Cuando empezó a ser habitual disponer de más de un ordenador en la misma instalación, apareció la necesidad de interconectarlos para poder compartir los diferentes recursos: dispositivos caros, tales como impresoras de calidad, un disco duro que almacenara los datos de la empresa, un equipo de cinta para realizar copias de seguridad, etc.

El diseño de las redes de área local[6] siguió caminos completamente diferentes de los que se siguieron para las redes de gran alcance. En las redes de área local se necesita, habitualmente, establecer comunicaciones "muchos a uno" y "uno a muchos", lo que es difícil de conseguir con las redes de conmutación, pensadas para interconectar dos estaciones. Para este tipo de redes es más adecuada la difusión con medio compartido, donde los paquetes que salen de una estación llegan a todo el resto simultáneamente. En la recepción, las estaciones los aceptan o ignoran dependiendo de si son destinatarias de los mismos o no.

Se habla de *difusión* porque los paquetes se difunden por todos lados, y de *medio compartido* porque esta difusión se lleva a cabo sobre un medio común que las estaciones comparten.

## 1.3. Arquitecturas de protocolos

De la década de los sesenta datan también los primeros estándares de arquitecturas de protocolos. Conviene tener presente que el intercambio de información entre ordenadores tiene toda una serie de implicaciones, entre las que se encuentran las siguientes:

- Aspectos eléctricos: los cables, los conectores, las señales, etc.

6. Con frecuencia se utiliza la sigla inglesa LAN (*local area network*) para identificar las redes de área local, y la sigla WAN (*wide area network*) para identificar las redes de gran alcance.

- La manera de agrupar los bits para formar paquetes y la de controlar que no se produzcan errores de transmisión.
- La identificación de los ordenadores dentro de la red y la manera de conseguir que la información que genera un ordenador llegue a quien se pretende.

Asumir todos estos aspectos de una manera global no es viable: demasiadas cosas y demasiado diferentes entre sí. Por ello, ya desde el principio, se desarrollaron modelos estructurados en niveles: en cada nivel se lleva a cabo una tarea y la cooperación de todos los niveles proporciona la conectividad deseada por los usuarios.

Conviene considerar que, en la época que nos ocupa, la informática estaba en manos de muy pocos fabricantes e imperaba la filosofía del servicio integral: cada fabricante lo proporcionaba todo (ordenadores, cables, periféricos, sistema operativo y software). Por tanto, cuando una empresa se quería informatizar, elegía una marca y quedaba vinculada a la misma para toda la vida.

Hablamos de empresas como IBM (International Business Machines) o DEC (Digital Equipment Corporation). Cuando estas últimas se propusieron ofrecer conectividad entre sus equipos, o bien local o bien remota, también lo hicieron aplicando la filosofía de la separación por niveles: IBM desarrolló la arquitectura SNA (*system network arquitecture*) y DEC, la DNA (*DEC network arquitecture*). Eran dos modelos completos, estructurados en niveles, pero incompatibles entre sí, según la filosofía de la informática propietaria.

En la década de los setenta el panorama cambió radicalmente, sobre todo a causa de tres acontecimientos:

- La propuesta del protocolo Ethernet para redes locales.
- La aparición del sistema operativo Unix, que no estaba vinculado a ninguna marca comercial, compatible con todas las plataformas de hardware existentes.
- La invención de los protocolos TCP/IP, embrión de la actual Internet.

Se había allanado el camino para la aparición de los sistemas abiertos: no era preciso vincularse a ninguna marca para tenerlo todo. El hardware podía ser de un proveedor, el sistema operativo de otro, las aplicaciones de otro y los protocolos, públicos.

TCP/IP[7] nació a partir de un encargo de la DARPA[8] a la comunidad científica americana para obtener una red mundial que fuera reconfigurable con facilidad y de forma automática en caso de destrucción de algún nodo o de algún enlace.

La pila TCP/IP era una jerarquía de protocolos que ofrecía conectividad y, a pesar de tener poco que ver con las que ya existían, constituía una opción más en el mercado. Ante una oferta tan grande y dispar de protocolos, la ISO (Organización Internacional de Estandarización, International Organization for Standardization) y el CCITT propusieron un nuevo modelo que intentaba reunir de algún modo todo lo que ya se había propuesto y que pretendía ser completo, racional y muy bien estructurado (la TCP/IP tiene fama de ser una pila de protocolos anárquica), con la intención, por tanto, de que se convirtiera en un modelo de referencia. Es la conocida como pila de protocolos OSI (*open systems interconnection*).

Internet, que nació y creció en las universidades, se empezó a popularizar en la década de los noventa, a medida que quienes conocían la Red la iban "enseñando", y su eclosión se produjo cuando saltó al mundo de la empresa, en todas sus vertientes: como escaparate de productos o como canalizador de contactos comerciales.

Sin embargo, el origen universitario de la Red ha marcado su evolución en muchos sentidos. Por ejemplo, el modelo cliente/servidor de aplicaciones distribuidas. Es un modelo sencillo y, al mismo tiempo, potente, y casi todas las aplicaciones que se utilizan en Internet lo siguen. El Telnet, o apertura de sesión remota, la transferencia de ficheros (FTP), el correo electrónico y, sobre todo, el WWW (*World Wide Web*) constituyen ejemplos claros de aplicaciones que siguen este modelo. Las dos primeras han caído un poco en desuso, pero tanto el correo como el WWW son las actuales estrellas en Internet. Tímidamente, aparecen nuevas propuestas de aplicaciones; sin embargo, el WWW, que nació como un servicio de páginas estáticas enlazadas con hiperenlaces, se está convirtiendo en la interfaz de usuario de toda la Red, puesto que en la actualidad se utiliza para servir páginas dinámicas (se crean en el momento en que se sirven), e, incluso, código que se ejecuta en el ordenador cliente (*applets*).

7. TCP/IP, siglas de *transmission control protocol/Internet protocol* (protocolo de control de transmisión/protocolo de Internet).
8. DARPA, siglas de Defense Advanced Research Project Agency (Agencia de Proyectos de Investigación Avanzada para la Defensa.

## 1.4. La digitalización de la red telefónica

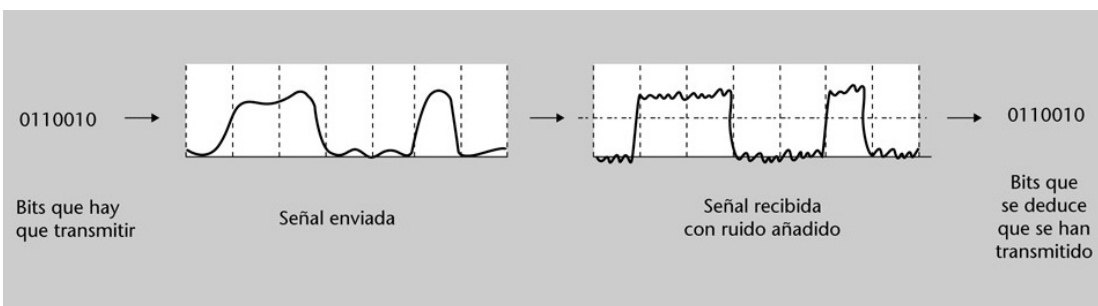
En este momento tenemos dos redes completamente independientes entre sí, pero de alguna manera superpuestas:

- Una red analógica, con conmutación de circuitos, pensada para voz.
- Una red digital, con conmutación de paquetes, pensada para datos.

La red telefónica, tal como la hemos descrito hasta ahora, es completamente analógica: la señal electromagnética que viaja desde un teléfono hasta otro es analógica (varía continuamente y en cualquier momento puede adoptar cualquier valor) y los circuitos electrónicos que componen la red también lo son.

Los enlaces entre centrales de la red telefónica se llevaban a cabo con señales analógicas con muchos canales multiplexados en frecuencia y, en ocasiones, debían recorrer grandes distancias. La atenuación de la señal inherente a la distancia que era preciso recorrer debía corregirse por medio de repetidores que la amplificaban, lo que aumentaba el ruido presente en la línea. A menudo, la señal recibida era de una calidad muy baja porque la transmisión analógica no permitía eliminar el ruido y las interferencias en la recepción. No existía ninguna manera de saber con exactitud qué se había enviado desde el origen y qué era ruido añadido.

En 1972, se hicieron públicos los primeros resultados del tratamiento digital de la señal aplicado a audio, básicamente orientado a su almacenamiento. El CD estaba viendo la luz. Convertir un sonido (una magnitud física que puede adoptar cualquier valor en cualquier momento) en una serie de 0 y 1 (dos únicos valores, conocidos) permitía corregir con facilidad cualquier ruido añadido:



En el caso de la señal analógica, viendo la señal recibida, no se puede deducir cuál ha sido la señal emitida. En cambio, en el caso de la señal digital, como se conocen los valores enviados, se establece un umbral en el punto medio entre los dos valores y se decide que todo lo que esté por encima corresponde a un 1 y todo lo que esté por debajo, a un 0.

Si el ruido que se ha añadido es superior a la diferencia entre el valor original y el umbral, se produce un error de recepción: se decide que se había enviado el valor equivocado. Las técnicas para luchar contra este tipo de errores se verán más adelante.

El descubrimiento del procesado digital de la señal, así como sus aplicaciones en los campos del sonido y la imagen, ha constituido un hito capital en el mundo de las comunicaciones. Básicamente, ha permitido reducir drásticamente el efecto del ruido, lo que ha posibilitado, por un lado, incrementar la calidad de recepción de las señales y, por el otro, aumentar la velocidad de transmisión con los mismos medios.

Las compañías telefónicas empezaron a sustituir los enlaces internos (entre centrales) por señales digitales, pero manteniendo el bucle de abonado (línea y terminal) analógico. La digitalización de la señal de sonido se lleva a cabo dentro de la central local, después del filtro de 4 kHz, y se vuelve a pasar a analógica en la central correspondiente al otro extremo de la comunicación.

Ello ha hecho cambiar sustancialmente los procesos de conmmutación: ahora debe trabajarse con bits y, por tanto, las centrales electromecánicas deben sustituirse por ordenadores.

Esta digitalización de la parte interna de la red de voz hizo que, de algún modo, las dos redes, la telefónica y la de datos, confluyeran: los enlaces digitales entre centrales se utilizaban indistintamente para paquetes de datos y para transmisiones de voz.

### *1.4.1. La red digital de servicios integrados*

Una vez digitalizada la red telefónica, el paso siguiente debía ser llevar la transmisión de bits hasta las casas. Ello permitía, por un lado, ofrecer a los usuarios en su casa la transmisión de datos además de la tradicional de voz y, por otro, ofrecer a los abonados un abanico de nuevos servicios asociados a una comunicación enteramente digital de punta a punta. Este servicio de transmisión digital por medio de la red telefónica se conoce como red digital de servicios inte-

grados (RDSI[9]). Ofrece dos canales independientes de 64 kbps, que permiten hablar y conectarse a Internet simultáneamente, o, con el hardware adecuado, aprovechar los dos canales juntos para navegar a 128 kbps.

## 1.5. La banda ancha

El uso de la red telefónica para transmitir datos tiene una limitación importante por lo que respecta al máximo de bits por segundo permitidos y las redes específicas de datos son muy caras para el uso doméstico. Desde la década de los noventa, se han estudiado maneras de llevar hasta las casas o las empresas un buen caudal de bits por segundo (banda ancha) a un precio razonable, de manera que las nuevas aplicaciones multimedia se puedan explotar al máximo.

Para conseguir esta banda ancha, se han seguido dos caminos completamente diferentes. Por lo que respecta al primero, se han promovido cableados nuevos con fibra óptica que permitan este gran caudal, con frecuencia llevados a cabo por empresas con afán competidor contra los monopolios dominantes. Estas redes se aprovechan para proporcionar un servicio integral: televisión, teléfono y datos.

En cuanto al segundo, las compañías telefónicas de toda la vida han querido sacar partido del cableado que ya tienen hecho y, por ello, se han desarrollado las tecnologías ADSL, que permiten la convivencia en el bucle de abonado de la señal telefónica y una señal de datos que puede llegar a los 8 Mbps (o 20 Mbps con tecnología ADSL+).

**Banda estrecha o banda ancha**

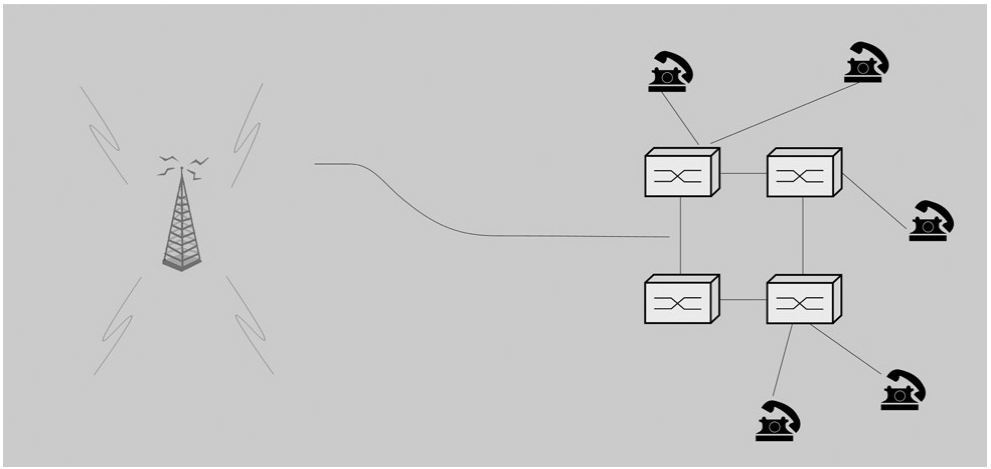
La frontera entre banda estrecha y banda ancha no está muy clara. Los 128 kbps de la RDSI se consideran banda estrecha y había quien calificaba de banda ancha a los 256 kbps de la ADSL en sus inicios.

Realmente, se considera banda ancha a partir de 1 Mbps.

9. RDSI. La red digital de servicios integrados (RDSI) corresponde a las siglas en inglés ISDN (*integrated services digital network*).

## 1.6. La telefonía móvil

La telefonía móvil, todo un fenómeno sociológico actual, ha vivido una evolución fulgurante: en menos de veinte años, ha pasado de la nada a constituir una tecnología de uso diario para más de un 80% de la población. En el 2006, el número de hogares con móvil en España superó por primera vez a los que tienen fijo.



Desde el punto de vista de sistema de comunicación, debemos ver los móviles como una extensión de la red telefónica convencional.

El sistema GSM, que constituye el actual estándar europeo, permite el acceso a la red de voz, cambiando el bucle de abonado: en lugar de ser un cable, es un enlace radioeléctrico entre una antena y el móvil. Se trata, por tanto, de una red de conmutación de circuitos y se continúa fijando la tarifa por tiempo de conexión.

El estándar GPRS permite el transporte de bits, pagando por tráfico en lugar de por tiempo. Por tanto, es aproximadamente el clónico de las redes de datos con hilos.

El estándar UMTS, también conocido como tercera generación (3G), permite transferencias del orden de megabits por segundo, necesarias para disponer de aplicaciones multimedia en el móvil. Sin embargo, requiere nuevas antenas y terminales.

## 2. Las redes de gran alcance

Como veíamos en el apartado anterior, las redes de conmutación de paquetes aparecieron para proporcionar transmisión de información a larga distancia sin tener que utilizar la red telefónica. Ahora veremos en este apartado los principios de funcionamiento de las redes de paquetes y cuáles son las principales diferencias con las redes de conmutación de circuitos.

### 2.1. Conmutación de circuitos

La conmutación de circuitos consiste en establecer un camino físico continuo a través de la red entre las dos estaciones que se quiere comunicar, como si fuera una línea punto a punto.

Este camino estará compuesto por diferentes enlaces entre los nodos, y hay que crearlo antes de iniciar la transmisión de los datos y deshacerlo al acabarla. De este modo, en todas las conexiones podemos distinguir tres fases:

- Establecimiento del circuito. Lo primero que debe hacerse es encontrar un camino a través de la red entre las dos estaciones. La estación de origen lo pide al nodo al que está conectada. Éste traslada la petición a uno de los nodos a los que está conectado; éste, a otro y así sucesivamente hasta llegar a la estación de destino. A medida que la petición va pasando por los nodos, se va componiendo el circuito. Si la estación de destino acepta la conexión, envía una señal a la estación de origen para indicárselo.
- Transferencia de datos. Una vez establecido el circuito, las dos estaciones se comportan como si hubiera un enlace punto a punto entre las mismas. La información se transmite sobre este circuito.
- Desconexión. Acabada la conexión, hay que liberar los recursos que se han utilizado (los enlaces entre nodos y dentro de los nodos) para que puedan ser utilizados en conexiones posteriores.

La figura siguiente representa un diagrama de tiempo. En horizontal tenemos el paso del tiempo, mientras que en vertical tenemos la distancia. Así, las líneas

inclinadas simbolizan el viaje de los datos desde un punto a otro. La inclinación da idea de la velocidad de transmisión: cuanto más vertical es la línea, menos tiempo para el mismo espacio y, por lo tanto, más velocidad de transmisión. Simbolizamos la presencia de nodos intermedios entre origen y destino con las líneas horizontales.

La figura siguiente ilustra estas tres fases:

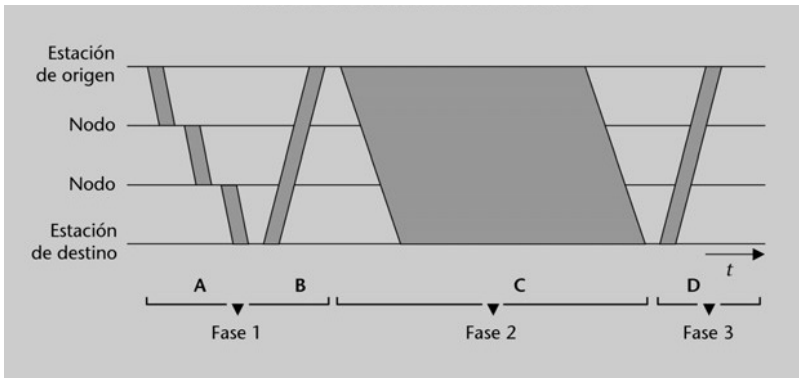


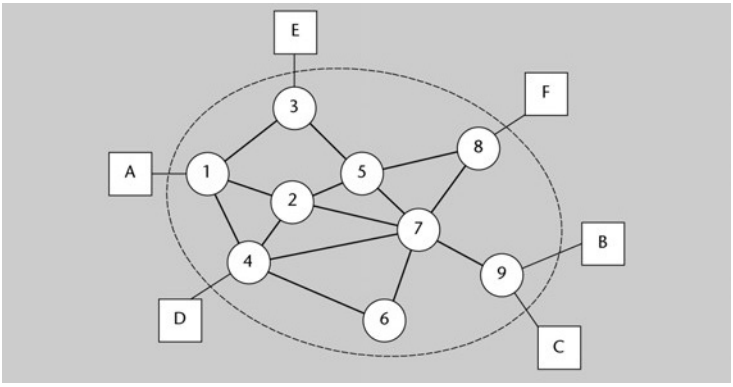

Diagrama de tiempo de una conexión mediante una red de conmutación de circuitos.

Para acabar, presentamos las tres características más importantes de este tipo de redes:

- Una vez establecido el circuito, la red no afecta en nada a la conexión, y la transferencia de datos tiene lugar como si se tratara de una línea punto a punto. Entre las estaciones se establece un circuito de datos, igual que si estuvieran conectadas directamente.
- Una vez establecido el enlace entre las dos estaciones, los recursos que lo forman están ocupados aunque no haya transferencia de datos, hasta que se solicite explícitamente el final de la conexión. Esto puede provocar un uso ineficiente de la red si el porcentaje del tiempo total de conexión durante el cual realmente se transfiere información es bajo.
- Las redes de conmutación de circuitos pueden presentar bloqueo. Esto sucede cuando, en situaciones de carga alta, hay tantos recursos ocupados que dos estaciones no se pueden conectar porque no hay camino posible dentro de la red.

### *2.1.1. Conmutación de mensajes*

Supongamos que dos estaciones, A y B, se quieren intercambiar mensajes y que se dispone de la red que se muestra en la figura siguiente:




Ejemplo de red de conmutación.

Si la red fuera de conmutación de circuitos, habría que buscar un camino entre A y B (por ejemplo, a través de los nodos 1, 2, 7 y 9); y, una vez establecida, las estaciones empezarían a transmitir los datos.

Supongamos ahora que A, en lugar de esperar a que se haya establecido el camino completo entre los dos extremos, envía un mensaje entero al nodo 1. Éste lo almacena y a continuación lo transmite al nodo 2. Una vez la estación A ha enviado el mensaje entero al nodo 1, no hay que mantener ocupado este enlace, se puede liberar y volverlo a ocupar cuando A tenga otro mensaje a punto para B. Lo mismo se puede decir para el enlace entre los nodos 1 y 2: una vez el nodo 2 ha recibido el mensaje, el enlace entre los dos ya se puede liberar. El proceso sigue (de 2 a 7, de 7 a 9) hasta que el nodo 9 envía el mensaje a su destinatario, la estación B.

El procedimiento en el sentido contrario es idéntico: B entrega el mensaje que quiere enviar a A al nodo 9; el 9, al 7; el 7, al 2; el 2, al 1, y finalmente el 1, a la estación A.

Si ahora A quisiera enviar otro mensaje a B, se repetiría todo el proceso.

De esta manera, se consigue lo que se perseguía: los enlaces sólo están ocupados el tiempo que realmente se utilizan y, por lo tanto, la eficiencia es superior.

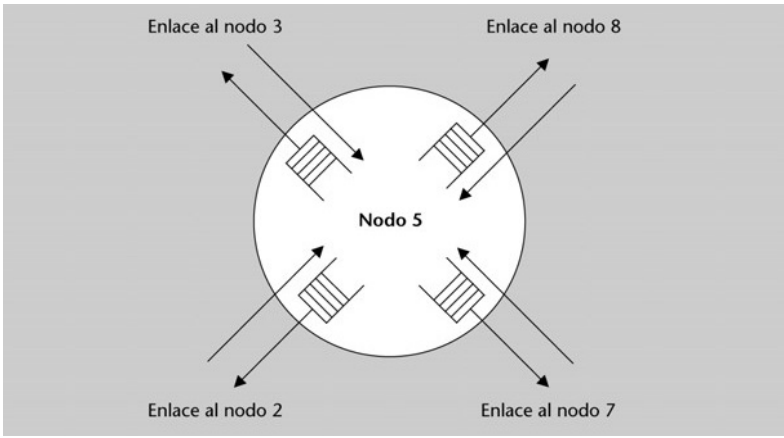
Además, no es necesario que estén disponibles todos los nodos: mientras la estación A transmite el mensaje al nodo 1, el enlace entre el 7 y el 9 puede estar totalmente ocupado por otras conexiones. Cuando el mensaje llegue al nodo 7, éste lo almacenará y esperará a que se libere uno de los canales del enlace hacia el nodo 9 para poder enviarlo.

La técnica que acabamos de describir se denomina conmutación de mensajes, como analogía de la conmutación de circuitos que hemos visto antes, porque los nodos envían mensajes de un enlace a otro. En este caso, sin embargo, no se utilizan elementos electromecánicos.

La conmutación de mensajes se basa en el almacenamiento y la retransmisión; por este motivo, en inglés se denomina *Store & Forward*. Esta técnica es la misma que se aplica en el sistema postal. Se lleva la carta a un buzón, en el que queda depositada temporalmente; un funcionario la lleva a un centro de clasificación, de éste se lleva a otro… hasta que el cartero la deja en casa del destinatario.

Los nodos de una red de conmutación de mensajes son ordenadores con capacidad de proceso y con una memoria estructurada en forma de colas, una en cada enlace de salida. Cuando llega un mensaje, el nodo decide por qué enlace tiene que salir y lo deja en la cola correspondiente, en la que espera el turno para ser enviado.

La figura siguiente muestra un esquema de cómo sería uno de los nodos de conmutación de mensajes de la red que acabamos de ver:




Esquema del nodo 5 de la red de conmutación de mensajes.

Es evidente que los enlaces que forman un camino trabajan de manera totalmente independiente unos de otros. Las estaciones no están conectadas directamente y, por lo tanto, pueden trabajar con diferentes velocidades de transmisión.

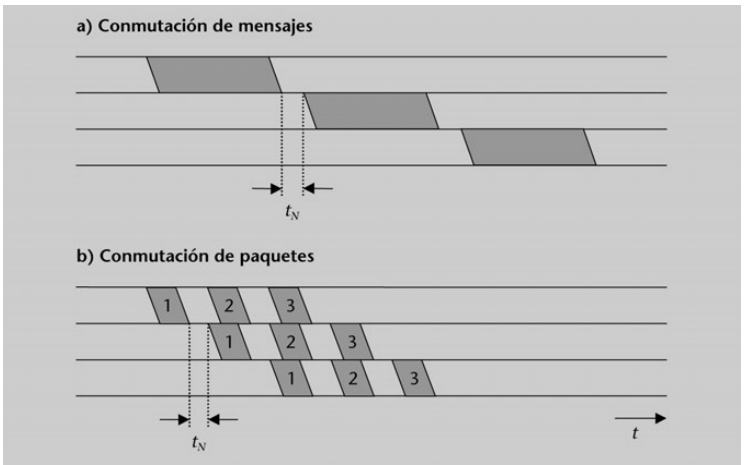
Esta técnica de la conmutación de mensajes, sin embargo, no se utiliza mucho tal y como se ha descrito porque presenta un inconveniente importante: cada nodo involucrado en la conexión tiene que esperar hasta haber recibido todo el mensaje para decidir hacia dónde lo transmite y almacenarlo en la cola de salida correspondiente, en la que el mensaje esperará a que los que tiene delante sean enviados.

Esto tiene una doble implicación:

a) La cantidad de memoria necesaria en cada nodo puede ser muy grande si los mensajes que las estaciones se intercambian lo son[10]. Se necesita memoria de trabajo del ordenador en el que residen los programas en ejecución y los datos que manipulan, y también memoria para las colas de salida.

b) El tiempo que los mensajes tardan en atravesar la red no es sólo el tiempo de propagación por los enlaces: hay que añadir los tiempos de procesamiento dentro de los nodos y los tiempos de espera dentro de las colas. Esto puede significar un retardo considerable, si consideramos que en conmutación de circuitos sólo cuenta el tiempo de propagación.

### *2.1.2. Conmutación de paquetes*




Comparación entre conmutación de mensajes y de paquetes.

10. No es exagerado pensar en mensajes de 1MB o más.

Con el fin de evitar los inconvenientes de la conmutación de mensajes, en la práctica se dividen los mensajes en trozos de unos pocos bytes denominados paquetes. De esta manera, se reduce drásticamente, por un lado, la memoria necesaria para almacenamiento y, por otro, el tiempo de procesamiento dentro de los nodos, con la consiguiente reducción del retardo acumulado dentro de la red.

- En el esquema a) de la figura podemos ver que el mensaje va pasando por los nodos hasta llegar a su destino. En este caso, cada nodo tiene que esperar a recibir todo el mensaje para enviarlo al siguiente.
- En el esquema b) el mensaje se ha dividido en tres paquetes; de esta manera, los nodos pueden empezar a transmitir cada paquete sin esperar a tener todo el mensaje.

El intervalo de tiempo $t_N$ de los dos esquemas incluye el tiempo de proceso y el tiempo de espera del mensaje o del paquete en la cola correspondiente de salida.

Hay que observar que, si se quiere hacer un diagrama de tiempo de una situación real y se quiere guardar unas proporciones entre las diferentes magnitudes de tiempo, el intervalo $t_N$ normalmente será bastante más largo de lo que muestra la figura.

La conmutación de paquetes es mejor que la conmutación de mensajes en todos los sentidos, pero continúa teniendo un problema que no presenta la conmutación de circuitos. Como hemos visto, los paquetes llegan a su destino con un cierto retardo, a causa del procesamiento que experimentan dentro de los nodos. Además, es muy importante señalar que este retardo es variable, porque las condiciones de la red influyen en todos los paquetes que circulan: los tiempos de espera en las colas de salida de los nodos no tienen por qué ser iguales durante toda la conexión. Por este motivo las redes de conmutación de paquetes no son adecuadas para aplicaciones de tiempo real o para situaciones en las que es necesario un flujo de datos constante.

Por otra parte, las redes de conmutación de paquetes no presentan bloqueo, porque no hay que asegurar la disponibilidad de todos los recursos necesarios antes de empezar la transmisión de datos.

El efecto que produce el incremento de tráfico en estas redes es un aumento del retardo global, porque las esperas en las colas son más elevadas. Si las condiciones empeoran, se puede producir congestión.

## 2.2. Datagramas y circuitos virtuales

Los nodos tienen que decidir hacia cuál de los otros nodos a los que están conectados envían los paquetes que les llegan. Esta decisión se puede tomar para cada paquete que forma el mensaje, o bien se puede tomar al principio; es decir, se puede decidir una ruta antes de empezar la transmisión, y entonces hacer que todos los paquetes que forman el mensaje o conexión la sigan. Estas dos posibilidades dan lugar a dos tipos diferentes de redes de conmutación de paquetes:

1) La primera posibilidad, decidir para cada paquete, da lugar a las redes de conmutación de paquetes en modo datagrama.

La técnica de conmutación de paquetes en modo datagrama consiste en encaminar cada uno de los paquetes independientemente de los otros.

Esto hace que los paquetes que forman un mensaje puedan seguir caminos diferentes y, si experimentan retardos distintos, puedan llegar al receptor desordenados. Entonces las estaciones deben encargarse de reordenarlos para obtener el mensaje tal y como se envió.

2) La segunda posibilidad, decidir la ruta al principio, da lugar a las redes de conmutación de paquetes en modo circuito virtual.

La técnica de conmutación de paquetes en modo circuito virtual consiste en elegir el camino por el cual pasarán todos los paquetes a través de la red antes de empezar una conexión.

De esta manera, los nodos no tienen que tomar decisiones de encaminamiento para cada paquete, y el receptor los recibirá siempre en el orden en el que fueron enviados. Es como un circuito físico, pero sin establecerlo realmente; por este motivo se habla de conexión lógica o circuito virtual.

En modo circuito virtual también se pueden distinguir las tres fases que hemos visto en conmutación de circuitos: establecimiento del circuito, transmisión de los datos y liberación del circuito, mientras que en modo datagrama no son necesarios ni el establecimiento ni la liberación, porque no hace falta un circuito (real o virtual) prefijado, lo cual los hace particularmente interesantes para transmitir mensajes cortos.

Los enlaces entre nodos están multiplexados con la técnica llamada STDM. Esto permite que por estos enlaces puedan pasar simultáneamente comunicaciones diferentes. En nuestro caso, estas comunicaciones se corresponden con los circuitos virtuales.

Dado que un enlace puede soportar diferentes circuitos virtuales, se tienen que distinguir de alguna manera. Esto se hace con el número del canal que ocupan dentro del enlace.

El circuito virtual entre la estación de origen y la estación de destino está definido por los enlaces que atraviesa y por el canal que ocupa en cada uno de los mismos.

En conmutación de datagramas, si un nodo (o un enlace) deja de funcionar, la red hace que los paquetes circulen por otro tramo y así no se pierde la conexión. En los circuitos virtuales, en cambio, la estación de origen debe restablecer el circuito para poder continuar la comunicación.

Se dice que las redes que trabajan con la técnica de datagramas ofrecen un servicio no orientado a conexión, mientras que las que trabajan con circuitos virtuales ofrecen un servicio orientado a conexión.

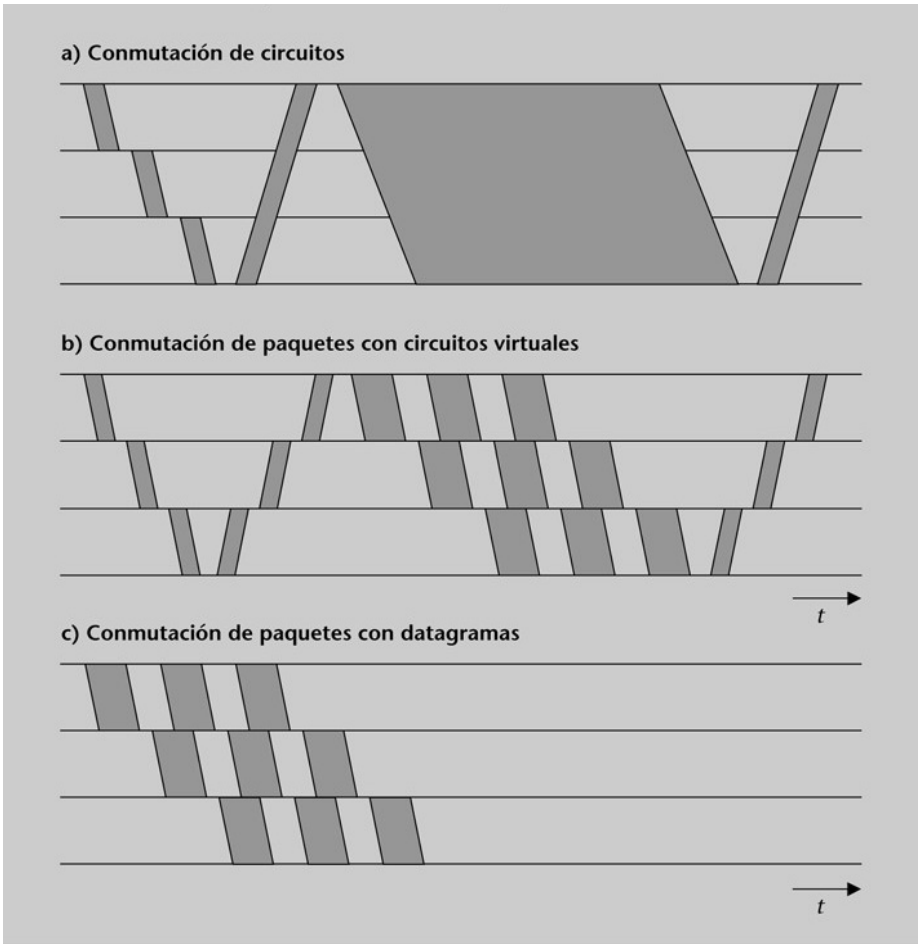
Aunque de entrada pueda parecer más conveniente que la red se encargue de todo y ofrezca un servicio orientado a conexión, muchas veces es preferible la flexibilidad y la sencillez de un servicio no orientado a conexión, dejando para niveles superiores al de red, en el caso de que sea necesario, las pesadas tareas del control de errores y de la reordenación. De esta manera, se traspasa trabajo de los nodos a las estaciones y la red puede operar de manera más eficiente. Además, en niveles superiores se dispone de los dos tipos de servicios y se puede elegir el más adecuado para cada aplicación.

## 2.3. Comparación entre conmutación de circuitos y conmutación de paquetes

Ahora que hemos visto los principios básicos de funcionamiento de las redes de conmutación de circuitos y de los dos tipos de redes de conmutación de paquetes (datagramas y circuitos virtuales), podemos resumir las características principales. Veámoslas en la tabla y en la figura siguientes:

| | **Circuitos** | **Paquetes** | |
|---|---|---|---|
| | | **Circuitos virtuales** | **Datagramas** |
| **Recursos** | Dedicados para toda la conexión | No dedicados | No dedicados |
| **Camino** | Lo mismo para toda la conexión | Lo mismo para todos los paquetes de la conexión | Puede ser diferente |
| **Transmisión** | Continua | En paquetes | En paquetes |
| **Establecimiento del camino previo a la transmisión** | Sí | Sí | No |
| **Retardo en la transmisión** | No | Sí | Sí |
| **Efectos de una sobrecarga** | Bloqueo | Bloqueo y aumento del retardo | Aumento del retardo |
| **Conversión de velocidad** | No posible | Posible | Posible |
| **Información de control añadida forzada por la red** | No | Sí | Sí |

Comparación entre las distintas técnicas de conmutación.




Comparación de los tres tipos de comunicación.

Esta figura pretende poner de manifiesto las diferencias de funcionamiento de las redes de conmutación de circuitos y de paquetes, pero no compara prestaciones. Esto sólo se puede hacer si se tienen en cuenta una serie de factores de la red como la topología, el tamaño, el nivel de carga, etc., que aquí no están reflejados.

- El esquema a) muestra una conexión mediante una red de conmutación de circuitos y las tres fases de las que se compone: establecimiento de llamada, información y liberación.
- El esquema b) muestra el funcionamiento de una red de conmutación de

paquetes con circuitos virtuales. También se pueden distinguir tres fases, pero la transferencia de información se hace con paquetes que sufren un retardo en cada nodo intermedio.

- El esquema c) muestra que la conmutación de datagramas no necesita la fase de establecimiento ni la de liberación.

### 2.4. *Encaminamiento*

El camino que seguirán los paquetes para llegar a su destino es fruto de las decisiones que toman los nodos de la red. Hay diferentes posibilidades con respecto a cuándo, dónde y cómo tomar estas decisiones, que se traducen en diferentes estrategias de encaminamiento:

1) La decisión de *cuándo* establece dos posibilidades que ya hemos visto:

- En la técnica de datagramas se decide para cada paquete.
- En la técnica de circuitos virtuales se decide durante el establecimiento del circuito.

2) La decisión de *dónde* establece tres posibilidades:

- En todos los nodos (encaminamiento distribuido).
- En el originador del paquete (encaminamiento de fuente).
- En un nodo especializado de la red (encaminamiento centralizado).

3) La decisión de *cómo*, es decir, qué criterio se aplica para elegir el camino, establece dos posibilidades:

- El camino más corto. En este caso, la ruta que se elige es la que incluye el menor número de saltos entre nodos. Se trata de un criterio bastante fácil de medir y pretende minimizar el número de recursos utilizados.
- El coste más bajo. El criterio anterior no tiene en cuenta las características particulares que pueden tener los enlaces entre nodos. El criterio del coste más bajo asigna a cada enlace un coste proporcional a alguna característica, como la velocidad de transmisión o el retardo medio en la cola de entrada, y busca la ruta que presenta el coste total más bajo. En el primer caso conseguimos maximizar el rendimiento de la red (o *throughput*), y en el segundo, minimizar el retardo total que experimentan los paquetes.

Un aspecto muy importante de las diferentes estrategias de encaminamiento que se han desarrollado es saber si se adaptan o no a cambios de la red, que pueden ser de topología o de carga. Este criterio permite clasificar las estrategias en dos grupos:

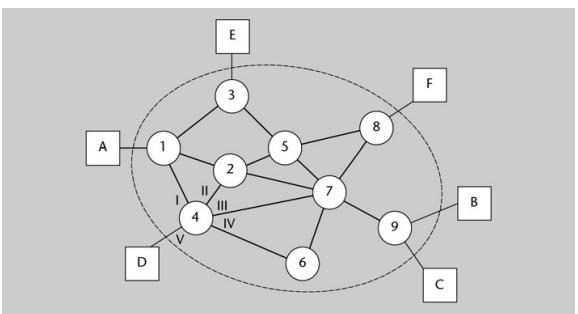
- Estrategias de encaminamiento adaptativas, que varían si hay cambios en la red.
- Estrategias de encaminamiento no adaptativas, que no varían a lo largo del tiempo.

Las segundas son más sencillas de implementar, pero en la práctica, hechos como la caída de un nodo o la sobrecarga de uno o distintos enlaces son bastante habituales, y esto hace que en la mayoría de los casos se utilice alguna estrategia adaptativa por la capacidad que tienen de reaccionar a estos cambios.

### *2.4.1. Encaminamiento no adaptativo. Encaminamiento estático*

La estrategia de encaminamiento estático consiste en decidir los caminos que se utilizarán para todas las conexiones posibles; es decir, para cada par de estaciones, en el momento en el que se crea la red.

Para decidir los caminos que se tienen que seguir, se puede aplicar tanto el criterio del coste mínimo como el del camino más corto. No es necesario guardar la información de los caminos enteros: sólo hay que saber, para cada nodo, cuál es el salto siguiente que deben hacer los paquetes, y cada nodo sólo es preciso que tenga información referente a sí mismo. Como ejemplo, supongamos que tenemos la red de la figura "Ejemplo de red de conmutación", y planteémonos cómo sería la tabla de encaminamiento del nodo 4:




Red con detalles de los puertos de salida del nodo 4.

Una posible tabla correspondiente a este nodo sería la siguiente:

| Destino | Puerto de salida |
|---|---|
| A | I |
| B | III |
| C | III |
| D | V |
| E | I |
| F | III |

Tabla del nodo 4.

Por cada paquete que recibe, el nodo consulta la tabla para saber qué tiene que hacer. Si, por ejemplo, recibe un paquete destinado a la estación B, la tabla le dice que debe enviarlo por el puerto III. Si recibe un paquete para la estación D, la tabla le dice que lo envíe por el puerto V, que es el que corresponde al enlace que lo une directamente a la misma.

La ventaja de este algoritmo es que resulta muy fácil de implementar, pero tiene un inconveniente muy grande: su inflexibilidad. No tiene en cuenta posibles caídas de nodos de la red, ni cambios de la topología, ni la evolución del tráfico.

Hay una variante de este algoritmo que pretende mitigar la inflexibilidad de manera que la carga se pueda repartir entre diferentes enlaces, y la red, en general, sea menos vulnerable a la caída de enlaces o de nodos: el *algoritmo de caminos múltiples*. Este algoritmo se caracteriza por el hecho de que las tablas, en lugar de indicar una sola salida posible, presentan diferentes opciones que el nodo elegirá según los criterios siguientes:

a) Con la filosofía del turno rotatorio o *round-robin* (es decir, rotatoriamente entre las distintas opciones).

b) De manera aleatoria.

c) De manera aleatoria, pero asignando a cada opción una probabilidad, para que las opciones más idóneas se elijan más a menudo.

*Flooding*

El encaminamiento por *flooding* consiste en que cada paquete que llega a un nodo se envía a todos sus vecinos.

Literalmente, cuando se quiere enviar un paquete de un punto a otro de la red, ésta es inundada con réplicas del paquete.

Este algoritmo es muy sencillo. No requiere el uso de tablas y, además, soporta de manera óptima la caída de nodos o los cambios de la topología. El inconveniente principal es que la red se carga mucho porque circulan múltiples copias de todos los paquetes simultáneamente.

**Gestión de bucles cerrados**

Si la red tiene bucles cerrados, y no se toman medidas, se colapsa enseguida, porque se generan copias de paquetes continuamente. Una manera sencilla de evitarlo es poner un contador de saltos en cada paquete. Se puede inicializar a un valor máximo de la red, como por ejemplo el diámetro (definido como la longitud del camino más largo de entre todos los de mínimo número de saltos), y cada vez que un nodo conmuta un paquete hace que el contador disminuya una unidad. Cuando el contador llega a cero, el nodo lo descarta.

Los receptores pueden recibir muchas copias del mismo paquete. Por lo tanto, se tienen que identificar y en el momento de la recepción hay que descartar todos los que sean iguales a uno que ya se haya recibido. Este identificador debe ser único en toda la red. Lo más sencillo es que cada estación asigne a los paquetes un número a partir de un contador, y que el identificador único sea la concatenación de la dirección del nodo de origen más este número.

El *flooding* es un método muy robusto porque se prueban todas las rutas posibles entre el origen y el destino. Aunque un segmento de la red desaparezca, el resto de los nodos puede seguir en comunicación mientras haya un camino posible. Por ello se puede utilizar en situaciones en las cuales la robustez es importante, como aplicaciones militares o redes cuya topología sea muy cambiante.

Al probarse todos los caminos posibles para cada conexión, es seguro que se encuentra el de mínimo retardo, lo cual confiere a esta estrategia una utilidad interesante: se puede utilizar para establecer un circuito virtual entre dos nodos a través del camino más corto.

Una situació en la que se utiliza el *flooding* porque la topología es muy cambiante es en una red que no es de gran alcance: la interconexión de LAN con puentes (*bridges*).

### *2.4.2. Encaminamiento adaptativo*

Para que los algoritmos se puedan adaptar a las condiciones de trabajo, es necesario que los nodos se intercambien información sobre el estado de la red. Esto se convierte en un compromiso porque, cuanta más información se intercambien y más frecuentemente, mejores serán las decisiones que se tomen, pero también será mayor el tráfico extra que se introduce en la red y, por lo tanto, el rendimiento será peor.

Una buena manera de clasificar las estrategias de encaminamiento adaptativo es en función de dónde se obtiene la información: del mismo nodo, de todos los nodos de la red o de los nodos vecinos. Consideremos cada una de estas estrategias:

1) La primera estrategia casi no se utiliza, porque no tiene en cuenta el estado del resto de los nodos de la red y, por lo tanto, las decisiones que se toman se suelen adaptar a los cambios que se puedan dar fuera del nodo considerado.

2) La segunda estrategia no se suele aplicar en redes de un gran número de nodos porque el tráfico extra de información de encaminamiento a intercambiar puede ser excesivo.

3) En la práctica, la mayor parte de las estrategias de encaminamiento que se utilizan son variantes de la tercera posibilidad. De hecho, aunque la información que reciba un nodo en un instante proceda sólo de sus vecinos inmediatos, de alguna manera también está recibiendo información del resto de los nodos porque su vecino anteriormente la había recibido de sus otros vecinos. La diferencia radica en el tiempo que necesita un nodo para tener información de la totalidad de la red: si este tiempo es largo, puede pasar que la adaptación a los cambios sea muy lenta.

A continuación veremos la estrategia de encaminamiento adaptativo distribuido, basada en esta tercera posibilidad.

*Encaminamiento adaptativo distribuido*

El encaminamiento adaptativo distribuido se basa en que los nodos elaboran las tablas de encaminamiento a partir de la información que reciben periódicamente de sus vecinos inmediatos. La información se basa en un parámetro de calidad de la red, el que viene definido por el criterio de selección de caminos. Las tablas de encaminamiento se actualizan en cada intercambio de información, para contener siempre los mejores caminos posibles según el parámetro de calidad elegido.

Veamos el funcionamiento de esta estrategia con un ejemplo.

Supongamos que tenemos la red de la figura "Red con detalle de los puertos de salida del nodo 4". Cada nodo mantiene una tabla en la que figuran, para cada estación posible de destino, el puerto de salida y el parámetro de calidad.

Si se utiliza el número de saltos como parámetro de calidad, se puede suponer que el nodo 4 tendrá, en algún momento, la tabla de encaminamiento siguiente:

| Destino | Puerto de salida | Saltos |
|---|---|---|
| A | I | 2 |
| B | III | 3 |
| C | III | 3 |
| D | V | 1 |
| E | II | 4 |
| F | III | 3 |

Tabla de encaminamiento del nodo 4.

Esta tabla dice, por ejemplo, que para llegar a la estación E, los paquetes deben salir por el puerto II y que la distancia hasta la estación de destino es 4.

Cada nodo tiene que ser capaz de medir la distancia hasta sus vecinos. En el caso de utilizar el criterio del camino más corto, esta distancia siempre vale 1. En el caso de utilizar el retardo, una forma de medir esta distancia puede ser la ocupación en bits de la cola de salida correspondiente al enlace, dividida por la velocidad de transmisión, $v_t$.

Para poder adaptarse a posibles cambios que hayan sucedido, cada cierto intervalo fijo de tiempo los nodos intercambian con sus vecinos las tablas con las distancias hasta las estaciones de destino.

Puesto que el nodo 4 conoce la distancia a la que se encuentra de sus vecinos, puede complementar las tablas recibidas añadiéndoles estas distancias. De esta manera, conocerá la distancia que le separa de cada estación. Finalmente, se comparan los valores antiguos con los nuevos y, si se obtiene una ruta mejor, se actualiza la tabla.

Por ejemplo, supongamos que el nodo 4 recibe de su vecino 1 la información siguiente:

| Destino | Saltos |
|---|---|
| A | 1 |
| B | 4 |
| C | 4 |
| D | 2 |
| E | 2 |
| F | 4 |

A partir de esta información, sabe que la distancia hasta la estación E, a través del nodo 1, es de 2 (según la tabla) más 1 (distancia desde 4 hasta 1). Por lo tanto, la distancia total hasta la estación E, calculada a través del nodo 1, es de 3, mientras que la información original era la siguiente: distancia 4 a través del nodo 2. El nodo 4 modifica la entrada de su tabla correspondiente a la estación E.

A partir de este momento, los paquetes que reciba el nodo 4 para la estación E se enviarán al nodo 1, porque es el camino más corto. La nueva tabla de encaminamiento asociada al nodo 4 es la que vemos a continuación:

| Destino | Puerto de salida | Saltos |
|---|---|---|
| A | I | 2 |
| B | III | 3 |
| C | III | 3 |
| D | V | 1 |
| E | I | 3 |
| F | III | 3 |

Tabla de encaminamiento del nodo 4.

**El encaminamiento en Internet**

La estrategia de encaminamiento adaptativo distribuido se diseñó para ARPANET, la red de conmutación de paquetes precursora de la actual Internet, y que utilizaba como criterio de selección la estimación de retardos en lugar del número de saltos.

Al cabo de unos cuantos años de experiencia, y al evidenciarse sus inconvenientes, fue sustituida por otra muy diferente. Ésta, sin embargo, hacía empeorar el rendimiento a medida que aumentaba la carga de la red y, por ello, hubo un tercer cambio en el que se varió la función que calcula los retardos previstos.

Por este motivo, se dice que hoy estamos en la tercera generación de Internet, en lo que respecta a estrategias de encaminamiento.

### *2.4.3. Encaminamiento jerárquico*

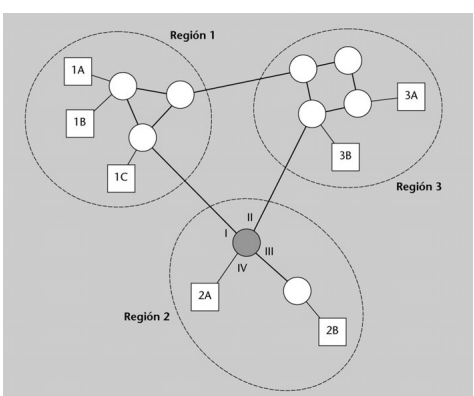
Todas las estrategias que hemos visto hasta ahora tienen un mismo problema: si la red crece, las tablas de cada nodo crecen y el trabajo para mantenerlas actualizadas es cada vez más pesado.

Una posible solución es el encaminamiento jerárquico.

El encaminamiento jerárquico consiste en dividir la red en regiones, que también se denominan *subredes*.

Con esta estrategia, las tablas de cada nodo sólo necesita tener una entrada para todas las estaciones que pertenecen a una región diferente de la propia. Incluir en la dirección de la estación una referencia a la región a la que pertenece hace más rápida la consulta de las tablas y, por lo tanto, el encaminamiento de los paquetes.

Consideremos un ejemplo en el que tenemos una red dividida en subredes, como se ve en la figura siguiente:




Direccionamiento jerárquico.

Ésta sería la tabla correspondiente al nodo marcado, si no se estableciera la subdivisión en regiones:

| Destino | Puerto de salida |
|---|---|
| 1A | I |
| 1B | I |
| 1C | I |
| 2A | IV |
| 2B | III |
| 3A | II |
| 3B | II |

Ahora bien, si se agrupan las estaciones en regiones, la tabla del mismo nodo quedaría de la manera siguiente:

| Destino | Puerto de salida |
|---|---|
| 1 | I |
| 2A | IV |
| 2B | III |
| 3 | II |

En este ejemplo las direcciones se han designado siguiendo el mismo esquema jerárquico (las estaciones de la región 1 se denominan 1A, 1B, etc.).

Para todas las estaciones que pertenecen a la región 1 o a la 3, sólo hay que mirar una entrada, la que corresponde a su región. Si se añaden estaciones a las regiones 1 y 3, las tablas de los nodos de la región 2 no cambian.

**Encaminamiento jerárquico de la red telefónica**

La red telefónica utiliza un esquema parecido en la numeración de los abonados. Los números de teléfono están compuestos de grupos de cifras que corresponden a diferentes "regiones": el que corresponde a la provincia (o al grupo "móviles"), a la central local y al abonado.

Más adelante veremos que el protocolo IP también utiliza un encaminamiento jerárquico; las direcciones de los ordenadores están compuestas de dos partes: el identificador de la subred en la que se encuentran y el identificador del ordenador dentro de la subred.

## 3. Arquitecturas de protocolos: el modelo OSI

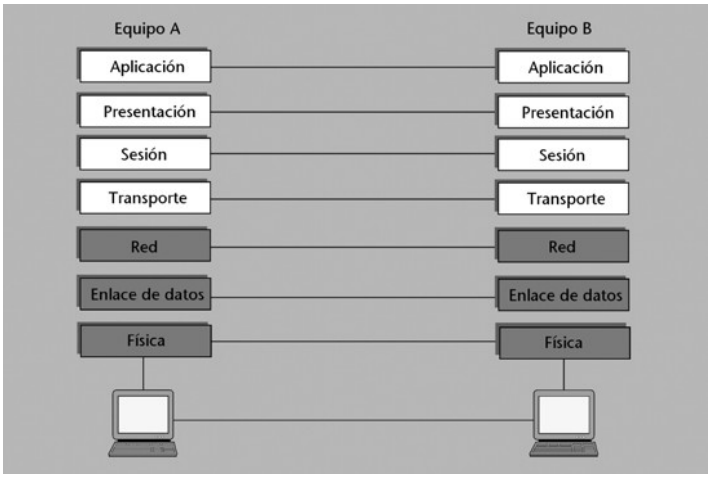
Como ya hemos comentado, cuando el CCITT y la ISO propusieron la torre OSI, en el mercado había muchas arquitecturas de protocolos, unas propietarias, otras abiertas, pero todas diferentes. La torre OSI pretendía ser un modelo básico de referencia, un marco para el desarrollo de estándares que permitieran la interoperabilidad completa. Diferentes razones han hecho que este modelo, así como las normas que del mismo se derivan, no hayan tenido la repercusión que se esperaba, entre las que destacan las siguientes:

- La complejidad del modelo, innecesaria en muchos casos.
- La complejidad de las normas desarrolladas a partir del modelo.
- El impulso del modelo Internet y la simplicidad de sus estándares.

A pesar de que el modelo OSI no se haya impuesto en los desarrollos, es muy útil como referencia para explicar qué debe hacerse y cómo. El hecho de que sea tan completo y cartesiano lo hace muy interesante para la pedagogía de los conceptos básicos de redes, y las arquitecturas que en realidad se utilizan se explican estableciendo una relación constante con el modelo OSI. Por ello, en este apartado explicamos los siete niveles de la torre OSI. A partir del capítulo siguiente, sin embargo, nos centraremos en la arquitectura TCP/IP, la que constituye la red Internet.

### 3.1. Definición

El modelo básico de referencia OSI, o simplemente modelo OSI, afronta el problema de las comunicaciones de datos y las redes informáticas dividiéndolo en niveles. Cada participante de la comunicación incorpora como mínimo uno de los mismos, y los equipos terminales los incorporan todos.




Los niveles de la torre se comunican en dos direcciones:

- Horizontal. La comunicación horizontal sólo se da entre niveles homónimos. Se podría pensar –y de hecho es así– que todo el nivel constituye un único sistema distribuido que tiene un representante en cada uno de los equipos. Un protocolo de nivel *i* (en el que *i* es el identificador del nivel correspondiente) especifica el formato, el significado y la temporización de la información que circula entre los miembros de este sistema distribuido.
- Vertical. La comunicación vertical sólo se da entre niveles adyacentes de un mismo sistema. Este tipo de comunicación posee un carácter totalmente local; es decir, puede materializarse por mecanismos de software (llamadas a liberías, comunicación entre procesos, etc.). De manera genérica, denominaremos estos mecanismos *servicio de nivel i* (en el que *i* es el identificador del nivel que proporciona el servicio, e $i + 1$, el nivel que lo utiliza).

## 3.2. Los protocolos

Con los protocolos se pretende la intercomunicación de entidades situadas en máquinas diferentes.

Entendemos por entidad un sistema electrónico y/o informático, ubicado dentro de un nivel del modelo OSI, que, en combinación con las otras entidades del mismo nivel situadas en otros sistemas, forma un todo (un sistema distribuido).

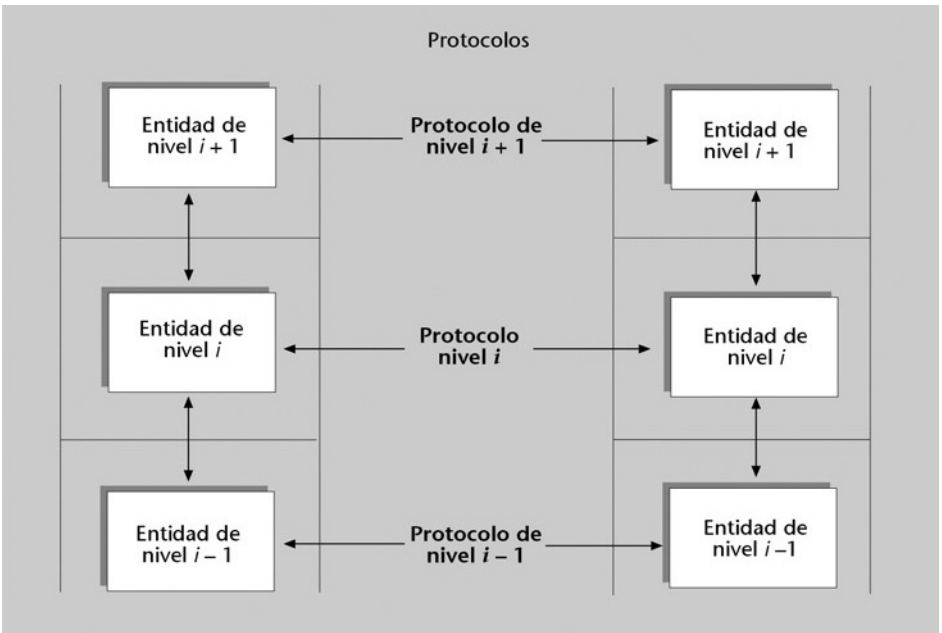
Por tanto, la especificación del protocolo que utilizamos debe llevarse a cabo en un estándar claramente definido que permita a desarrolladores que no trabajan juntos implementarlo de manera totalmente idéntica.

La recepción de una secuencia de bits en un momento inesperado o de una longitud incorrecta, o en una disposición imprevista, puede hacer que la entidad destinataria no reaccione correctamente y deje de inmediato el nivel (las dos entidades que lo forman) en una situación inestable.

Evidentemente, ello no se puede permitir. Por este motivo, la implementación del protocolo debe ser extremamente esmerada y, por consiguiente, también la especificación del estándar.

En un sistema encontramos tantos protocolos como niveles lo formen.

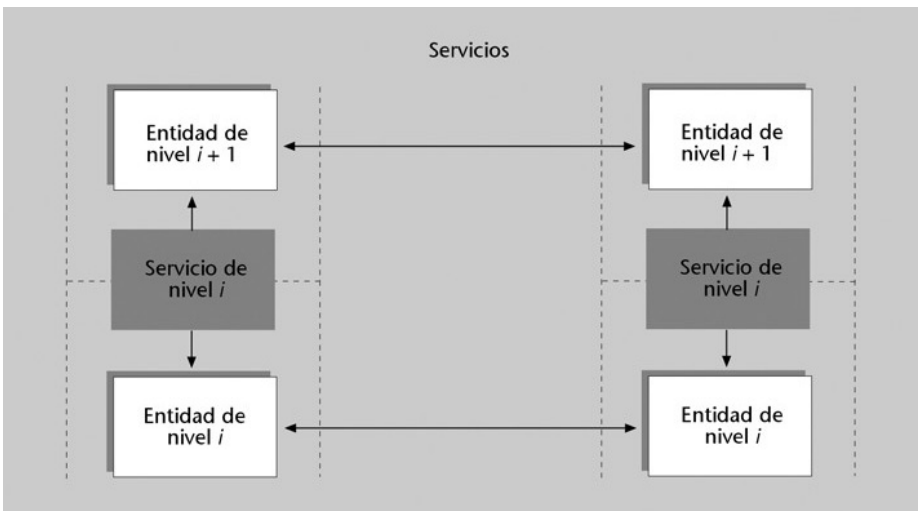
Los sistemas a los que se conecte directamente deberán tener la misma especificación que los estándares para todos los niveles que implemente el protocolo.



## 3.3. Los servicios

La especificación de un servicio es siempre menos estricta que la de un protocolo. Por *servicio* entendemos la comunicación que se produce dentro de una misma máquina y, por consiguiente, dentro de un único ámbito de responsabilidad. La funcionalidad de las interfaces de cada uno de los niveles (y, por tanto, de las entidades que la implementan), la determinarán los estándares que utilicen; sin embargo, su especificación precisa no es relevante para los estándares involucrados. Cada sistema individual puede materializarlas de una manera u otra según convenga.

Sea como sea, la cantidad de papel que ocupa la descripción de un servicio siempre será muy inferior al que ocupa la especificación de un protocolo.




## 3.4. Los siete niveles del modelo OSI

### *3.4.1. Nivel físico*

El nivel físico se encarga de las tareas de transmisión física de las señales eléctricas (o electromagnéticas) entre los diferentes sistemas. Las limitaciones del nivel físico (equipos de transmisión y recepción, medios de transmisión, ampli-

ficadores, etc.) imponen otras al resto del sistema: por un lado, limitan la velocidad de transmisión (en bits por segundo) y, por otro, hacen aparecer una probabilidad de error, el porcentaje de bits erróneos que llegan a destino.

La primera limitación es casi insalvable partiendo de un medio de transmisión dado, puesto que los parámetros físicos de este último imponen un límite superior no superable por medio de una mejora tecnológica. Los medios de transmisión poseen una capacidad de transmisión acotada y la electrónica que utilizamos para llevar a cabo las transmisiones en los mismos puede mejorar la velocidad de transmisión, pero no puede superar este límite. Esta limitación viene dada por el ancho de banda, o ancho del espectro eléctrico, que puede atravesar el medio de transmisión (doblar el ancho de banda significa que se puede doblar la velocidad de transmisión) y por la imposibilidad práctica de recibir la señal libre de cualquier interferencia.

La probabilidad de error puede corregirse por medio de algoritmos y protocolos si su valor está contenido. Si las cotas de error son inferiores al 1%, se puede reducir su impacto si se agrupan los bits en pequeños bloques de datos (de este modo, se minimiza la probabilidad de que haya bits erróneos dentro del bloque) y se añade software en el receptor que vigile la corrección de los datos recibidos.

### *3.4.2. Nivel de enlace*

El nivel de enlace proporciona un servicio similar al nivel físico, mejorando las características de fiabilidad de la transmisión. Añade bits adicionales a los que forman el mensaje para poder detectar errores de transmisión en el mismo y poder pedir su retransmisión. Para ello, es preciso conferir una estructura a los bits: se agrupan en bloques denominados tramas, que contienen los bits de mensaje, los bits añadidos para detectar errores y diferentes campos de control, tales como el número de trama.

El transmisor calcula estos bits adicionales a partir del resto por medio de una operación que el receptor conoce y aplica igualmente. Si el receptor detecta una discrepancia entre los bits adicionales (redundantes) y los que ha calculado a partir del resto, detecta que el bloque es erróneo y pedirá una retransmisión del mismo.

La adición de los bits redundantes y su comparación en recepción se denomina *detección de errores*. Los procedimientos de corrección a partir de dicha detección se conocen como *control de errores*.

Además de esta tarea de control de errores, el nivel de enlace lleva a cabo otra importante: el *control de flujo*.

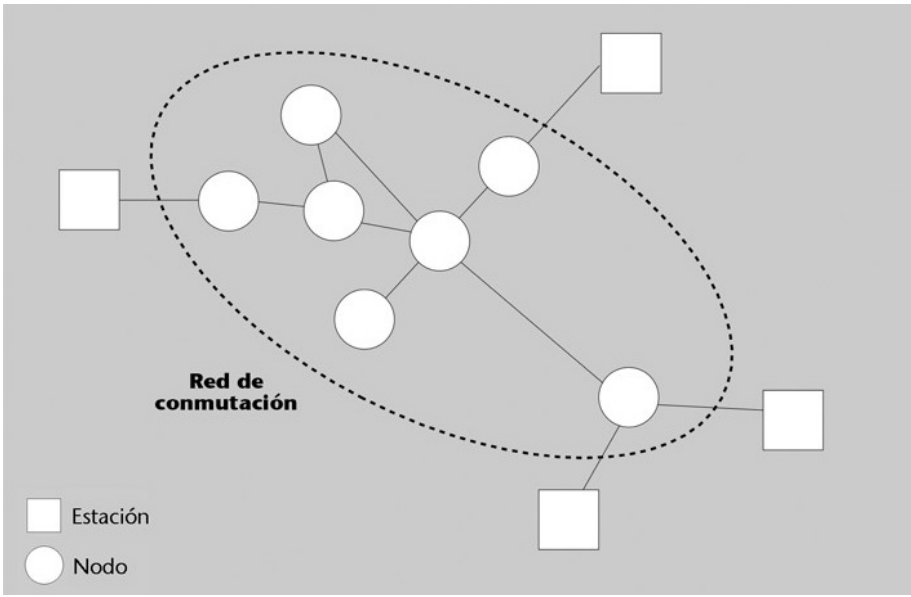
El receptor debe procesar las tramas a medida que las recibe. En algunos casos, este proceso comporta un gasto de tiempo mínimo, teniendo en cuenta la velocidad de transmisión (por ejemplo, guardar los datos en disco); sin embargo, en otros, puede ser costoso. En esta situación, el receptor necesita un mecanismo que notifique al transmisor que momentáneamente detenga la transmisión con el objetivo de disponer del tiempo necesario para llevar a cabo esta tarea.

El nivel de enlace no sólo sirve para controlar líneas punto a punto, sino también para controlar líneas compartidas por diferentes terminales (redes de área local).

### *3.4.3. Nivel de red*

Ya hemos visto que las redes de conmutación de paquetes constituyen el tipo de red más eficiente para transmitir datos desde diferentes puntos de vista: uso de recursos, coste, capacidad de mantener distintas conexiones simultáneas, etc. El modelo OSI, por tanto, sólo habla de redes de conmutación de paquetes.

En el nivel de red se distingue entre estaciones terminales y nodos de conmutación.



Los nodos de conmutación disponen de diferentes enlaces hacia otros nodos o hacia terminales, y son los que permiten que los paquetes viajen por la red desde una estación terminal a otra.

Existen dos tipos de redes de conmutación de paquetes:

- Redes que funcionan en modo datagrama. Podríamos decir que este tipo de redes son las básicas, puesto que incorporan la funcionalidad mínima para que un grupo de nodos y de terminales interconectados puedan hacer pasar información de un punto a otro.

  El problema de las redes en modo datagrama radica en la dificultad de garantizar la entrega correcta y completa de la información, puesto que los diferentes paquetes que forman la transmisión no mantienen un vínculo conocido por la red. Los paquetes pueden llegar fuera de orden, duplicados, o incluso se pueden perder sin que la red pueda hacer gran cosa al respecto. Se deja al terminal receptor la responsabilidad de restaurar los posibles daños que haya tenido el paquete durante la transmisión.
- Redes que funcionan en modo circuito virtual. Estas redes pueden garantizar que la entrega de los paquetes sea correcta y completa, y lo hacen aportando el concepto de conexión propio de las redes de conmutación de circuitos. Es el circuito virtual. Este último permite agrupar los paquetes relacionados de manera que el receptor los recibe correctamente sin problemas de orden, pérdida o duplicación.

La asignación de direcciones (*addressing*) es uno de los conceptos básicos del nivel de red. Permite a la red, como sistema distribuido pero único, decidir cuál de los múltiples terminales es el destinatario final.

El encaminamiento (*routing*) constituye el procedimiento que permite a este sistema distribuido conducir la información por los diferentes nodos de origen a destino, minimizando el trayecto y el tiempo de tránsito, optimizando recursos, etc.

### *3.4.4. Nivel de transporte*

El nivel de transporte permite una conexión fiable sobre cualquier tipo de red (fiable o no). En las redes de conmutación de paquetes en modo datagrama es donde este nivel se muestra más importante, puesto que es el responsable de controlar las posibles deficiencias de las transmisiones.

Es importante distinguir una red en modo datagrama con un nivel de transporte de una red en modo circuito virtual sin nivel de transporte (o con un nivel de transporte nulo). En las redes en modo circuito virtual sin nivel de transporte, este nivel y los superiores sólo se implementan en los terminales y no en los nodos. Este punto es capital, puesto que permite que redes "simples" (en modo datagrama) funcionen también como redes más complejas (en modo circuito virtual) simplemente añadiéndole funcionalidad a los extremos. Internet constituye un ejemplo de este razonamiento.

La función principal de este nivel consiste en asegurar la calidad de transmisión entre los terminales que utilizan la red, lo que implica recuperar errores, ordenar correctamente la información, ajustar la velocidad de transmisión de la información (control de flujo), etc.

### *3.4.5. Niveles de sesión, presentación y aplicación*

Estos tres niveles se suelen explicar de manera conjunta, puesto que existen pocos ejemplos prácticos de protocolos de sesión y de presentación y, asimismo, la arquitectura Internet delega todos los trabajos por encima de transporte a la aplicación. No obstante, en el modelo OSI están definidos como tres niveles diferentes e independientes, con atribuciones propias.

El nivel de sesión es, en teoría, el encargado de gestionar las conexiones de larga duración, la recuperación de caídas de red de manera transparente y los protocolos de sincronía entre aplicaciones.

El nivel de presentación se encarga de conseguir que las diferentes plataformas (sistemas operativos, procesadores, etc.) se puedan entender al conectarse por medio de una misma red. Dicho de otra manera, es el encargado de definir una manera universal de codificar la información. Dicha codificación puede poseer propiedades de eficiencia (por medio de la compresión, por ejemplo), propiedades de confidencialidad (por medio de la criptografía), etc.

En el nivel de aplicación residen los programas. En este nivel podemos encontrar servidores, clientes que acceden a estos últimos, aplicaciones que trabajan según un modelo simétrico (*peer-to-peer*), etc.

Capítulo II

# Transmisión de datos

*Llorenç Cerdà Alabern*
*Jordi Íñigo Griera*

## Introducción

En este capítulo se expondrá la parte física de la comunicación entre ordenadores; es decir, todo aquello que tiene relación con la generación de señales electromagnéticas que pueden ser transmitidas con medios de transmisión reales, como por ejemplo el par de hilos, el cable coaxial o la fibra óptica.

Repasaremos los conceptos básicos de la transmisión de datos, desde un punto de vista más formal que analítico y suponiendo que el lector dispone de los fundamentos físicos asociados a las señales electromagnéticas.

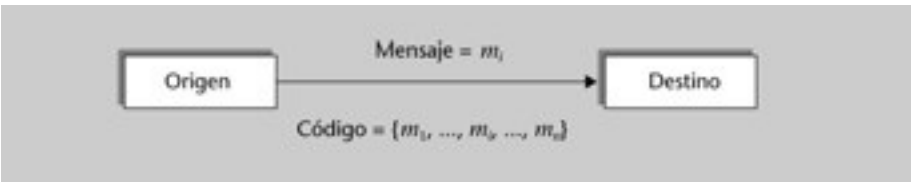
## 1. Conceptos básicos de transmisión de datos

Cuando se habla de transmisión de datos, se supone que estos datos son información digital. Esto quiere decir que la podemos representar con una secuencia de símbolos diferentes elegidos de un conjunto finito. Puede ser tanto la versión digitalizada de fenómenos analógicos como los mensajes representables mediante un conjunto finito de cifras, letras o palabras. El único requisito es que el conjunto básico de cifras, letras o palabras sea finito, aunque el resultado de sus combinaciones no hace falta que lo sea (y, en general, no lo será).

Por información digital entendemos cualquier información del tipo "sí" o "no", y, por extensión, información del tipo "un valor de entre un conjunto finito".

Para cada uno de estos "átomos" de comunicación (muestras, cifras, letras, etc.), utilizaremos el nombre abstracto de *palabra código*. El conjunto de todas las palabras código lo denominaremos *tabla del código*, o simplemente *código*.

Por lo tanto, nuestra información estará compuesta por una secuencia de longitud indefinida, o no, de palabras código. El resultado de la concatenación de palabras código en una secuencia establecida por el origen es lo que denominaremos mensaje.




Esquema de la transmisión de información.

Para conseguir la transmisión de la información sólo necesitamos dos cosas:

1) Transmitir el mensaje, lo cual se consigue simplemente transmitiendo la secuencia de palabras código que lo forman. Cada una de estas palabras código normalmente está representada por una secuencia de bits (secuencia binaria).
2) Que ambos extremos, que denominaremos origen y destino, compartan el mismo significado para todas y cada una de las palabras código del código que utilicen[1].

Para poder transmitir información, tenemos que estar seguros de que todo el mundo comparte unos conocimientos previos, que tienen que haber sido aprendidos por mecanismos diferentes de los de la transmisión de la información. Si alguno de los participantes de la transmisión discrepa en el significado de alguna de las palabras código del código utilizado, se producirán malas interpretaciones de la información transmitida.

1. Esto equivale a decir que ambos extremos tienen que utilizar el mismo "diccionario", con las mismas palabras (palabras código), y que el significado de cada uno es compartido.

Cuando origen y destino utilizan el mismo código y el significado dado a cada una de las diferentes palabras código es estrictamente el mismo, decimos que transmisor y receptor comparten una misma codificación de fuente. En la figura siguiente describimos esta situación de manera esquemática:




**Ejemplos de codificación de fuente**

A continuación presentamos un par de ejemplos de codificación de fuente que os pueden servir para entender mejor el concepto:

**1) La tabla ASCII**

La tabla ASCII (*American Standard Code for Information Interchange*) es quizá el ejemplo más típico de codificación de fuente. Se trata de un código con 128 palabras, que se corresponden con otras tantas letras (mayúsculas ['A' … 'Z'], minúsculas ['a' … 'z''), cifras ['0' … '9'], signos de puntuación [':', '?', '!'…], códigos de control de transmisión [<stx>, <syn> …], edición [<retorno>, <tabulador> …] y miscelánea ['$', '@', '+'…]).

Para codificar las 128 palabras código se utilizan 7 bits. En el código que presentamos a continuación se utilizan las 128 combinaciones de 7 bits posibles. Usar o no todas las combinaciones posibles de bits para implementar un código tiene sus implicaciones.

**Tabla ASCII**

| | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **A** | **B** | **C** | **D** | **E** | **F** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | *NUL* | *SOH* | *STX* | *ETX* | *EOT* | *ENQ* | *ACK* | *BEL* | *BS* | *HT* | *LF* | *VT* | *FF* | *CR* | *SO* | *SI* |
| **1** | *DLE* | *DC1* | *DC2* | *DC3* | *DC4* | *NAK* | *SYN* | *ETB* | *CAN* | *EM* | *SUB* | *ESC* | *FS* | *GS* | *RS* | *US* |
| **2** | SP | ! | " | # | $ | % | & | ' | ( | ) | * | + | , | - | . | / |
| **3** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | ? |
| **4** | @ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| **5** | P | Q | R | S | T | U | V | W | X | Y | Z | [ | \ | ] | ^ | _ |
| **6** | ` | a | b | c | d | e | f | g | H | i | j | k | l | m | n | o |
| **7** | p | q | r | s | t | u | v | w | X | y | z | { | \| | } | ~ | DEL |

La codificación se obtiene de la tabla de la manera siguiente: si tomamos el carácter 'R', vemos que está en la fila 5, columna 2. Ésta es la posición hexadecimal 0x52 y, por lo tanto, la 'R' se codifica con el código 0110010.

**2) Una codificación para quinielas futbolísticas**

No todos los ejemplos de codificación de fuente son de uso general. Podríamos definirlos también de uso particular.

Supongamos que queremos transmitir la información generada por las quinielas futbolísticas. En este caso, tenemos que cada vez que queremos enviar una quiniela debemos transmitir quince valores que pueden ser 1, X o 2. Un posible código sería el siguiente:

| Símbolo | Significado | Codificación |
|---|---|---|
| **1** | Victoria local | 0001 |
| **X** | Empate | 0010 |
| **2** | Victoria visitante | 0100 |

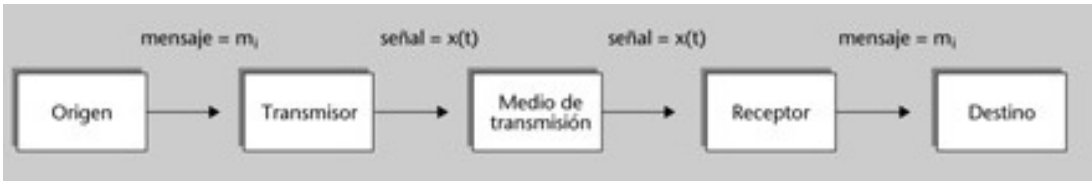
Código para quinielas.

Obsérvese que en este caso no hemos usado el mínimo número de bits para cada palabra código. Más adelante veremos el cálculo del número óptimo de bits que hay que utilizar y la utilidad de usar más.

## 1.1. Modelo sencillo de comunicaciones

Desde el punto de vista lógico, el modelo que se extrae de la explicación anterior es completo: tenemos un origen que genera un mensaje y un destino que lo interpreta (véase la figura anterior). Es evidente, sin embargo, que el nivel de abstracción es demasiado elevado para deducir los fenómenos físicos en los que se puede basar la comunicación de los mensajes. Por lo tanto, tenemos que refinar el modelo, y para hacerlo presentaremos paso a paso lo que denominaremos modelo sencillo de comunicaciones.

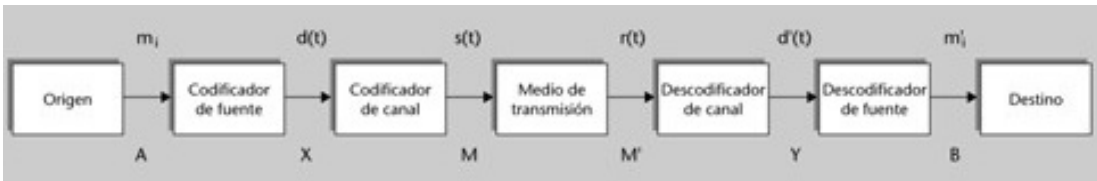
Lo primero que hay que constatar es que el mensaje generado llegará a destino gracias a un medio de transmisión (un par de hilos, ondas radiadas, etc.).




A continuación, nos tenemos que plantear que necesitamos algo más tangible que una secuencia de bits para poner en este medio, como por ejemplo electrones o fotones. A estos fotones, electrones, etc. los englobaremos dentro del nombre genérico de señal eléctrica o electromagnética, o simplemente señal. Estas señales se pueden representar de manera matemática como una función dependiente del tiempo $x(t)$.

Alguien debe convertir la secuencia de bits en las señales electromagnéticas adecuadas para el medio que tienen que atravesar: el codificador de canal[2]. Como es lógico, en el otro extremo de la comunicación tendremos un descodificador de canal, que extrae de la señal recibida la secuencia de bits transmitida.

Ahora ya tenemos el modelo completo:




*mi* es el mensaje enviado, *d*(*t*) son los datos enviados, *s*(*t*) es la señal que se inyecta en el medio, *r*(*t*) es la señal recibida en la salida del medio, *d'*(*t*) son los datos recibidos y *m'i* es el mensaje recibido.

Lo que hay entre el punto X y el punto Y se denomina *circuito de datos*, y se utiliza este término cuando los detalles del codificador/descodificador de canal y medio de transmisión no son relevantes. El circuito de datos recibe una secuencia de bits en el origen y la libera en destino.

Análogamente, cuando el trabajo del codificador/descodificador de fuente tampoco importa, lo que hay entre A y B se denomina *enlace de datos*.

2. En algunos entornos se denomina transmisor al conjunto codificador de fuente/codificador de canal, y receptor al conjunto descodificador de canal/descodificador de fuente.

Hemos escrito *mi* y *m'i*, $d(t)$ y $d'(t)$, $s(t)$ y $r(t)$ porque desgraciadamente, en situaciones reales, el mensaje recibido no es exactamente igual al mensaje enviado. La razón principal es que los medios de transmisión (por su naturaleza y la distancia) están expuestos a perturbaciones que degradan la señal transmitida. Si la degradación es considerable, el descodificador de canal extrae una secuencia de bits diferente a la original que hace que el descodificador de fuente entregue en destino un mensaje equivocado. Cuando pasa esto decimos que ha habido un *error de transmisión*.

Podemos elegir medios de transmisión que degraden poco la señal y podemos diseñar los codificadores/descodificadores de canal de manera que minimicen la probabilidad de que se produzcan errores de transmisión, pero no podemos garantizar una degradación nula. Por ello, como ya hemos visto, una de las atribuciones del nivel de enlace (el nivel siguiente al modelo OSI) es luchar contra esta probabilidad de error. Es decir, asumimos que los circuitos de datos no son perfectos, que presentan una cierta probabilidad de error (a menudo perfectamente acotada) y que hay que establecer mecanismos dentro del origen/destino para luchar contra esta probabilidad.

El resto de este capítulo lo dedicaremos a estudiar con detalle los diferentes componentes del modelo sencillo de comunicaciones, de fuera hacia dentro:

1) La interfaz DTE-DCE. Para interconectar ordenadores se decidió que los codificadores/descodificadores de canal fueran dispositivos periféricos externos y normalizar de alguna manera los puntos de acceso. En este caso, el ordenador recibe el nombre de DTE (*Data Terminal Equipment*) y el dispositivo que recibe una secuencia de bits y la convierte en señales electromagnéticas susceptibles de atravesar un medio (y viceversa) recibe el nombre de DCE (*Data Communication Equipment*). En este apartado repasaremos las características de esta interfaz normalizada DTE-DCE y veremos un ejemplo paradigmático: la interfaz RS-232.
2) Los codificadores de canal. Veremos las características de los codificadores de canal, los tipos de señales que generan, la posibilidad de multiplexar diferentes secuencias de bits en la misma señal y un parámetro muy importante en comunicaciones: la velocidad máxima de transmisión.
3) El medio de transmisión. El medio de transmisión es fundamentalmente un sistema pasivo formado por un medio conductor de ondas eléctricas o electromagnéticas, que puede ser un material, el aire o incluso el vacío. A menudo se complementa con otros elementos activos para mejorar alguna de sus cualidades.

## 2. Interfaz DTE-DCE

La interfaz DTE-DCE se podría considerar un circuito de datos entre DTE y DCE. Sin embargo, de hecho, la distancia entre los dos equipos, normalmente muy corta, hace que las soluciones que se adopten sean muy diferentes de las que veremos en los circuitos de datos. Además, sería deseable que el DTE tuviera bastante con una única interfaz DTE-DCE para poder conectarse con diferentes tipos de DCE (apropiados para distintos tipos de medios de transmisión). Desgraciadamente, esto no es así, y en realidad nos encontramos con que los DTE tienen que llevar diferentes interfaces (DTE-DCE) para conectarse a diferentes DCE.

Las características que definen las interfaces DTE-DCE se pueden resumir en tres:

1) Transmisión serie o transmisión paralela. En una interfaz DTE-DCE serie, los bits se transmiten uno tras otro. En las interfaces paralelas los bits se transmiten en grupos, mediante diferentes hilos (circuitos). Las interfaces paralelas de 8 o 16 bits de ancho son habituales.
2) Mecanismo de sincronización (de bit, de byte, etc.). Usualmente, las interfaces DTE-DCE disponen de líneas dedicadas a la transmisión de uno o más relojes (transmisión, recepción, de bit, de byte). En este caso hablaremos de interfaces síncronas. En el caso de que la interfaz no tenga un reloj dedicado, diremos que es una interfaz asíncrona.
3) Velocidad de transmisión. Hay interfaces DTE-DCE apropiadas para alta velocidad y las hay adecuadas para baja velocidad.

Según los requerimientos del DCE, elegiremos una interfaz u otra. Para velocidades bajas, de hasta 10 kbps, se utiliza casi exclusivamente la interfaz RS-232. Para velocidades superiores, normalmente se usan las interfaces síncronas y paralelas.

### 2.1. RS-232

De entre todas las interfaces sólo estudiaremos la RS-232 por su alta disponibilidad (casi cualquier sistema informático dispone al menos de una).

La interfaz RS-232 es del tipo serie, para sistemas de baja velocidad, y asíncrona. Nosotros nos centraremos aquí exclusivamente en el formato que adopta la señal de transmisión.

**La interfaz RS-232**

Debemos tener en cuenta algunos detalles respecto de la interfaz RS-232:

El estándar prevé hasta 20 kbps de velocidad, aunque hay implementaciones que permiten alcanzar velocidades máximas de transmisión que van de 100 kbps a 1 Mbps.

El estándar también define un alcance de hasta 15 m.

La transmisión suele ser asíncrona, aunque el estándar prevé la posibilidad de trabajar con circuitos de reloj y adoptar, por lo tanto, la forma síncrona. Esta forma, sin embargo, no acostumbra a implementarse en circuitos estándar.

### *2.1.1. Transmisión asíncrona*

El término *transmisión asíncrona* simplemente alude al hecho de que no se transmite el reloj de *bit* con un circuito específico. Es un término que sólo tiene sentido en interfaces RS-232. por el bit de arranque. La codificación que utiliza RS-232 es del tipo NRZ, es decir, la señal toma un valor positivo concreto durante todo el tiempo de bit, si el bit es 1, y si el bit es 0, toma el valor 0.

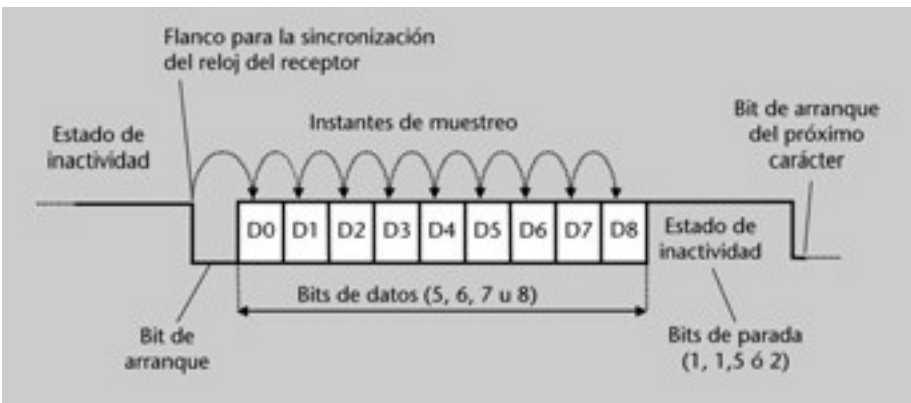
El sistema de transmisión asíncrona permite al receptor no perder el sincronismo de bit mientras el error no supere el 10% (de retardo o adelantamiento). Para poder funcionar, los paquetes se agrupan en caracteres, que pueden tener entre 5 y 8 bits de datos. Para definir este mecanismo de sincronización, tenemos que definir dos nuevos símbolos, que sumados a los dos que ya tenemos para los bits 1 y 0 hacen un total de cuatro símbolos.

- Bit 0: bit codificado con el símbolo positivo.
- Bit 1: bit codificado con el símbolo negativo.
- Estado de inactividad: estado en el que se encuentra el sistema desde que arranca. También encontraremos el estado de inactividad entre carácter y carácter. La codificación que utilizamos sólo permite dos niveles; por lo tanto, el estado de inactividad usa el mismo símbolo que uno de los bits.
- Señal de arranque: símbolo que permite separar el estado de inactividad del de un carácter. El nivel utilizado es el inverso de aquél del estado de inactividad, y ofrece un flanco que permite al receptor sincronizar su reloj (sin-

cronismo de *bit*) y saber dónde empieza el carácter (sincronismo de carácter). Esta señal tiene la misma duración que un bit, lo cual hace que normalmente se conozca como *bit* de arranque o bit de *start* (aunque no lleva ninguna información).

Una vez se ha enviado el bit de arranque, se envían los bits que forman el carácter de manera consecutiva. Tanto el receptor como el transmisor tendrán que haber establecido los bits que tiene cada carácter y la velocidad de transmisión.

Cuando todos los bits han sido enviados, la línea se pone en estado de inactividad. Para enviar el carácter siguiente, sólo hay que hacer preceder sus bits por el bit de arranque. En la figura de la página siguiente se muestra el funcionamiento de este mecanismo:




Carácter asíncrono en Interfaces RS-232.

Hay que tener en cuenta que, para que el flanco provocado por el bit de arranque sea "visible", es necesario que el estado de inactividad se mantenga durante un tiempo razonable (un tiempo de inactividad de cero segundos lo haría desaparecer). El tiempo mínimo de inactividad es el equivalente a un bit. Normalmente, este tiempo mínimo de inactividad se conoce como *bit* de parada o bit de *stop*.

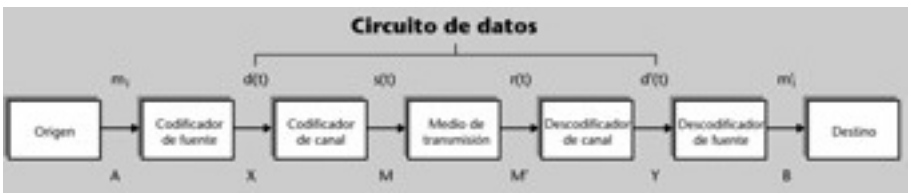
**Bits de parada en interfaces serie**

En algunas interfaces serie se pueden configurar hasta 2 bits de parada. Esto disminuye los caracteres por segundo que se pueden enviar, y dan a DCE lentos cierto tiempo para absorber los datos. En la actualidad ya sólo se utiliza un bit de parada.

## 3. Circuito de datos

Entendemos por *circuito de datos* el conjunto formado por el canal de datos (medio de transmisión, amplificadores y ecualizadores) y los equipos extremos para codificar $s(t)$ y descodificar $r(t)$.

La función básica del circuito de datos es la gestión de las señales eléctricas o electromagnéticas que se transmitirán. Recordemos la figura del modelo sencillo de comunicaciones. El codificador de canal inyectará una señal $s(t)$ al punto M y el descodificador la recuperará del punto M'.

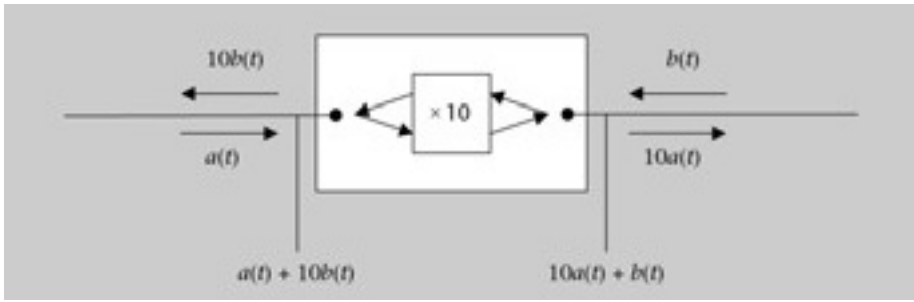



### 3.1. Circuitos unidireccionales y bidireccionales

Hasta ahora hemos hablado de un medio que transmitía señales de un punto a otro. El modelo que hemos utilizado era puramente unidireccional. Sin embargo, tanto las ondas eléctricas como las electromagnéticas se pueden cruzar dentro de un medio físico sin molestarse. Cualquier medio de transmisión es, por naturaleza, bidireccional.

Sin embargo, si ponemos amplificadores o cualquier otro equipo activo, como ya veremos, la cosa cambia, a pesar de que en muchos entornos sí que es técnicamente posible desarrollar equipos activos que sean capaces de trabajar con interfaces que sean entradas y salidas simultáneamente. El esquema de estos equipos podría ser el de la página siguiente:

Si os fijáis en la figura, veréis que el amplificador recibe, por una parte, la señal que hay que amplificar (en el esquema, $a(t)$) sumada con la señal que ya ha amplificado (en el esquema, $10b(t)$). Evidentemente, tiene que ser capaz de multiplicar sólo $a(t)$, si no el sistema se convertiría en inestable: $10b(t)$ se multiplicaría por 10 otra vez, y así sucesivamente. Esto también se puede aplicar a los (des)codificadores y a los ecualizadores.




Amplificador × 10 bidireccional.

Un circuito de datos cuyos equipos activos permiten todos el paso de señales simultáneamente en los dos sentidos se denomina *circuito bidireccional* o *circuito full duplex* (FDX).

Si uno o más de los equipos activos del circuito son unidireccionales, hablamos de *circuito unidireccional* o *circuito simplex*.

Los circuitos que pueden funcionar alternativamente en una dirección o bien en otra reciben el nombre de *circuitos half duplex* (HDX).

**Ejemplos de sistemas de transmisión**

a) Un ejemplo de transmisión simplex podría ser la conexión a una impresora (sólo recibe) o un ratón (sólo transmite).

b) Un ejemplo de sistema HDX son los walkie-talkies: cada aparato puede hablar y escuchar, pero no puede hacer las dos cosas simultáneamente.

En muchos casos, la manera más sencilla de obtener un circuito de datos FDX es utilizar dos circuitos de datos *simplex* simultáneamente y en sentidos opuestos.

## 3.2. Velocidad de transmisión y velocidad de modulación

La velocidad de transmisión ($v_t$), denominada también *ritmo de transmisión,* es el número de bits por unidad de tiempo que se generan, se reciben o se transmiten.

Normalmente, su forma de expresión es en bits/*s* o bps (bits *per second*).

Ara, debemos generar dos señales que puedan atravesar el medio, uno para cada bit posible (el 0 y el 1). Pero los medios permiten transmitir más de dos señales diferentes sin que se introduzcan errores. Por lo tanto, podemos generar, por ejemplo, cuatro símbolos diferentes, y asignar al primero la combinación 00; al segundo, la 01; al tercero, la 10, y al cuarto, la 11.

Cuantas más señales diferentes, más bits transmitimos por unidad de tiempo.

Supongamos que tenemos un conjunto de N señales:

$$señal_1, señal_2..., señal_N.$$

Con un conjunto de *N* señales podemos asignar a cada símbolo un número de bits dado por:

$$n = \log_2 N,$$

donde *n* son los bits de datos que puede llevar cada una de las señales. Denominamos *símbolos* a cada una de estas señales.

Una codificación de canal determinada se distingue por un grupo de *N* símbolos específicos, diferentes y preestablecidos para los dos DCE.

Cuanto mayor es el número de símbolos disponibles, mayor es el número de bits que puede transportar cada uno, según la expresión de *n* que acabamos de ver.

La velocidad de modulación ($v_m$), denominada también *ritmo de modulación*, es el número de símbolos por unidad de tiempo que se generan, se reciben o se transmiten. Normalmente, se expresa en símbolos/s o baudios.

La relación entre las dos velocidades viene dada por la *n*:

$$v_t = v_m \cdot n = v_m \cdot \log_2 N$$

La velocidad de transmisión se ve en la entrada del codificador de canal, mientras que la velocidad de modulación es la que se ve en la entrada del medio de transmisión.

### *3.2.1. Límite de la velocidad de modulación: criterio de Nyquist*

El criterio de Nyquist identifica la velocidad de modulación máxima a la que se puede transmitir a través de un medio sin que la distorsión provoque una interferencia intersimbólica (ISI) tan grave que no permita al DCE receptor extraer los datos.

El número de símbolos que podemos transmitir por una línea está limitado por su ancho de banda.

El criterio de Nyquist nos dice que, si no queremos tener interferencia intersimbólica (ISI), se tiene que verificar la relación siguiente:

$$V_m \text{ máx} = 2BW$$

Según este criterio, si doblamos el ancho de banda, doblamos la velocidad de modulación máxima.

El criterio de Nyquist nos da un límite máximo teórico. En la realidad encontraremos que los diferentes tipos de codificación de fuente aprovecharán más o menos este límite.

### *3.2.2. Límite de la velocidad de transmisión: teorema de Shannon*

El criterio de Nyquist no limita la velocidad de transmisión alcanzable con una línea de transmisión dada:

$$V_t|_{\text{máx}} = V_t|_{\text{máx}} \cdot n|_{\text{máx}} = 2BW \cdot n|_{\text{máx}}$$

Al ver la fórmula anterior se puede pensar que fijada una BW por Nyquist, haciendo $n$ muy grande se pueden obtener velocidades de transmisión muy grandes.

Ciertamente, según la tecnología usada, se pueden alcanzar $n$ grandes; pero no infinitamente grandes, por culpa del ruido en recepción. A partir de cierto umbral nos encontramos con que el ruido añadido puede hacer que el receptor crea que el símbolo que le han enviado no es el que realmente le habían enviado. Al confundir el símbolo, el receptor descodificará una secuencia de uno o más bits diferentes de los originales y se produce un error de transmisión.

La combinación del límite que impone el ancho de banda del medio y la robustez frente al ruido define la capacidad del canal (C), que incluye tanto el efecto del medio de transmisión como del codificador de canal y que identifica la velocidad de transmisión máxima:

$$v_t \leq C$$

Shannon, en su teoría de la información, enunció un teorema que da una expresión para la capacidad de un canal de datos y, por lo tanto, fija la velocidad máxima de transmisión (en bps) sobre este canal:

Teorema de Shannon:

$$C = BW\log_2\left(\frac{PS}{PN} + 1\right)$$

El parámetro que pone de manifiesto el efecto del ruido sobre la capacidad es la relación seña ruido, *PS*/*PN*., o también *S*/*N* o *SNR*. Este parámetro da la proporción entre la potencia de señal y la potencia de ruido en un punto, que normalmente será el receptor o descodificador. Por lo tanto, no se trata de no tener poco ruido, en términos absolutos, sino en términos relativos. Es decir, que la proporción de la potencia de señal respecto de la de ruido sea elevada.

El teorema de Shannon es un límite teórico insuperable. Es tecnológicamente imposible conseguir trabajar de manera sostenida a velocidades de transmisión superiores a la capacidad de la línea. La única manera de transmitir a un ritmo más alto es variar alguno de los parámetros de la ecuación: aumentar el ancho de banda (BW), aumentar la potencia de transmisión (PS) o disminuir el ruido (PN).

El teorema de Shannon tiene en cuenta la posibilidad de que se utilice un protocolo de control de errores para disminuir la probabilidad de error, de manera que es posible enviar datos por encima de la capacidad bruta del canal, pero sólo si son redundantes. En otras palabras, no se puede superar el límite establecido por el teorema utilizando mecanismos de compresión (reducción de la entropía de los datos) para conseguir enviar información a una velocidad superior a *C*. Y eso porque la capacidad *C* está medida en bits/s después de haber eliminado la redundancia.

## 3.3. Tipos de codificaciones de canal

Hay dos grupos de codificaciones de canal:

- Codificaciones digitales. Generan señales con un número finito de niveles (señales digitales); es decir, tienen transiciones de nivel repentinas separadas por intervalos en los que el nivel se mantiene fijo. Para transmitirlas se necesitan medios de transmisión de paso bajo.
- Modulaciones digitales. Generan señales con un número infinito de niveles (señales analógicas); es decir, varían su valor continuamente. Se pueden transmitir por medios de transmisión de paso bajo y de paso banda.[3]

### *3.3.1. Codificaciones digitales*

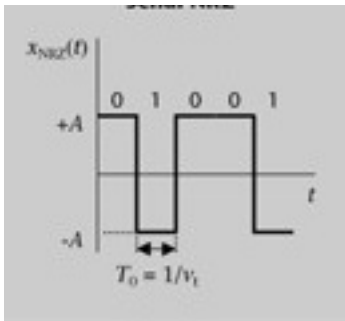
Veremos tres grupos de codificaciones digitales (NRZ, Manchester y bipolar), y compararemos las características siguientes:

- El ancho de banda de la señal, que identifica la velocidad máxima de modulación (y de transmisión) que se puede utilizar en un canal de un ancho de banda determinado.
- La robustez frente al ruido, que indica cuánto afecta a la probabilidad de error la presencia de ruido para señales de la misma potencia.
- La facilidad de sincronización, que informa sobre la facilidad con la que el receptor recupera el sincronismo de bit.

*Codificación NRZ (Non-Return to Zero)*

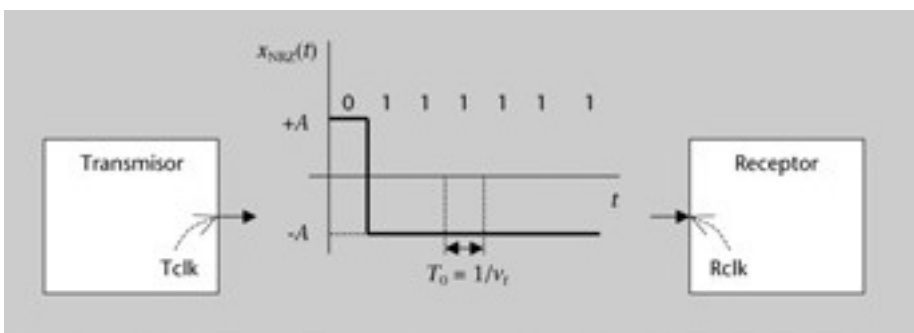
La codificación NRZ utiliza dos niveles: uno con amplitud $+A$ y otro con amplitud $A$, que indican los bits 0 y 1. En el ejemplo de la figura el nivel positivo es el 0 y el negativo es el bit 1. Cada símbolo lleva sólo un bit; por lo tanto, para la codificación NRZ se cumple que $v_t = v_m$.

3. Las versiones analógicas de las modulaciones se usan para la radiodifusión de emisoras de radio y televisión, aunque también hay emisiones de televisión en digital (mediante satélite y cable) y además últimamente han empezado las de radio con el sistema DAB (*Digital Audio Broadcasting)* y de televisión mediante las antenas convencionales con el sistema TDT (Televisión Digital Terrestre).




Señal NRZ

El NRZ es una señal muy sencilla de generar. Lo que ya no es tan sencillo es descodificarla, sobre todo si imaginamos que el transmisor puede enviar una secuencia larga de bits:



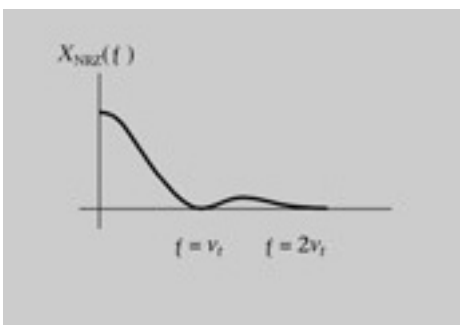

Sincronización del receptor.

La señal digital llega siempre con más o menos distorsión porque ningún canal de transmisión tiene un ancho de banda infinito y, además, ningún transmisor puede generar la señal cuadrada ideal (el NRZ o cualquier otra de las codificaciones digitales que veremos).

El circuito de datos debe tener dos relojes, uno de transmisión ($t_{Tclk} = T_0$) y uno de recepción ($t_{Rclk} = T_0'$). Cada vez que enviamos la secuencia "…01…" o la secuencia "…10…", la señal tiene una transición (el flanco de sincronización) entre los dos bits, y en este momento el reloj de recepción puede recuperar el sincronismo de bit. Sin embargo, cuando tenemos una secuencia larga de ceros o de unos (como la figura anterior), el reloj de recepción no tiene ningún lugar donde recuperar el sincronismo. Puesto que *t*Tclk ≠ *t*Rclk al cabo de un rato el receptor puede "inventar" bits (si *t*Tclk < *t*Rclk) o "perderlos" (si *t*Tclk > *t*Rclk).

Por ello, el NRZ debe utilizarse con algún mecanismo adicional para recuperar el sincronismo[4].

El espectro aproximado del ancho de banda es el siguiente:

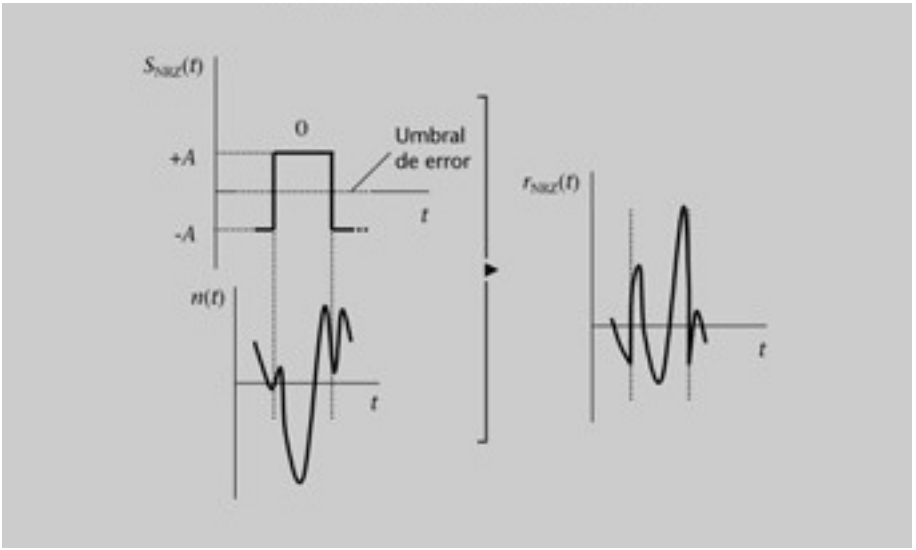



Espectro de una señal NRZ.

Para transmitir una señal digital sin ningún tipo de distorsión, es necesario un ancho de banda infinito. De todos modos, la mayor parte de la potencia está concentrada entre el origen y la frecuencia $f$=0,7·$v_t$. Ésta es la fracción mínima requerida[5] para que un receptor tolere la distorsión producida.

La robustez frente al ruido es relativamente buena. Para que se produzca un error, es necesario que la amplitud del ruido justo en el momento de la descodificación sea mayor que la amplitud de la señal y, en sentido contrario:

4. Línea de reloj, preprocesamiento de la información para evitar series largas de ceros o unos, etc.
5. El valor de 0,7 depende mucho de la calidad técnica del receptor




Umbral de error de una señal NRZ.

Si el receptor descodifica justo en medio del símbolo (bit), descodificará (erróneamente) el bit como un 1, en lugar del 0 original:

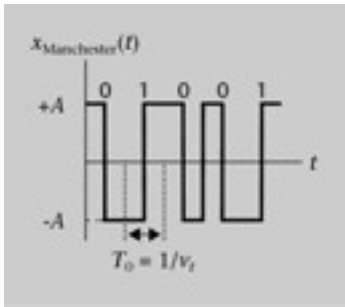
$$r_{NRZ}(t = t_{medio}) < 0,$$

Para que esto suceda, es necesario que la potencia de ruido sea bastante alta (comparable a la potencia de señal), lo que no es nada habitual.

*Codificación Manchester*

La codificación digital Manchester corrige alguno de los problemas de la codificación NRZ a costa de empeorar su ancho de banda (casi necesita el doble).

El problema principal de la codificación NRZ era la dificultad que tenía el receptor para extraer la sincronización de bit. La codificación Manchester impide de manera radical la falta de flancos de sincronización y pone al menos uno en cada bit, independientemente de la secuencia de bits transmitida.




Señal Manchester.

Podríamos decir que cada símbolo lleva "medio bit". En algunos casos diremos que cada bit se codifica en un símbolo Manchester y, en otros, que se codifica en dos. Esto puede confundir, pero lo que ocurre es que el concepto de *símbolo* lo utilizamos con significados ligeramente distintos según el contexto:

a) Si nos referimos al ancho de banda, entenderemos por *símbolo* 'un nivel constante'. Este concepto de símbolo es el utilizado en la fórmula de cálculo de la velocidad de modulación (símbolos/segundo).

b) Si nos referimos a formas de onda disponibles en una codificación, tiene que quedar claro que hablamos de formas dentro de un conjunto de formas disponibles. En este caso los símbolos Manchester no son los niveles, sino las dos combinaciones de dos niveles más el flanco central (observad la figura).

Para construir un bit necesitamos consecutivamente un nivel alto y uno bajo. Ceros y unos se distinguen por el orden en el que se encuentra el símbolo positivo respecto del negativo. De hecho, podríamos decir que la información no va en los niveles, sino en los flancos. Si el flanco es ascendente, lleva un bit (por ejemplo, 0) y si es descendente, el otro (por ejemplo 1).

Esta codificación ofrece al receptor una gran cantidad de flancos donde puede volver a sincronizar el reloj. Simplemente necesita distinguir los flancos centrales al bit de los flancos que pueden aparecer entre bits (entre dos ceros o dos unos).

En sistemas que usan la codificación Manchester, se recupera el sincronismo de bit sin necesidad de ningún mecanismo adicional.

El ancho de banda es peor (mayor) a causa del número mayor de transiciones de la señal cuadrada resultante, como vemos en el espectro de la señal Manchester de la figura siguiente. En cambio, no tiene componente continuo.

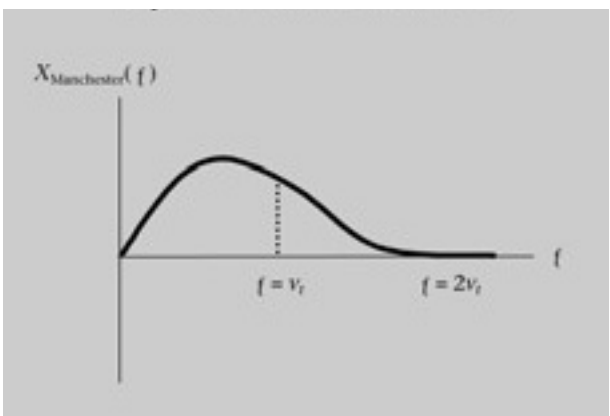
**Componente continuo**

La codificación Manchester no tiene componente continuo porque cada bit aporta una media nula a la señal.

En cambio, en la codificación NRZ cada bit aporta una parte de componente continuo neto. En algunos casos este componente continuo se puede cancelar, como por ejemplo en una secuencia con el mismo número de unos que de ceros. Sin embargo, normalmente esto no sucede.

En algunos canales puede ser deseable que la señal no tenga componente continuo.

Los espectros que hemos dado son para secuencias aleatorias estándar, que no tienen el mismo número de ceros y unos.




Espectro de una señal Manchester.

Para que la señal llegue con una distorsión aceptable, hay que llegar al menos hasta la $f = 1{,}3v_t$ (también son prescindibles las frecuencias bajas, hasta $f = 0{,}1v_t$ aproximadamente).

En cuanto a la robustez frente al ruido, hay que pensar que la información se puede extraer de los flancos de la señal. Si consideramos que el ruido afecta a la señal al sumarse, nos daremos cuenta de que un flanco difícilmente puede ser enmascarado por el ruido.

Manchester, por lo tanto, es muy robusto al ruido, pero es mucho más sensible a la distorsión, que puede afectar contundentemente a los flancos de señal.

*Codificaciones NRZ-I y Manchester diferencial*

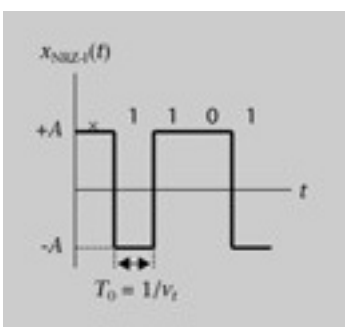
En líneas de transmisión con dos hilos, como el par de hilos o el par trenzado, se puede dar el problema de que en recepción no se conozca el signo de cada uno de los hilos.

El hecho de conectar la línea con una polaridad o la otra puede causar que se reciba la secuencia correcta o la secuencia invertida; es decir, todos los bits cambiados ("…010001…" en lugar de "…101110…").

Una solución posible es utilizar hilos de colores, de manera que en la recepción no haya ninguna confusión posible. También se pueden utilizar codificaciones invulnerables a la inversión de polaridad: las codificaciones diferenciales.

Tanto la codificación Manchester como el NRZ tienen sus versiones diferenciales: la codificación Manchester diferencial y la codificación NRZ-I[6]. En cualquiera de los dos casos no hay una asignación directa símbolo(s)-bit, sino que se codifican los bits en el cambio de señalización: si el símbolo actual es lo mismo que el anterior, tenemos un bit (por ejemplo 0); si es diferente, tenemos el otro bit (por ejemplo 1). Esto hace que tengamos que enviar al menos un símbolo al principio de la transmisión para que el receptor tenga la referencia inicial. Veamos un ejemplo de cada caso:

- NRZ o NRZ-I

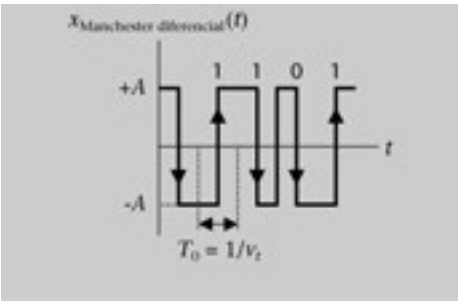



Señal NRZ-I.

6. NRZ-I viene de la expresión inglesa *NRZ-Invert on Ones*.

El primer símbolo positivo no se sabe si es un 1 o un 0, porque no se ve el símbolo anterior. A partir de aquí, los cambios son unos y cuando el símbolo se mantiene, cero.

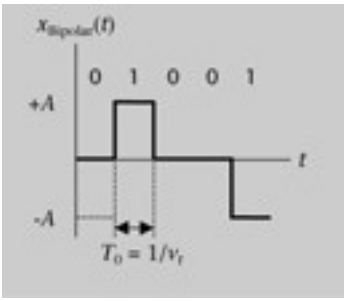
- Codificación Manchester diferencial




Señal Manchester diferencial.

En los dos casos la velocidad de transmisión está afectada de una manera muy ligera, ya que el primer símbolo no lleva información y, además, sólo se añade uno para toda la transmisión.

*Codificaciones bipolar y B8ZS*



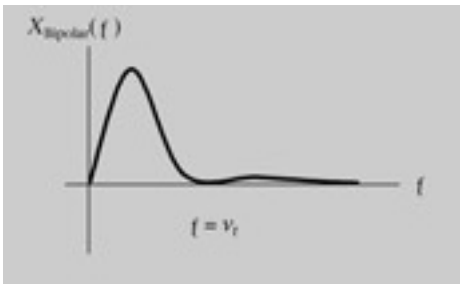

Señal bipolar.

La codificación bipolar mejora las características del NRZ sin alguno de los inconvenientes de la codificación Manchester, pero a costa de perder robustez frente al ruido.

La codificación bipolar utiliza tres símbolos en lugar de dos. Cada símbolo es un nivel constante durante todo el tiempo que dura. Los niveles posibles son+*A*, 0 y *A*. Los bits cero se codifican con una señal nula, y los unos, con una señal de amplitud *A* alternativamente positiva y negativa.

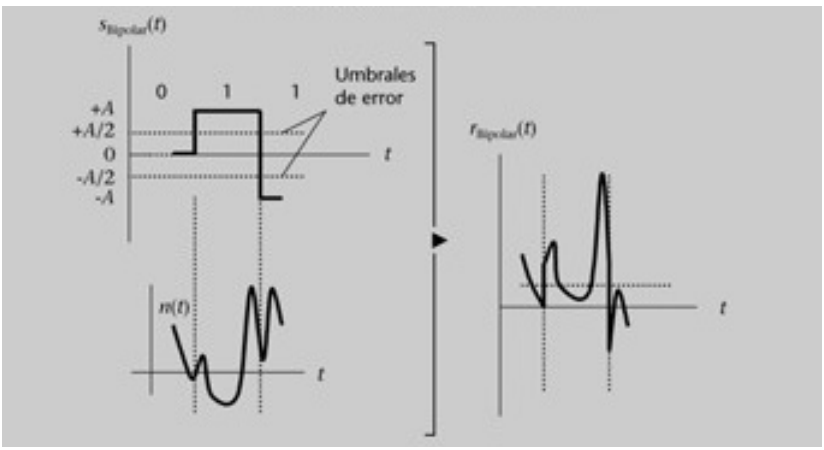
Esta señal es invulnerable a la inversión de polaridad sin tener que usar el mecanismo diferencial.

Desde el punto de vista de ancho de banda, la codificación bipolar es similar al NRZ (tiene un solo nivel por bit), pero sin componente continuo, ya que el componente continuo neto se cancela cada dos unos:



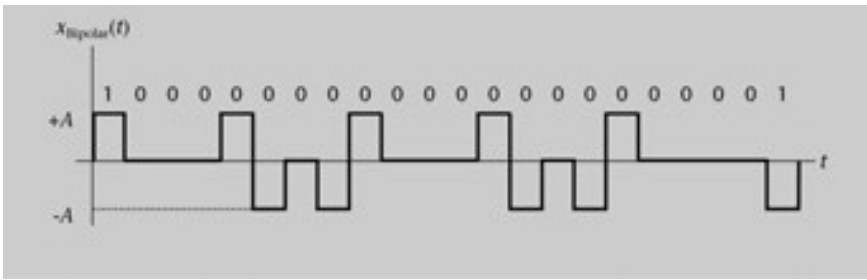

Espectro de una señal bipolar.

Desde el punto de vista de la robustez frente al ruido, esta codificación ya no es tan buena por el hecho de tener tres niveles diferentes. Para que se produzca una equivocación en la descodificación, sólo es necesario que la amplitud de ruido sea aproximadamente A/2:




Umbral de error de una señal bipolar.

La recuperación del sincronismo de bit en codificación bipolar puede ser tan mala como en el NRZ, y por ello no se usa directamente y se utiliza una variante: la codificación B8ZS (*Bipolar with 8 Zeros Substitution*). Esta variante sustituye las cadenas de ocho o más ceros por una codificación con flancos. Las secuencias de siete o menos ceros no varían respecto de cómo serían codificados en bipolar sencilla. El ejemplo siguiente muestra una secuencia con veinte ceros seguidos:




Señal B8ZS con una cadena larga de ceros.

Los ocho ceros seguidos se codifican con los símbolos 0 0 0 $+A$ $-A$ 0 $-A$ $+A$ o bien con los símbolos 0 0 0 $-A$ $+A$ 0 $+A$ $-A$, en función de la polaridad del último 1 anterior a la secuencia de ceros, de manera que no haya confusión posible con la secuencia válida de "00011011". Los cuatro bits últimos del ejemplo se codifican 0 0 0 0 (puesto que no llegan a ocho ceros, la sustitución tiene lugar).

La secuencia de ocho ceros no aporta componente continuo neto.

**Ejemplos de utilización de las codificaciones digitales**

Podemos ver las diferentes aplicaciones de codificación digital en los ejemplos siguientes.

- NRZ: bus de ordenador (sincronismo de bit con línea de reloj), interfaz RS-232 o interfaces serie (sincronismo de bit con preprocesamiento de la información/transmisión asíncrona).
- Manchester: redes de área local Ethernet.
- Manchester diferencial: redes de área local Token Ring.
- Bipolar: RDSI.
- B8ZS: en Europa, en accesos de alta velocidad a RDSI se usa una variante denominada *HDB3* (*High Density Bipolar*-3 *zeros substitution*). No lo hemos comentado aquí porque la HDB3 efectúa un mecanismo de sustitución de cadenas largas de ceros equivalente al de la B8ZS (utilizada en EE.UU.), pero más complejo.

### *3.3.2. Modulaciones digitales*

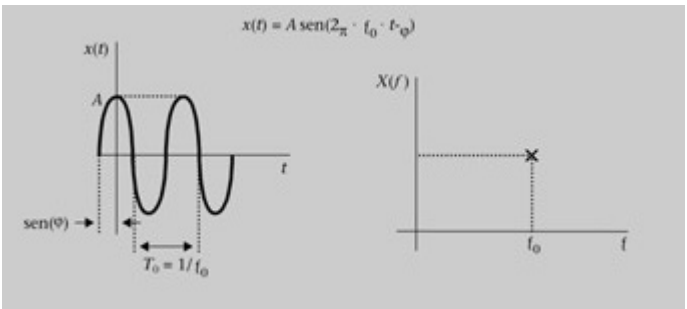
La ventaja principal de las modulaciones digitales respecto de las codificaciones digitales es que tienen el espectro desplazado de la frecuencia cero.

El espectro de las modulaciones digitales se centrará en torno a una frecuencia, la frecuencia portadora, que se puede ajustar (sintonizar) a la posición más adecuada.

Normalmente, la frecuencia portadora nos la dará el medio de transmisión disponible.

*Modulación de la señal*

El mecanismo general para producir una modulación digital se denomina modulación. El proceso de modulación es completamente diferente del proceso de codificación digital y parte de una sinusoidal:




La característica principal de esta señal es que su espectro se encuentra alejado de la frecuencia cero; por lo tanto, cuando introduzcamos los datos dentro de esta sinusoidal sin modificarla mucho, conseguiremos tener una señal alejada de la frecuencia cero, válida para medios de transmisión de paso banda.

Para "poner bits" en una sinusoidal, hay que modificar sus parámetros: la amplitud ($A$), la frecuencia ($f_0$), la fase ($\varphi$) o una combinación de los mismos. Estas cuatro posibilidades configuran otras tantas modulaciones principales:

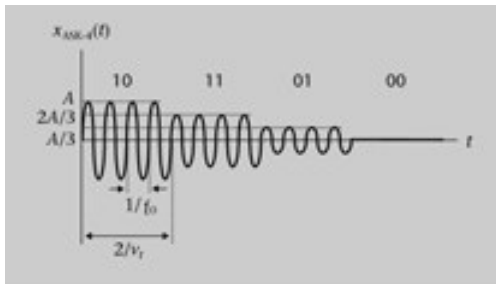
- Modulación de amplitud o *Amplitude Shift Keying* (ASK).

La modulación ASK se obtiene al modificar la amplitud de la sinusoidal según la información que se quiere transmitir.

En general podemos utilizar más de dos amplitudes diferentes. Hablamos de ASK-2, ASK-4, etc. para identificar el número de niveles de la modulación ASK. Haremos lo mismo para el resto de las modulaciones.

**Modulación ASK-4**

Si utilizáramos cuatro amplitudes diferentes, podríamos enviar 2 bits a cada símbolo.



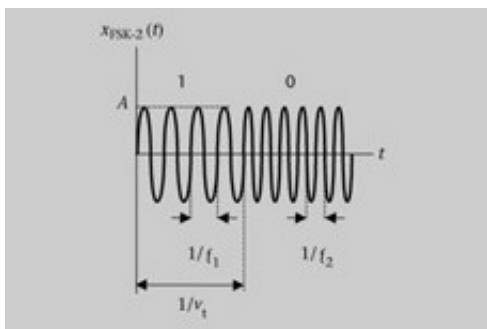

En la figura utilizamos las amplitudes 0, $A/3$, $2A/3$ y $A$ para enviar los símbolos 10, 11, 01 y 00.

- Modulación de frecuencia o *Frequency Shift Keying* (FSK).

La modulación FSK se obtiene al modificar ligeramente la frecuencia portadora.

**Modulación FSK-2**

Con una modulación FSK-2 tendremos una frecuencia $f_1$ $f_0 - 1/2f_\Delta$ y la otra será $f_2 = f_0 + 1/2f_\Delta$. La amplitud se mantendrá constante.

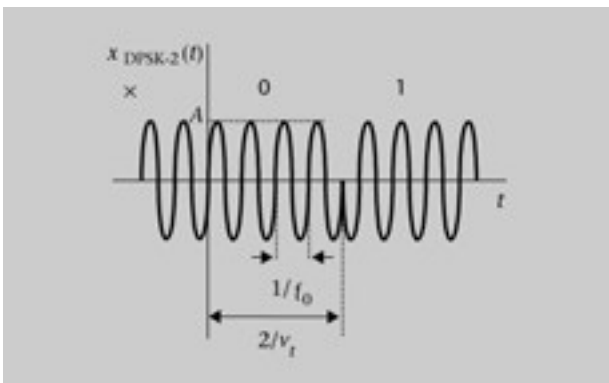


- Modulación de fase diferencial o Differential Phase Shift Keying (DPSK).

La modulación PSK se obtiene al modificar la fase de la señal. Dado que la fase no es una magnitud absoluta, necesitaríamos una fase de referencia (a cero grados). Y puesto que esto no es práctico, codificaremos los bits no en los símbolos, sino en la diferencia entre éstos. Tenemos así la modulación PSK diferencial o modulación DPSK.

**Modulación DPSK-2**

Con la modulación DPSK-2, si enviamos una secuencia de símbolos, todos con la misma fase, enviaremos siempre el mismo bit (por ejemplo, ceros); si la fase cambia 180o, se obtiene una secuencia del otro bit (por ejemplo, unos).

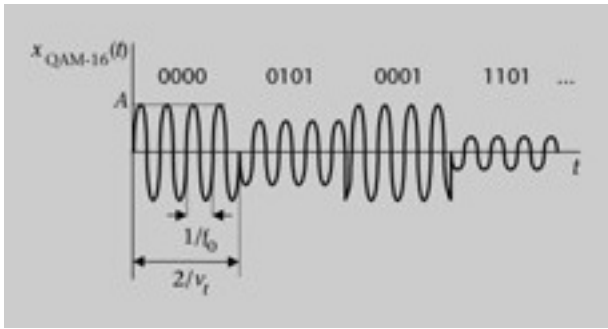



- Modulación de amplitud en cuadratura o *Quadrature-Amplitude Modulation* (QAM).

La modulación QAM se obtiene al modificar la amplitud y la fase de una misma sinusoidal para conseguir transmitir más bits en cada símbolo.

**Modulación QAM**

Podemos utilizar, por ejemplo, cuatro amplitudes diferentes y cuatro fases distintas, para conseguir $4 \cdot 4 = 16$ símbolos diferentes (QAM-16). En la figura vemos cuatro de los dieciséis símbolos que podrían resultar.




Para resumir:

El ancho de banda de las diferentes modulaciones es similar. Para una misma velocidad de modulación tenemos anchos de banda similares a una codificación digital (aproximadamente el doble).

El espectro de la modulación queda centrado de manera simétrica en torno a la frecuencia portadora.

Las modulaciones QAM presentan más robustez frente al ruido porque utilizan una combinación de ASK y de DPSK.

**Uso de QAM en los módems**

Los módems actuales transmiten utilizando modulaciones derivadas de la QAM. Se consiguen velocidades de transmisión de hasta 33,6 kbps con los módems más rápidos (norma V.34), y se utilizan velocidades de modulación en torno a los 3.000 baudios.

## 3.4. Cálculo de la probabilidad de error

Como hemos dicho al principio del capítulo, la secuencia de bits en la salida del descodificador de canal del receptor no coincidirá, por regla general, con la secuencia en la entrada del codificador de canal del transmisor, por culpa de los errores de transmisión. Dado que tratamos secuencias no fijas de bits, se habla de probabilidad de error, en términos estadísticos, para cuantificar este efecto.

**Probabilidades de error típicas**

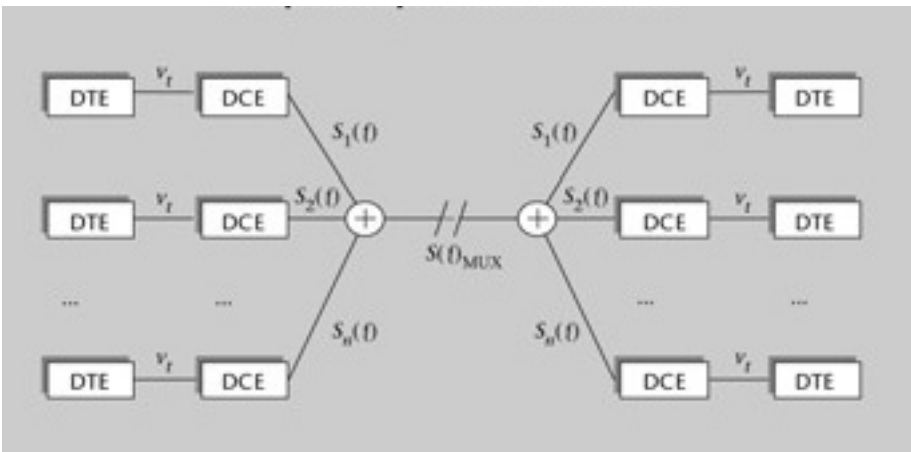
Valores típicos para la probabilidad de error en sistemas convencionales pueden ir desde $10^{-4}$, hasta los $10^{-9}$ de las fibras ópticas; es decir, entre un bit erróneo de cada 10.000 transmitidos hasta un bit erróneo de cada 1.000.000.000.

Al enunciar el teorema de Shannon ya hemos introducido el concepto de relación señal/ruido como el principal culpable de que se produzcan estos errores de transmisión. Ahora, al definir las codificaciones y las modulaciones digitales, hemos visto cómo se introducen nuevas variables como la robustez ante el ruido de cada codificación o modulación o el número de niveles distintos que puedan utilizar y, por lo tanto, la distancia entre estos niveles.

Dar una expresión cerrada para esta probabilidad de error no es obvio. La carga matemática es fuerte y las suposiciones que hay que hacer no están al alcance de este capítulo. Para hacernos una idea podemos dar esta expresión:

$$P\varepsilon = Q\left(\frac{S}{N} \cdot \frac{BW}{v_t}\right)$$

La probabilidad de error en el bit ($p_e$) depende de la relación señal/ruido, del ancho de banda del canal y de la velocidad de transmisión, todo mediante la función Q, que es una integral.




Probabilidad de error de las codificaciones NRZ y bipolar.

Nosotros no entraremos en la demostración matemática. Simplemente nos interesa saber que la probabilidad de error depende de la relación señal-ruido, del ancho de banda y de la velocidad de transmisión.

Para cada una de las codificaciones hay una función que devuelve la probabilidad de error en función de estos parámetros. Por ejemplo, para una transmisión NRZ y una bipolar, las probabilidades de error en función de $S/N \cdot BW/v_t$ serán las que se ven en la figura de la página anterior.

## 3.5. Multiplexación

La multiplexación es el mecanismo que permite compartir un medio de transmisión único en circuitos de datos diferentes.

Hay dos técnicas fundamentales para conseguirlo: la multiplexación por división en frecuencia (FDM) y la multiplexación por división en tiempo (TDM). Empezaremos por ver la multiplexación por división en frecuencia, que fue la primera en ser usada, y después veremos la multiplexación por división en tiempo, que es la que se utiliza más en la actualidad.

### *3.5.1. Multiplexación por división en frecuencia (FDM)*

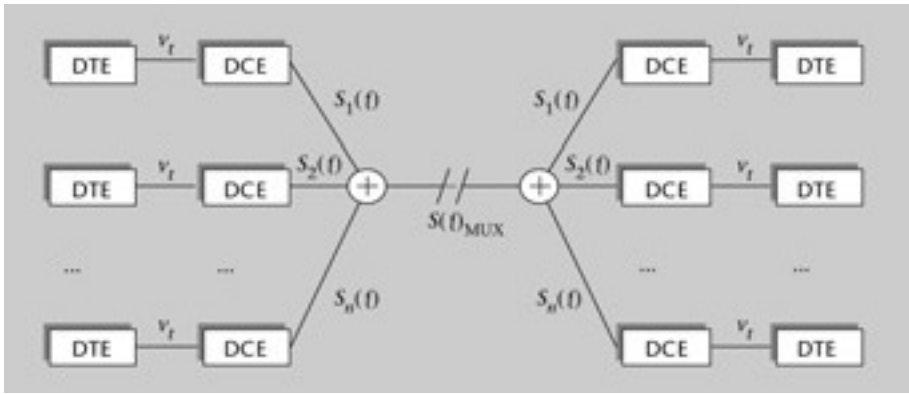
Si tenemos un medio de transmisión con un ancho de banda grande, o incluso prácticamente infinito, como el aire, es posible que la señal que generamos no lo ocupe todo.

Mediante técnicas como la modulación digital, podemos llenar los márgenes libres del espectro con modulaciones a otras frecuencias.

Las bandas de los canales tienen que encajar dentro de la banda de paso del medio y se tiene que cumplir la relación siguiente:

$$BW_{medio} > \sum_{i=1}^{n} BW_i$$

En la figura siguiente podemos ver el esquema de un medio de transmisión caracterizado por la función de transferencia $H(f)$, que puede transportar $n$ canales ($S_1(f)$, …, $S_n(f)$), centrados en las portadoras $f_{p1}$, …, $f_{pn}$:




Multiplexación por división en frecuencia.

Entre canal y canal se tienen que respetar unos espacios vacíos denominados bandas de guarda, para que no haya interferencias entre los mismos. La magnitud de estas bandas de guarda depende de la eficacia de los codificadores y descodificadores que se utilicen para discriminar espectros adyacentes.

En la actualidad, la multiplexación FDM se usa fundamentalmente en transmisiones radioeléctricas, porque permite tratar las señales de canales diferentes de manera independiente, de forma contraria a lo que pasa con la multiplexación TDM. Es decir, no es necesario que los equipos de transmisión y recepción colaboren especialmente entre sí; sólo hace falta que utilicen bandas separadas y que respeten las bandas de guarda. Otra ventaja de la multiplexación FDM es el ahorro de cableado.

Antes de que se utilizara la multiplexación en telefonía, cada par de hilos llevaba una única llamada. En 1888, en Nueva York, la densidad de los cables que viajaban por las calles hizo que la nieve se acumulara sobre los mismos y causara unos destrozos que dejaron a la ciudad sin servicio telefónico.

**Multiplexación FDM de señales analógicas**

El concepto de *multiplexación FDM* es aplicable a la transmisión de modulaciones tanto digitales –ASK, DPSK, etc.– como analógicas –amplitud modulada (AM), frecuencia modulada (FM).

Las transmisiones de las diferentes emisoras de radio y de los distintos canales de televisión están multiplexadas entre sí. La acción de "sintonizar" una emisora es equivalente a decir que "elegimos" una frecuencia portadora determinada.

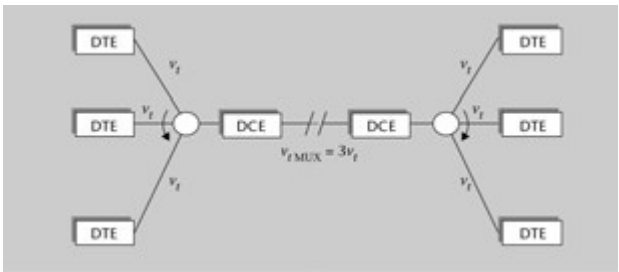
### *3.5.2. Multiplexación por división en tiempo (TDM)*

Al contrario de lo que pasaba con la multiplexación FDM, en la multiplexación TDM es necesario que todos los transmisores y receptores se pongan de acuerdo antes de transmitir la señal y después de recibirla. El multiplexor TDM se encarga de tomar los bits generados por todas las fuentes y los agrupa (los multiplexa) en una sola. Esta nueva fuente de bits funcionará a la velocidad de transmisión siguiente:

$$v_t|_{\text{MUX}} = \Sigma v_t|_{\text{cada canal}}$$

La capacidad del canal tendrá que ser igual o superior a la velocidad de transmisión resultante.

En recepción el proceso será el inverso: se entregarán los bits a cada uno de los DTE correspondientes.




Multiplexación por división en tiempo.

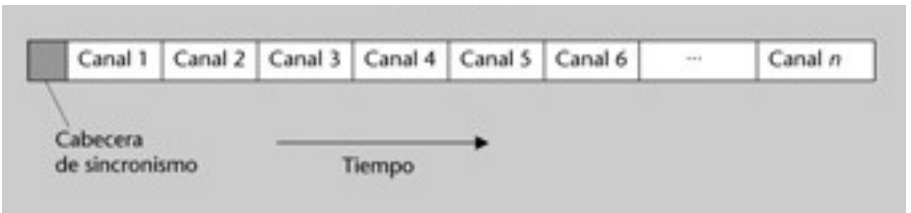
Ahora bien, una vez multiplexado el flujo de bits dentro del circuito de datos, se necesita algún elemento dentro de la señal o dentro de los bits que identifique de manera inequívoca la señal o los bits de datos para que el desmultiplexor pueda saber a qué DTE corresponde cada uno de los bits individuales del flujo. Hay dos maneras de conseguirlo:

1) Señal específica de sincronismo. Señal o símbolo diferente de los símbolos utilizados para codificar bits. Cuando el receptor la recibe, sabe que a continuación viene un bit (o diferentes bits) del primer DTE, después uno (o diferentes) del segundo, etc.
2) Secuencia específica de bits. Secuencia de bits que indica en el desmultiplexor que se inicia una secuencia de bits nueva para el primer DTE, y después para el segundo, etc.

El grupo formado por los bits (o señales) que corresponden a la sincronización y a los bits de cada uno de los DTE lo denominaremos trama.

En cualquiera de los dos casos, la sincronización (sincronismo de trama) normalmente se sitúa sólo antes de los bits del primer DTE; el resto de los bits se asigna a los DTE correspondientes contándolos.

Aunque las tramas TDM son la forma más simple de trama, contienen los elementos básicos de su estructura, que se esquematiza en la figura siguiente:




Trama TDM.

### *3.5.3. Multiplexación por división en tiempo estadística (STDM)*

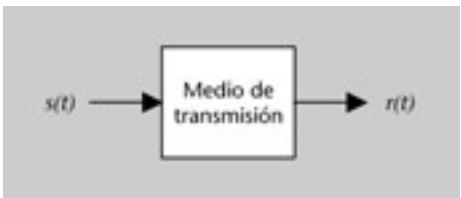
En transmisiones de datos, las multiplexaciones FDM y TDM tienen un problema grave: no están adaptadas a las características del tráfico generado por los ordenadores y otros equipos informáticos: el tráfico informático es casi siempre de corta duración. En caso de utilizar la FDM o la TDM, las tramas irán casi vacías la mayor parte del tiempo. Si un DTE decide hacer uso del canal, se encontrará sólo con su fracción de la capacidad, cuando el resto del DTE no utiliza su fracción.

La multiplexación por división en tiempo estadística (STDM - *Statistical Time Division Multiplexing*) resuelve el problema de manera efectiva: las tramas contienen datos de un solo canal.

Si hay un solo DTE que necesita el circuito de datos, lo tendrá completamente disponible. Si hay dos, se lo repartirán, una mitad para cada uno, etc. Para que esto pueda llevarse a la práctica, es necesario que el desmultiplexor distinga los diferentes DTE origen. Cada una de las nuevas tramas tiene que ir identificada. A este identificador lo denominaremos *canal* o *dirección*, e irá normalmente situado dentro de la cabecera de la trama, después de las señales o bits de sincronismo.

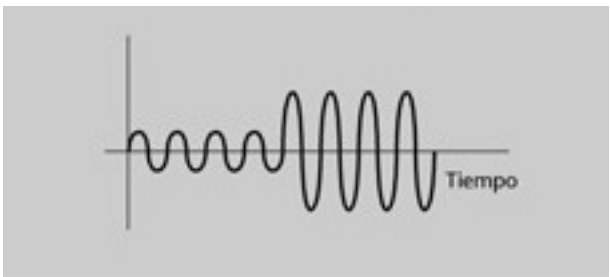
## 4. Medio de transmisión

Siguiendo el modelo sencillo que hemos propuesto, el medio de transmisión se compone, en principio, sólo de un elemento, el medio mismo, al cual inyectamos una señal $s(t)$ y, como salida, recibimos una señal $r(t)$.




Sería ideal tener un medio de transmisión que cumpliera que $r(t) = s(t)$; es decir, que la señal recibida fuera la misma que la enviada. Esto, sin embargo, es imposible, a causa de múltiples factores perturbadores de la señal que a continuación describiremos.

Cuando escribimos $r(t)$ o $s(t)$, nos estamos refiriendo a la evolución de una magnitud física, normalmente la tensión eléctrica, relacionada con la señal, en función del tiempo:




Evolución temporal de una señal.

Esta magnitud física, normalmente, es la tensión eléctrica que en el sistema internacional se mide en voltios (V), y puede tomar valores positivos o negativos.

También se suele representar una señal eléctrica mediante la potencia, que es proporcional a la tensión elevada al cuadrado. En el sistema internacional, la unidad de medida de potencia es el vatio (W).

## 4.1. Perturbaciones de la señal

Las perturbaciones que afectan a la señal son básicamente tres:

- Atenuación
- Distorsión
- Ruido

La atenuación y la distorsión son inherentes a la señal, mientras que el ruido es una perturbación independiente. De hecho, las perturbaciones independientes de la señal son diversas, pero las agrupamos con la denominación genérica de ruido.

Si bien la atenuación es un caso particular de distorsión, la estudiamos por separado porque la manera de luchar contra ambas es diferente.

### *4.1.1. Atenuación*

La atenuación es la disminución de la potencia de la señal a medida que se propaga por el medio. La forma de la señal se mantiene.

Si una señal emitida, $s(t)$, sólo se viera afectada por la atenuación, la señal recibida, $r(t)$, sería fruto de la ecuación:

$$r(t) = s(t)/a$$

$a$ es el valor de la atenuación.

Por ejemplo, si tenemos una atenuación de 2, la señal recibida sería de la mitad, y su evolución temporal sería como la que se ve en la figura siguiente:




Atenuación de una señal.

En el caso de que quisiéramos expresar la potencia media recibida, y no la evolución de la tensión de la señal recibida, tendríamos que utilizar la ecuación siguiente:

$$P_R = P_S/a^2$$

Por el medio de transmisión de la imagen anterior, con una atenuación de 2, la potencia media recibida sería una cuarta parte de la potencia original:

$$P_R = P_S/2^2 = P_S/4$$

Utilizaremos el convenio de escribir las potencias con mayúsculas y las tensiones con minúsculas: $P_R$ es la potencia media asociada a la señal r(t).
En este sentido, utilizaremos *a* para la atenuación de la tensión y *A* para la atenuación de potencia.

La atenuación de tensión y la atenuación de potencia están relacionadas de la misma manera que la tensión y la potencia:

$$A = a^2$$

Los fenómenos físicos que provocan la atenuación en un medio de transmisión son los siguientes:

- Resistividad del conductor. En medios de transmisión conductores (por ejemplo, el par de hilos de cobre) se produce una pérdida de tensión porque el material conductor tiene una cierta resistencia. Ésta se manifiesta por un calentamiento del material que comporta una pérdida de energía, la cual será mayor cuanto mayor sea la longitud de la línea.
- Fenómenos de radiación. Bajo ciertas condiciones, los medios de transmisión se comportan de manera análoga a una antena: parte de la potencia de la señal transmitida se convierte en radiación que se escapa de la misma línea y esta pérdida todavía es más importante que la debida a la resistividad del conductor.[7]

Sea cual sea la causa de la atenuación, el cálculo numérico de su efecto no variará lo que acabamos de presentar.

7. Más adelante, veremos que esta radiación que se escapa se puede convertir en un ruido de interferencia.

Tanto la resistividad como la radiación dependen fuertemente de la longitud del medio, o sea, de la distancia entre origen y destino. Por lo tanto, es muy importante caracterizar el efecto de la distancia en las ecuaciones anteriores.

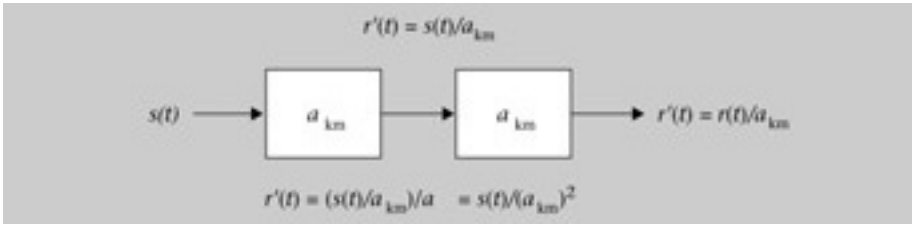
Supongamos que $a_{km}$ es la atenuación de un medio de transmisión de 1 km de longitud:




Atenuación por kilómetro.

Para conocer la atenuación de este mismo medio de transmisión si abarca el doble de distancia, tenemos que aplicar la ecuación anterior dos veces:

- $r'(t) = r(t)/a_{km}$.
- $r(t) = s(t)/a_{km}$.




Progresión geométrica de la atenuación.

Si generalizamos la ecuación anterior, obtenemos el resultado siguiente:

$$r(t) = s(t)/(a_{km})^d$$

Para saber la potencia recibida, podemos utilizar la ecuación equivalente siguiente:

$$P_R = \frac{P_S}{A_{km}{}^d}$$

$d$ es la distancia, expresada en kilómetros. $A_{\text{km}}$ es la atenuación de potencia, por kilómetro.

De esta forma, podemos definir la atenuación total de un medio de transmisión de longitud $d$ a partir de la atenuación por kilómetro, como:

$$A = (A_{\text{km}})^d$$

*El decibelio*

Utilizar potencias reales o hacer raíces con índices reales hoy no plantea ningún problema. Debemos pensar, sin embargo, que las líneas de transmisión se utilizan desde el siglo XIX, y entonces no había calculadoras. Por este motivo, tradicionalmente, a las ecuaciones que involucran potencias y atenuaciones se les ha aplicado logaritmos en base 10. De este modo, la relación entre la potencia recibida y la transmitida queda de la manera siguiente:

$$\begin{aligned} \log_{10} P_R &= \log_{10}(P_S/(A_{\text{km}})^d) =, \\ &= \log_{10} P_S - \log_{10}(A_{\text{km}})^d =, \\ &= \log_{10} P_S - d \cdot \log_{10} A_{\text{km}}. \end{aligned}$$

Por razones puramente estéticas, los valores anteriores se multiplican por 10:

$$10 \log_{10} P_R = 10 \log_{10} P_S = d \cdot 10 \log_{10} A_{\text{km}}.$$

Con esto, se simplifica la ecuación: se pasa de una división y una potencia a una resta y una multiplicación. Habitualmente, pues, la atenuación no viene expresada en formato lineal, sino logarítmico.

Cuando un valor $x$ (en este caso, la atenuación) se expresa como $10 \log_{10} x$, diremos que el valor $x$ *se* expresa en decibelios o en dB.

Según esta definición, la ecuación anterior queda de la manera siguiente:

$$10 \log_{10} \frac{P_S}{P_R} = d_{\text{dB/km}}$$

La atenuación total, expresada en decibelios, queda de esta forma:

$$A_{dB} = d \cdot A_{dB/km}$$

**Ejemplo de cálculo de la potencia recibida**

Supongamos que tenemos una fibra óptica con una atenuación de 0,2 dB/km. ¿Qué atenuación tendrá una línea de 15 km hecha con esta fibra?
Pasemos a dB la ecuación de la atenuación total:

$$A_{dB} = d \cdot A_{dB/km}$$

Si aplicamos los datos del enunciado, obtenemos la cantidad siguiente:

$$A_{dB} = 15 \cdot 0{,}2 \text{ dB} = 3 \text{ dB}.$$

Por ejemplo, si transmitiéramos 1 W de potencia, lo que recibiríamos al cabo de 15 km sería:

$$10 \log P_S/P_R = 3 \text{ dB}$$
$$P_S/ P_R = 10^{3/10} = 10^{0,3} = 1{,}995 = 2$$
$$P_S = 2P_R$$
$$P_R = P_S/2 = 0{,}5 \text{ W}.$$

Aunque pueda parecer que hoy día no tiene demasiado sentido utilizar el dB porque las calculadoras ya nos hacen los cálculos complicados, lo cierto es que sí que se utiliza, en parte por tradición, y también porque permite hacer cálculos mentales ágiles. Veámoslo con un ejemplo. Consideremos la tabla siguiente, que contiene las conversiones de números clave:

| Valor lineal | Valor en dcibelios |
|---|---|
| 1 | 0 |
| 2 | 3 |
| 10 | 10 |
| 0,1 | -10 |
| $10^n$ | 10n |

Conversión a decibelios.

Es muy sencillo calcular otros valores a partir de esta tabla. Por ejemplo, si tenemos una atenuación de 36 dB, en lugar de hacer el cálculo:

$$A = 10^{36/10} = 4.000,$$

podríamos ver que:

$$A_{dB} = 36 \text{ dB} = 10 \text{ dB} + 10 \text{ dB} + 10 \text{ dB} + 3 \text{ dB} + 3 \text{ dB},$$

y, por lo tanto, siguiendo la tabla:

$$A = 10 \cdot 10 \cdot 10 \cdot 2 \cdot 2 = 4.000.$$

*Los amplificadores*

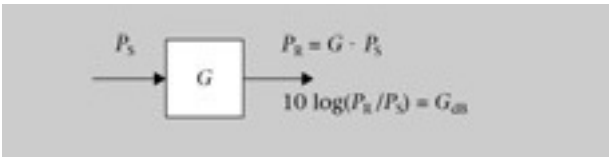
En ciertas situaciones puede pasar que la señal recibida sea demasiado débil para que el receptor la pueda descodificar. Esto limita el alcance de los diferentes medios de transmisión.

Consideremos la situación siguiente:

En una línea, unos receptores necesitan recibir la señal de, como mínimo, 1 mW para poder descodificarla, y el transmisor emite con 1 W; la atenuación máxima admisible será, por lo tanto, de 30 dB. Si el medio de transmisión elegido para esta línea tiene una atenuación $A_{\text{dB/km}} = 2$ dB, emisor y receptor no podrán situarse a más de 15 km de distancia. Si se quiere llegar a 100 km con este medio, será necesario utilizar amplificadores.

Un amplificador es un equipo electrónico capaz de aumentar la potencia de la señal de entrada según un factor de amplificación o ganancia, *G*.

Gráficamente, un amplificador se representa según el esquema siguiente:




Ganancia de potencia de un amplificador.

Habitualmente, la ganancia $G$ se mide en decibelios, como la atenuación. Desde el punto de vista de cálculo, podríamos decir que un amplificador con una ganancia de $G$ es equivalente a un medio de transmisión con una atenuación ("negativa") de $A_{dB} = -G_{dB}$ . En valor lineal, esto sería equivalente a $A = 1/G$.

Reanudando el ejemplo anterior, si se quisiera abarcar los 100 km con el medio de transmisión descrito, la potencia de recepción sería 200 dB inferior a la de origen, es decir, $P_R = 10^{-20} \cdot P_S = 10^{-20} \cdot 1\ \text{W} = 10^{-20}\ \text{W}$. Esta potencia es prácticamente inmedible. Además, los receptores sólo pueden descodificar señales de, como mínimo, 1 mW. Por lo tanto, necesitamos un amplificador de:

$$G = 1\ \text{mW} / 10^{-20}\ \text{W} = 10^{17}\ (\text{o } 170\ \text{dB}).$$

Si hubiéramos hecho el cálculo con decibelios, habría sido más rápido: si la caída es de 200 dB, y permitimos sólo una caída de 30 dB (de 1 W a 1 mW), necesitamos un amplificador de 170 dB.

Matemáticamente, todo es correcto; en la práctica, sin embargo, irrealizable: un amplificador llega a los 30 o 40 dB como máximo.

La razón por la cual un amplificador no puede alcanzar niveles de amplificación mayores es la presencia del ruido eléctrico[8]. El amplificador mismo genera ruido. A partir de ciertos niveles de amplificación, lo único que conseguiremos amplificar es el ruido generado por el mismo amplificador. Por ahora no entraremos en más detalles, pero el ruido eléctrico es un punto lo bastante importante como para que le dediquemos un subapartado más adelante.

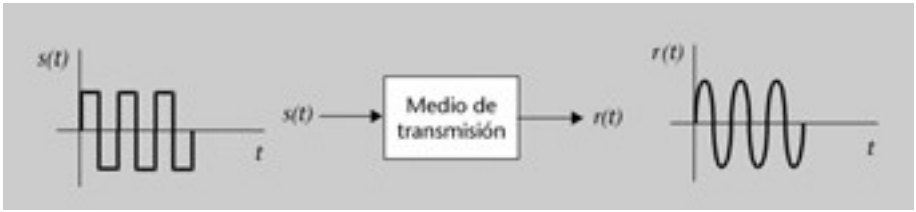
Para alcanzar los 100 km en las condiciones mencionadas en la práctica, será necesario trocear el medio de transmisión en medios más cortos y encadenarlos con amplificadores: antes de que la potencia baje por debajo de cierto umbral, se amplifica.

### *4.1.2. Distorsión*

Decimos que un medio de transmisión distorsiona una señal si la señal recibida no tiene la misma forma que la original. La distorsión no se puede corregir simplemente con un amplificador, como pasaba con la atenuación.

8. El concepto de *ruido eléctrico* es análogo al concepto de ruido "sonoro".

Veamos un ejemplo gráfico que nos ayudará a comprender el concepto de distorsión:



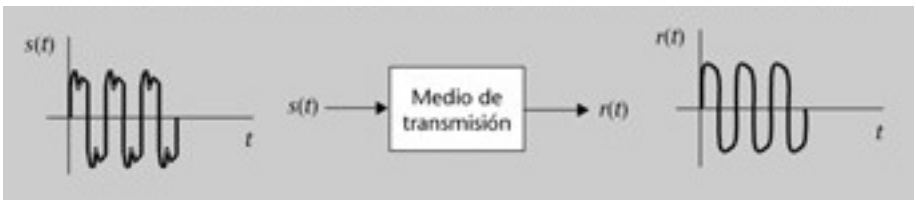

Ejemplo de distorsión. La señal recibida, r(t), es claramente distinta de la original, s(t); se ha distorsionado.

Habitualmente, la distorsión provoca un "redondeo" de la señal que será más o menos importante según la velocidad de variación de la señal (es decir, la velocidad a la que la señal varía), del tipo de medio de transmisión y también de su longitud.

Si la señal enviada contiene detalles de duración corta, la distorsión provocada por el medio de transmisión la puede hacer desaparecer totalmente, como se ve en la figura siguiente:

Hablaremos de *distorsión inaceptable* si estos detalles son importantes para el receptor y de *distorsión tolerable,* si no lo son.

El fenómeno de distorsión puede hacer que un medio de transmisión determinado sea totalmente inadecuado para una señal concreta y, en cambio, perfectamente válido para otra.




Efecto de "desenfoque" de la distorsión: pérdida de los detalles de la señal.

**Ejemplos de distorsión en señales sonoras**

Podemos encontrar muchos ejemplos de distorsión en el sonido:

a) El teléfono o la radio provocan una tonalidad especial en el sonido original de la voz. Somos perfectamente capaces de distinguir que la voz de una persona determinada nos llega a través de teléfono, la radio o nos habla de tú a tú a un metro de distancia. Por lo tanto, percibimos claramente la distorsión introducida en la señal transmitida por cada uno de los tres medios, ya que los oímos de forma distinta. Aun así, somos capaces de entender qué nos dicen e incluso reconocer al interlocutor en cualquiera de los tres casos. Convendremos, pues, que la distorsión de los tres medios de transmisión sonora –aire, teléfono y radio– es tolerable para mantener una conversación.
b) La reverberación, que es un caso particular de eco, es provocada por la manera particular en la que las ondas sonoras rebotan por paredes, tierra y elementos próximos al frente de onda sonora. Algunas iglesias, por ejemplo, provocan una reverberación (distorsión) tal que desde según qué lugares no podremos entender el sermón; evidentemente, en este caso la distorsión no es tolerable.

El análisis de la distorsión es mucho más cómodo desde el punto de vista del dominio de la frecuencia:

Denominamos *distorsión* al efecto que provoca sobre la señal atravesar un medio de transmisión con una función de transferencia que no toma el mismo valor para todas las frecuencias.

Nos podemos encontrar con casos en los que el medio afecte poco a la señal y, por lo tanto, la distorsión sea poco apreciable. En cambio, puede ocurrir que la señal y el medio no "encajen" nada bien y la distorsión sea severa. Se dice entonces que la señal y el medio no están adaptados.

Otro detalle importante es que la mayoría de los medios responde muy mal a frecuencias altas.

De la misma manera que para reducir la atenuación se sitúa un amplificador (o diferentes) en la salida del transmisor, en la entrada del receptor, o en el medio, para reducir la distorsión se utiliza un equipo activo equivalente, el ecualizador.

Si denotamos por *H*ec(*f*) la función de transferencia del ecualizador, entonces lo que se tendría que conseguir para anular el efecto del medio es lo siguiente:

$$R(f) = H_{ec}(f) \cdot H(f) \cdot S(f) = S(f).$$

Por lo tanto:

$$H_{ec}(f) \cdot H(f) = 1 \Rightarrow Hec(f) = H(f)^{-1}.$$

Si se consigue que el ecualizador quede ajustado convenientemente, es decir, que sea exactamente el inverso de la función de transferencia del medio, el receptor recibirá la señal sin distorsión:




Resultado de la ecualización.

Dado que las funciones de transferencia multiplican la señal y, por la propiedad conmutativa de la multiplicación, el ecualizador se puede situar al principio, al final o en la mitad del medio de transmisión, pasa lo mismo que con la atenuación y los amplificadores. De hecho, si reflexionamos un momento, veremos que un amplificador es tan sólo un caso particular de ecualizador, y la atenuación es simplemente una distorsión igual para todos los armónicos.

**Ejemplo de ecualizador**

Una atenuación de 3 dB ($A_{dB}$ = 3 dB o $A$ = 2) necesitaría un amplificador de 3 dB ($A$dB = 3 dB o $A$ = 2), que expresado en términos de funciones de transferencia nos daría dos funciones constantes: $H(f)$ = 0,5 y$H_{ec}(f)$ = 2.

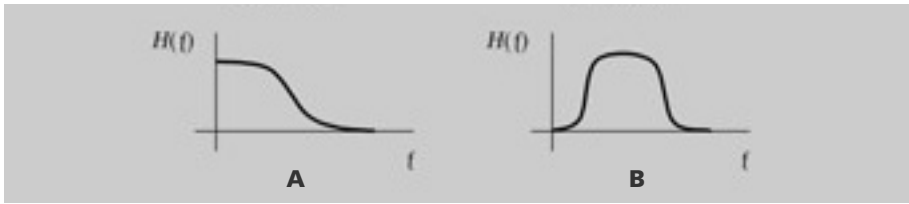
*Canal ideal*

Hablamos de canal de transmisión cuando se dispone de un medio de transmisión combinado con diferentes equipos activos (amplificadores, ecualizadores, etc.) para mejorar las características de transmisión de la señal.

Si se llega a conseguir que se verifique la expresión $H_{eq}(f) = H(f)^{-1}$, se ha obtenido lo que se denomina un canal ideal para la transmisión.

Desgraciadamente, no siempre será posible conseguir un canal ideal, porque para determinados valores de $f$ la función de transferencia puede ser muy próxima a 0 y, por lo tanto, la amplificación que tendría que hacer $H_{ec}(f)$ a estas frecuencias debería ser enorme.

En la figura siguiente podemos ver los dos comportamientos típicos de los medios de transmisión:

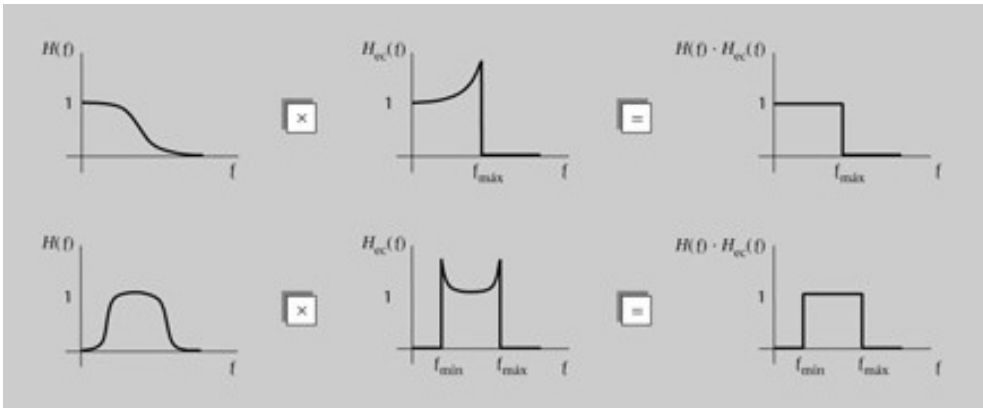



A: Función de transferencia paso bajo. B: Función de tranferencia paso banda.

- Medios de transmisión paso bajo. Medios caracterizados por una función de transferencia que permite el paso de las frecuencias bajas y que atenúa fuertemente las altas.
- Medios de transmisión paso banda. Medios caracterizados por una función de transferencia que permite el paso de ciertas frecuencias, pero no el de las frecuencias bajas ni de las altas.

Con estos dos comportamientos, está claro que no es nada sencillo llegar a conseguir un canal ideal.

Una manera de conseguir un "canal ideal" es considerar un margen útil de frecuencias e intentar alcanzar el comportamiento de canal ideal sólo dentro de este margen. De este modo, los medios modelo de la figura anterior quedarían, gráficamente, de la manera siguiente:

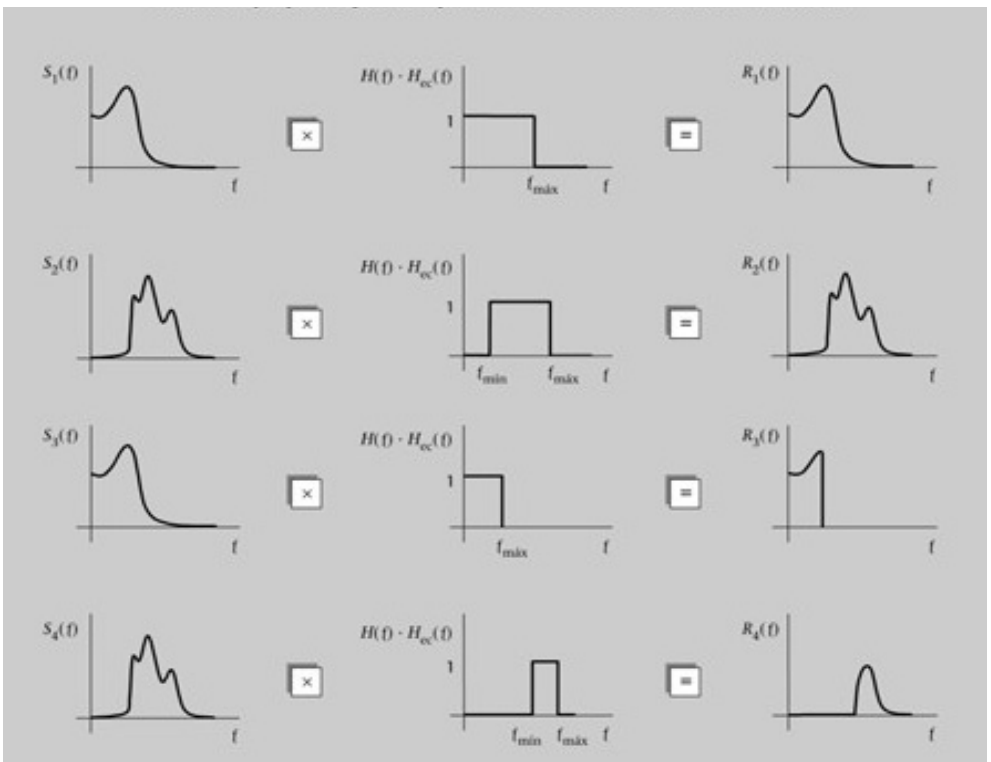



Resultado de la ecualización.

Se puede considerar como canal ideal cualquiera de los dos medios ecualizados de la figura, porque no producen distorsión mientras las señales estén confinadas dentro del margen de frecuencias. De este modo, para que un canal ideal se comporte como ideal, es necesario que las bandas, tanto de la señal como del canal, estén convenientemente ajustadas.

**Ejemplo de señales que experimentan distorsión**

En la figura siguiente, las señales $S_1(f)$ y $S_2(f)$ no experimentan distorsión y, en cambio, $S_3(f)$ y $S_4(f)$ sí que la experimentan, ya que las señales no encajan correctamente dentro de la ventana de transmisión ofrecida por el medio.




Diferentes ejemplos de presencia y ausencia de distorsión en canales ecualizados.

Normalmente, hablaremos de ventana de transmisión cuando la forma de la función de transferencia del canal de transmisión no sea relevante, y sólo sean relevantes las frecuencias de paso mínima y máxima del medio, en relación con

las frecuencias mínima y máxima de la señal que puede pasar. Evidentemente, esto es lo que ocurre cuando hablamos de canales ideales.

Las características de transmisión, como ya se ha visto, están definitivamente marcadas por la distancia que hay entre la frecuencia máxima y la mínima, tanto de la señal como del medio de transmisión. Esta distancia la denominaremos *ancho de banda* (*BW*):

$$BW = f_{máx} - f_{min}.$$

El concepto *ancho de banda* es aplicable tanto a funciones de transferencia (ya sea de un medio de transmisión, de un ecualizador o incluso de un amplificador) como a los espectros de la señal.

Para que una señal pueda pasar sin distorsión por un canal, es condición necesaria que el canal tenga un ancho de banda superior al de la señal. La condición será suficiente si la banda de paso de la señal está incluida totalmente dentro de la del canal y, además, el canal se comporta como canal ideal (al menos dentro de la banda de la señal).

### *4.1.3. Ruido*

Para modelar la señal recibida, todo lo que hemos visto (atenuación y distorsión) se puede resumir en la ya conocida ecuación siguiente:

$$R(f) = H(f) \cdot S(f).$$

La expresión anterior equivale a la siguiente:

$$r(t) = h(t) * s(t).$$

Como ya hemos comentado al inicio de este subapartado, hay perturbaciones de la señal que no dependen de la señal y son lo que denominamos *ruido*.

Según las ecuaciones que acabamos de presentar, si la señal es nula, no obtendríamos nada. Sin embargo, esto no es verdad. Intentemos sintonizar con un aparato de radio un punto donde no se oiga ninguna emisora y aumentemos su volumen: oiremos ruido donde tendría que haber silencio:

para $s(t)= 0$ tenemos que $r(t)= n(t) \neq 0$.

¿De dónde sale esta señal $n$(t) que podemos amplificar con el aparato de radio? ¿De dónde salen los puntos blancos y negros de un televisor sin sintonizar? Estos puntos no son causa de ninguna imperfección de los receptores. En realidad descodifican una señal real. Y esta señal la podemos encontrar tanto en receptores de señales de radio como en receptores por cable, como el teléfono, o incluso vía fibra óptica.

El ruido –$n(t)$–, de hecho, no lo produce un fenómeno único, sino que es consecuencia de la acción de diferentes fenómenos. Por lo tanto, si queremos una expresión de $r(t)$ final, tenemos la siguiente:

$$r(t) = h(t) * s(t) + n(t).$$

Siempre encontraremos ruido, y eliminarlo no es fácil en todos los casos. Por lo tanto, se intenta trabajar siempre con niveles de señal que estén muy por encima de los niveles de ruido para que su efecto sea mínimo.[9]

A continuación estudiaremos los diferentes tipos de ruido con los que nos podemos encontrar.

*Ruido térmico*

El hecho de que todos los cuerpos materiales se encuentren a cierta temperatura diferente del cero absoluto (0 K) hace que las partículas atómicas que los forman vibren. Cuanto más alta es la temperatura, mayor es la vibración de protones, neutrones y electrones. Esta vibración es mucho mayor en los electrones que en los neutrones y los protones, porque su masa es inferior. Podemos imaginar una red de núcleos atómicos prácticamente fijos y una nube de electrones oscilando de forma errática.

Este movimiento aleatorio de los electrones se superpone al movimiento que los electrones experimentan dentro de los conductores, y lo vemos como un ruido. Puesto que se debe a la temperatura de los cuerpos, lo llamamos *ruido térmico*.

9. Hay algunos tipos de ruido que no se pueden atenuar de esta manera, como veremos más adelante.

Desde el punto de vista espectral, este ruido térmico presenta un espectro plano; es decir, lo podemos encontrar en cualquier frecuencia de trabajo. Por analogía con los colores, cuando una señal tiene un espectro plano, se denomina *señal blanca*.

El tipo de señal generada por el movimiento de los electrones se denomina *ruido blanco*, y se caracteriza por una potencia de salida que depende exclusivamente de la temperatura del material y del ancho de banda que estamos usando.

**El ruido térmico es una variable aleatoria**

El ruido térmico es una de las variables aleatorias más perfectas que se conocen. En la actualidad el ruido térmico se usa para sistemas altamente sensibles a la predictibilidad que tiene una variable aleatoria generada por ordenador, como los sistemas de generación de semillas criptográficas que se utilizan para el cifrado de datos.

Una manera de evitar el ruido térmico sería refrigerar todo el sistema a 0 K o, como mínimo, a una temperatura próxima. Sin embargo, esto es costoso. De todos modos, tenemos que pensar que la potencia de ruido generada a las temperaturas de trabajo habituales (en torno a los 300 K; es decir, 23 °C) normalmente es baja o muy baja.

**Ejemplo de refrigeración para evitar el ruido térmico**

En radioastronomía, a causa de que las señales estelares que se quieren medir son de una potencia extraordinariamente pequeña, el receptor se refrigera dentro de helio líquido a 3 K (–269 °C) para evitar al máximo el ruido térmico.

Normalmente, tendremos otros ruidos que nos preocuparán más que éste porque tienen más potencia.

*Ruido de eco*

El ruido de eco electromagnético es totalmente análogo al eco sonoro producido dentro de un valle cerrado: si gritamos, nuestro grito nos vuelve al cabo de

unos instantes. Este efecto se debe a que las montañas del valle no absorben el 100% de la onda sonora y una parte de la fracción no absorbida de potencia vuelve hacia el emisor.

Se produce ruido de eco en hilos de cobre y en fibras ópticas en cada punto de inhomogeneidad de la línea, (conexiones, empalmes, etc.). El resultado es una mezcla de la señal original y la misma señal, retardada y con menos potencia. Dado que el ruido es la misma señal y que, además, el retardo se puede calcular fácilmente, este tipo de ruido es relativamente sencillo de combatir.

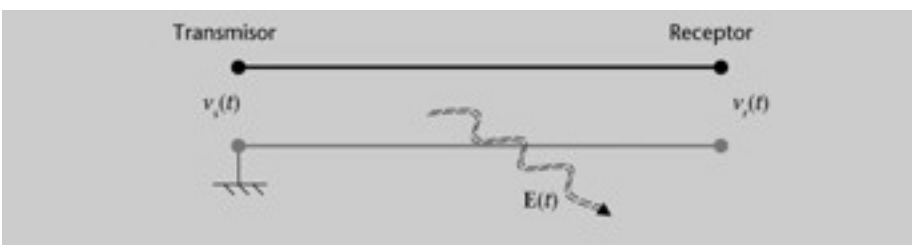
*Ruido de intermodulación o saturación*

La saturación es un ruido causado por problemas en los sistemas activos que conforman el canal de transmisión. Se manifiesta con la aparición de nuevos armónicos en el espectro, con cambios en el ancho de banda, etc.

Para evitar la saturación simplemente hay que vigilar que no entre mucha señal en los equipos activos. Esta restricción, sin embargo, está contrapuesta a las necesidades requeridas para atenuar el efecto de la mayoría de los ruidos: que el nivel de potencia de señal se mantenga muy por encima del nivel de potencia de ruido.

*Ruido de interferencia*

Como hemos comentado antes, en ciertas condiciones los medios de transmisión se pueden comportar como antenas y, por lo tanto, provocar que parte de la potencia de la señal transmitida se pierda en forma de potencia radiada:




Generación de una interferencia.
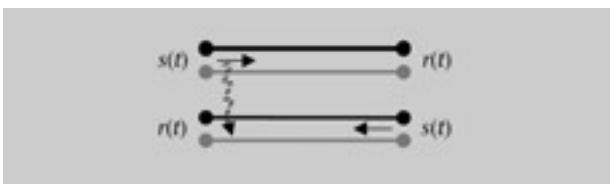
Esta pérdida por radiación no sería más que una atenuación de la señal si las líneas de transmisión estuvieran aisladas, pero las líneas de transmisión suelen estar

agrupadas, en una misma manguera plástica, o en un canalón clavado en la pared o suspendido de un techo. Y si una línea de transmisión se puede comportar como una antena, puede ser tanto una antena emisora como receptora; es decir, tanto puede radiar potencia de la señal que transmite como captarla del exterior y añadirla a la señal que transmite. Esto es lo que se conoce como *interferencia*.

Interferencia es la captación por parte de un medio de transmisión de potencia de señal radiada por otro medio de transmisión próximo. Esta señal que se superpone a la original se trata como un ruido, porque no tiene nada que ver con él.

**Interferencia NEXT**

La interferencia NEXT es un caso particular que se produce en sistemas de transmisión bidireccionales cuando la transmisión en un sentido interfiere en la transmisión en el otro sentido:




Interferencia NEXT

Lamentablemente, el ruido de interferencia no se puede eliminar del todo. Lo que haremos será disponer los elementos conductores de manera que la influencia entre los mismos sea más baja; por ejemplo, podemos alejar las líneas entre sí, hacerlas pasar en ángulo recto entre sí, etc., o utilizar conductores con geometrías que emitan (y, por lo tanto, reciban) menos interferencias. Más adelante, cuando describamos los medios de transmisión más habituales, detallaremos más las diferentes geometrías.

Este fenómeno de radiación que presentan las líneas de transmisión es más notable cuanto más alta es la frecuencia de la señal transmitida. Análogamente, también será más difícil proteger una línea de captar señales interferentes de alta frecuencia. Una demostración de lo que decimos es que a frecuencia cero no hay interferencias. Desgraciadamente, no podemos transmitir datos con una señal continua.

*Ruido impulsivo*

El ruido impulsivo, más que un ruido, es un grupo de ruidos caracterizados por su corta duración, como máximo de milisegundos. Son, más o menos, inofensivos en transmisiones analógicas (televisión, telefonía), donde sólo se manifiestan como picos espurios de señal: puntos blancos en la pantalla de televisión, *crecs* en el auricular telefónico, etc.

En cambio, en una transmisión de datos, un pico de un solo microsegundo puede estropear decenas o centenares de bits, e invalidar completamente el fichero o el programa transmitido.

Algunos de los causantes de estos picos de señal son los siguientes:

- Chispas próximas. La chispa producida por un motor eléctrico, una bujía de un motor de gasolina, el rayo de una tormenta lejana, las luces fluorescentes o la simple utilización de un interruptor son picos de tensión cortos pero que pueden ser de cierta potencia. Esta corta duración equivale a una alta frecuencia (1 μs → 1 MHz) y una alta frecuencia implica facilidad para interferir. Este tipo de ruido lo trataremos como una interferencia cualquiera.
- Imperfecciones de las fuentes de alimentación. Las fuentes de alimentación de los equipos activos no dan una tensión absolutamente fija y las pequeñas variaciones en la tensión de alimentación hacen que varíen ligeramente su comportamiento. Por ejemplo, un amplificador de G = 1.000 alimentado a 12 V quizás baja la G a 995 cuando la alimentación baja a 11,8 V. Una oscilación de la alimentación puede aparecer mezclada con la señal amplificada. Hay que utilizar fuentes de alimentación de calidad para combatir este tipo de ruido.
- Microcortes. Los cortes provocados por imperfecciones en los contactos en los que se depositan óxidos hacen que la resistencia no sea nula. Pequeños movimientos como microseísmos y vibraciones producidas por el tráfico pueden mover ligeramente los conductores y provocar picos de ruido. Si queremos evitar el ruido producido por los microcortes, tendremos que utilizar conexiones de alta calidad bañadas en oro (que no se oxida y difícilmente se corroe).

## 4.2. Tipo de medios de transmisión

Un primer criterio de clasificación de los medios de transmisión de señales electromagnéticas nos permite identificar los dos tipos siguientes según el material del que están hechos:

- Conductores. Medios compuestos por un par de conductores (normalmente de cobre) a los cuales se aplica una tensión o se inyecta una corriente.
- Dieléctricos: medios compuestos por material dieléctrico transparente a ciertas ondas electromagnéticas (el aire, el vacío, cristal, plástico, etc.), al cual se aplica una onda electromagnética.

También se puede hacer esta otra clasificación, según su geometría. Ésta será la clasificación que seguiremos para explicarlos:

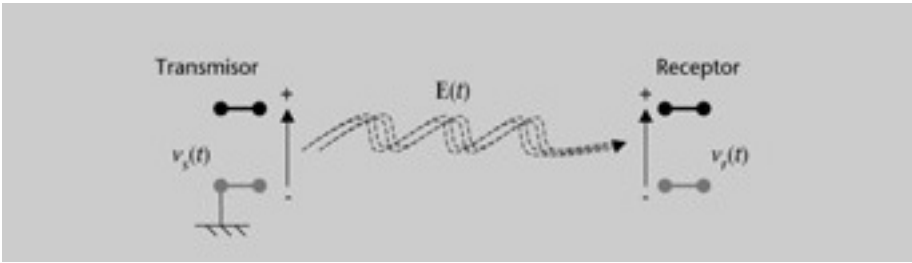
1) Medios de transmisión guiados. Medios que adoptan una disposición que evita en gran manera la dispersión de la potencia mientras se acerca a su destino. Es sinónimo de este concepto el de líneas de transmisión.
2) Medios de transmisión no guiados. Medios que no adoptan ninguna disposición particular, sino que se basan en el hecho de que el transmisor disponga de un elemento capaz de enfocar la potencia sobre un elemento captador situado en el receptor. La transmisión a través de estos medios también se denomina transmisión inalámbrica. Los medios no guiados útiles son siempre dieléctricos: el aire y el vacío.

### *4.2.1. Medios de transmisión no guiados*

La base del mecanismo de transmisión es que el transmisor genere un campo electromagnético focalizado sobre el receptor. El transmisor tiene que aplicar la señal de salida en forma de tensión $v(t)$ sobre un par de conductores metálicos. Este par de conductores se dispone de una manera peculiar que permite una conversión eficaz de tensión $v(t)$ a campo electromagnético $E(t)$[10]. Esta disposición la denominamos *antena*. A causa de la simetría de las ecuaciones de Maxwell (las que describen el comportamiento de las ondas electromagnéticas) la antena de recepción y la de transmisión tienen que ser muy parecidas.

10. Al contrario de lo que deseábamos para el ruido de interferencia.

En la figura siguiente se esquematiza el mecanismo de transmisión:




Transmisión de medios no guiados.

La forma que adopta la antena depende de la longitud de onda. Normalmente, las antenas deben tener un tamaño aproximado de una fracción de longitud de onda ($\lambda/2$ o $\lambda/4$). Por ejemplo, para longitudes de onda cortas, como las microondas (entre 3 GHz y 30 GHz), es necesaria una antena de aproximadamente:

$$\lambda/2 = c/f = 3 \cdot 10^8 \text{ m}/10 \text{ GHz} = 3 \cdot 10^{-2} \text{ m} = 3 \text{ cm},$$

mientras que para una antena de televisión (banda UHF, 300 MHz-3 GHz) necesitaremos una antena de aproximadamente:

$$\lambda/2 = c/f = 3 \cdot 10^8 \text{ m}/1 \text{ GHz} = 3 \cdot 10^{-2} \text{ m} = 30 \text{ cm}.$$

En el primer caso, la antena recibe un área muy pequeña del frente de onda, con lo cual la potencia que se podría extraer sería demasiado pequeña en la mayoría de los casos. Por ello, se usan reflectores (de forma parabólica) que concentran sobre la antena una cantidad de señal mayor. Para longitudes de onda más largas (UHF y superiores) usualmente no es necesario ningún tipo de reflector, dado que la antena *per se* ya es bastante grande.

**Usos de los medios de transmisión no guiados**

En transmisión de datos los sistemas no guiados se utilizan en radioenlaces terrestres y vía satélite. También se utilizan en redes de área local inalámbricas, telefonía móvil, mandos a distancia, etc.

Las frecuencias más usadas en transmisión de datos son las microondas. Por microondas se entienden las señales que tienen un espectro dentro del margen frecuen-

cial que va de los 3 a los 30 GHz. El ancho de banda en la franja de microondas es bastante grande:

$$BW = 30\ \text{GHz} - 3\ \text{GHz} = 27\ \text{GHz}$$

En general, cuanto más altas son las frecuencias, mayor es la capacidad para transmitir datos (más ancho de banda disponible). Sin embargo, las microondas son, hoy por hoy, el límite que nos permite tanto la tecnología como la meteorología. A frecuencias por encima de los 10 GHz la tecnología se encarece mucho. Además, a partir de esta frecuencia, las señales son sensibles a la lluvia (pensemos que las gotas de agua son de un tamaño comparable a la longitud de onda). Esto hace que uno de los cálculos más importantes a la hora de diseñar un radioenlace sea conocer cuál es el chaparrón más fuerte (medido en litros por hora) que puede tolerar sin que la transmisión se corte.
En la actualidad el problema que presentan los radioenlaces (tanto los radioenlaces tierratierra como los vía satélite) es la sobresaturación del espectro electromagnético aéreo: dentro de una misma banda de frecuencias podemos encontrar otros radioenlaces trabajando, con las interferencias consiguientes.
Si en el futuro la tecnología no aporta nuevos mecanismos de transmisión inalámbrica o nuevas frecuencias de trabajo (que, a la fuerza, deben ser más altas, ya que las inferiores están utilizadas al 100%), las transmisiones inalámbricas tienen una cota que les impedirá evolucionar a partir de cierto punto. Los medios guiados son la única escapatoria práctica, y marginaremos las transmisiones inalámbricas en entornos donde la instalación de un cable sea imposible o poco cómoda, como en transmisiones vía satélite, informática móvil, telefonía móvil, etc.

Como cualquier sistema que transmita una señal, podemos caracterizar el comportamiento de un medio no guiado con una función de transferencia $H(f)$. En el caso del vacío, esta función es trivial: permite pasar todas las frecuencias excepto el componente continuo. Si trabajamos con aire, las cosas no cambian mucho si hablamos de frecuencias inferiores a las microondas (GHz). A partir de las microondas nos encontramos con las frecuencias de absorción de los diferentes elementos que componen el aire. En caso de que el aire presente alguna inclemencia meteorológica, la función de transferencia se verá alterada con una nueva banda de absorción.

### *4.2.2. Medios de transmisión guiados*

Hoy por hoy la mayor parte de transmisiones de datos se hace sobre medios de transmisión guiados (líneas de transmisión).

Los diferentes medios de transmisión que expondremos se caracterizan por unas funciones de transferencia H($f$) diferentes, y, por lo tanto, después de ser ecuali-

zados y amplificados, dispondremos de cierto ancho de banda que se encuentra ajustado al componente continuo o bien desplazado.

Como vimos en el apartado anterior, cuando el ancho de banda se encuentre ajustado al componente continuo, hablaremos de medios de transferencia paso bajo; y cuando éste, por el contrario, se encuentre desplazado, diremos que se trata de medios de transferencia paso banda.

A continuación veremos diferentes medios de transferencia guiados: el par de hilos, el par trenzado, el par apantallado, el coaxial, la guía de microondas y las fibras ópticas.

*Par de hilos*

El medio guiado más simple consiste en un par de hilos de cobre protegidos con algún material aislante que impida el cortocircuito accidental de los dos conductores, como se ve en el esquema siguiente (el aislante no sale dibujado):




Par de hilos.

Habitualmente, el transmisor aplica una diferencia de potencial $v_s(t)$ sobre la línea que, en principio, tendría que llegar a su destino.

Las limitaciones básicas de este medio son debidas al hecho de que es muy vulnerable a las interferencias. El par de hilos tiene unas características de radiación muy buenas. El área definida entre los dos conductores nos da una idea aproximada de lo bueno que es como antena, tanto para recibir interferencias como para emitirlas:




Área de un par de hilos.

Si doblamos el área, doblamos la capacidad receptora/transmisora. Por lo tanto, doblamos la potencia de ruido captada/emitida. Además, empeoramos las características de la función de transferencia, ya que recordemos que a medida que aumenta la frecuencia también aumentan las capacidades de radiación de una señal.

Es vital disminuir el área definida entre los conductores: la separación entre los conductores tiene que ser la mínima posible. Cero sería un valor ideal (área cero, interferencia cero, etc.), pero se produciría un cortocircuito. Siempre será necesario un pequeño grosor de aislante que separe los conductores, que debe ser el mínimo necesario, pero nunca puede tender a cero, ya que se puede producir un cortocircuito.

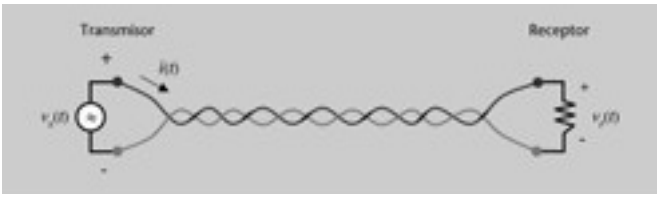
Los factores que acabamos de mencionar limitan estos tipos de líneas a distancias cortas y a frecuencias bajas:

- Alcance: 1-100 m;
- *BW*: 1-100 kHz.

Se puede aumentar un poco su alcance si se disminuye la frecuencia,o al revés, se puede aumentar su frecuencia si el alcance es corto.

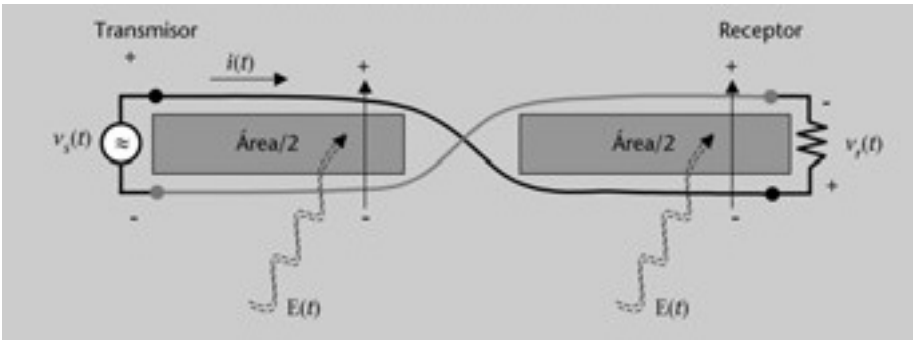
*Par trenzado*

El par trenzado es una opción para mejorar las características de transmisión del par de hilos. Trenzar los hilos mejora la atenuación, el ancho de banda y la robustez frente a las interferencias:




Par trenzado.

Al trenzar los hilos se consigue que el área sea aproximadamente cero. Veámoslo ampliando una de las trenzas:




Protección contra las interferencias.

La señal será el resultado de aplicar una mitad de la onda interferente en sentido positivo y la otra mitad, en sentido negativo; simplificando, se obtiene el resultado siguiente:

$$v_r(t) = v_s(t) = v_{E(t)}(t)/2 + (-v_{E(t)}(t)/2) = v_s(t)$$

La interferencia se cancela a sí misma. A causa de la simetría del fenómeno interferente, la radiación también es inferior.

Una sola trenza no es suficiente, ya que normalmente los frentes de onda no son homogéneos. El porcentaje de cancelación mejora si se hace un trenzado más apretado (es lo que se hace en la práctica), una trenza cada pocos centímetros. Con esto mejora mucho la calidad del par de hilos:

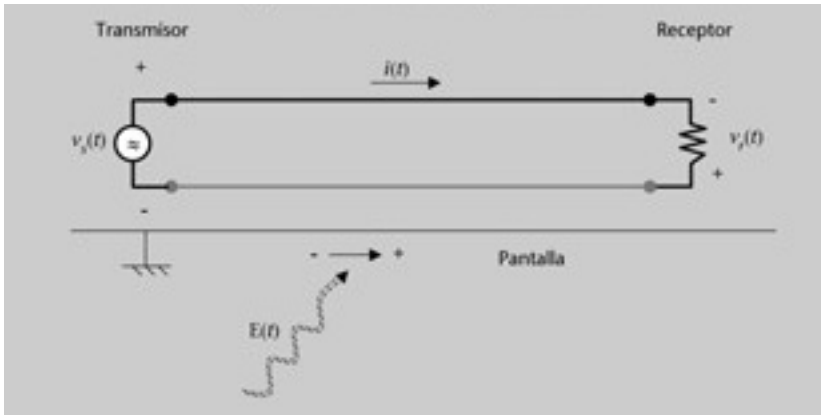
- Alcance: 100 1.000 m;
- *BW*: 10 1 MHz.

**Cableado de edificios: cableado estructurado**

El par trenzado es el tipo de cableado más usado para llevar a cabo las infraestructuras de comunicaciones de una empresa por la excelente relación calidad/precio que tiene, y la forma que toma es la del cableado estructurado.

*Par apantallado*

El apantallamiento de líneas es una manera efectiva, pero poco sutil, de luchar contra las interferencias, porque las corta de raíz. Se trata de recubrir la línea con un plano metálico que cortocircuite el campo eléctrico del frente de onda interferente:



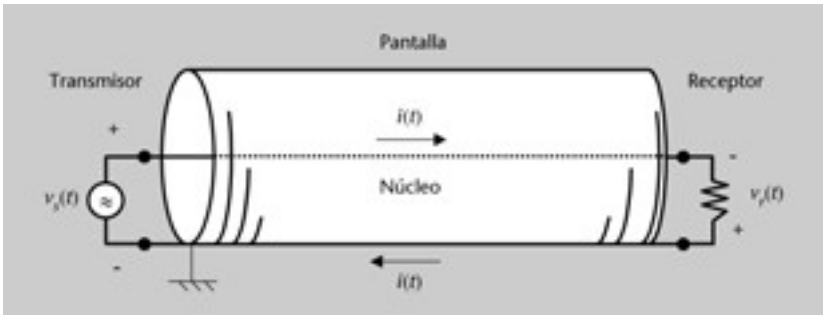

Apantallamiento de un par de hilos.

Este recubrimiento, que se denomina *pantalla*, es separado de los conductores por material aislante y, puesto que debe proteger todo el recorrido de la línea, es necesario que tenga una forma aproximadamente cilíndrica.

La presencia de la pantalla, sin embargo, tiene unos efectos negativos sobre la calidad de propagación de la señal (pérdida de *BW*); por lo tanto, se usa sólo en casos muy particulares.

*Coaxial*

Los efectos de la pantalla mejoran enormemente si la aprovechamos para transmitir la corriente de la señal. Es decir, la pantalla pasa a ser un conductor cilíndrico que recubre al otro conductor (que denominaremos *núcleo*):




Coaxial.

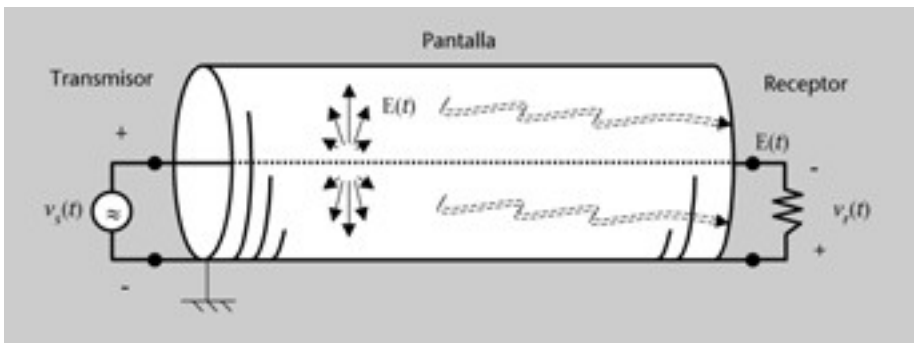
Esta mejora es debida a la simetría coaxial (nos ahorraremos sus detalles teóricos), que debe mantenerse correctamente a lo largo de todo el trayecto.

La robustez frente a las interferencias es tan buena como en el par apantallado, pero el ancho de banda aumenta de manera drástica.

- Alcance: 1 10 km;
- *BW*: 10 100 MHz.

*Guía de microondas*

En un coaxial se puede llegar a frecuencias bastante elevadas. Para distancias cortas podemos llegar hasta 1 GHz, o quizá un poco más pero tampoco demasiado, por un efecto de radiación muy curioso. En el interior del cilindro formado por la pantalla se crea un campo eléctrico originado por la diferencia de potencial entre la pantalla y el núcleo:

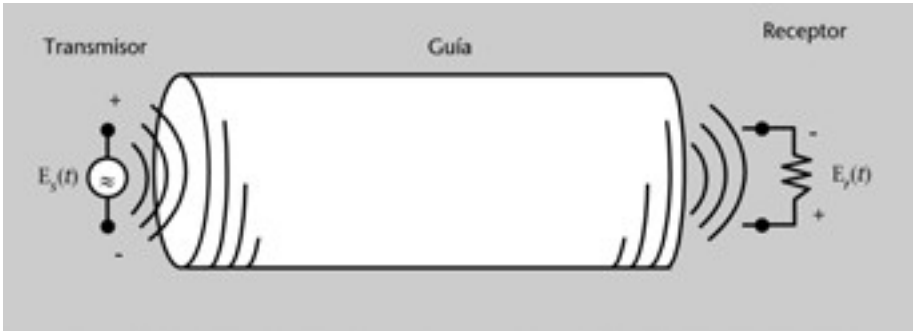



Campo eléctrico dentro de un coaxial.

El campo eléctrico generado no puede salir fuera de la pantalla; por lo tanto, si pusiéramos un receptor de campo electromagnético (una antena) en el receptor, seríamos capaces de obtener este campo E($t$), que evidentemente sería proporcional a $v_s(t)$ y, por lo tanto, teóricamente podríamos extraer sus datos a partir del campo eléctrico recibido.

A medida que se trabaja con frecuencias más altas, este efecto es más intenso, de manera que llega un punto en el que lo único que se consigue con el núcleo del coaxial es cortocircuitar la onda que viaja dentro del tubo. Para evitarlo, se puede sacar el núcleo del coaxial, dejar el tubo sin ningún tipo de núcleo

conductor y llenarlo de aislante. Para que esto funcione será necesario montar el sistema de manera diferente, porque ha desaparecido uno de los dos conductores: deberemos poner una antena para la transmisión y otra para la recepción, como se ve en la figura siguiente:




Guía de microondas.

En este caso hablaremos de transmisión de señales electromagnéticas y no de señales eléctricas, ya que la antena emitirá ondas radioeléctricas.

Hay otro detalle importante. Estas ondas electromagnéticas, aparte de tener longitud de onda, tienen un equivalente en ancho (llamémosle *ancho de banda*) que podemos considerar que es aproximadamente igual a la longitud λ. Es necesario que el diámetro de la guía de ondas encaje exactamente con un múltiplo entero de λ/2, de manera que la onda tenga uno nulo cuando se apoya dentro del tubo. Dicho de otra manera, hay que sintonizar la frecuencia con la guía (podéis ver la figura al margen).

Si tomamos el diámetro más pequeño, λ/2, encontramos que para un diámetro de 5 cm tenemos ondas de λ= 10 cm y, por lo tanto, de frecuencia:

$$f = c/\lambda = 3 \cdot 10^8 \text{ m/s} / 0{,}1 \text{ m} = 3 \text{ GHz}$$

Los tubos o guías de onda prácticos mecánicamente son aquéllos de entre 0,5 y 5 cm de diámetro; es decir, entre 3 y 30 GHz. Esta medida es exactamente el margen de frecuencias, que se corresponde con lo que denominamos *microondas*. Aunque teóricamente podemos hacer guías a otras frecuencias, las guías de microondas son las únicas prácticas.

De este modo, sin darnos casi cuenta de ello, hemos visto el primer medio de transmisión dieléctrico. Observad que, aunque el tubo es metálico, lo que conduce la señal (E(*t*)) es el aire interior al tubo (el dieléctrico).

Las guías de onda no se utilizan para transmitir a mucha distancia, sino que normalmente se usan para comunicar internamente los distintos componentes de un radar o de un radioenlace (terrestre o vía satélite); es decir, a pocos metros. También las podemos encontrar en los captadores de las antenas parabólicas y dentro de los hornos microondas.

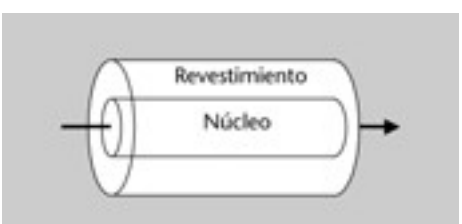
La razón del alcance limitado es que el factor forma es, en este caso, primordial: un pequeño golpe en el tubo cambiaría el ancho, que dejaría de ser el correcto y produciría una atenuación que podría ser demasiado grande.

*Fibras ópticas*

La construcción de guías de onda metálicas para trabajar con frecuencias muy altas se convierte en impracticable: un tubo metálico de una fracción de milímetro es técnicamente irrealizable.

Afortunadamente, no es obligatorio utilizar un tubo metálico para confinar la señal electromagnética dentro de la línea, se puede construir una guía de ondas totalmente dieléctrica, jugando con diferentes índices de refracción del dieléctrico.

Si se construye un tubo de dieléctrico que tenga un núcleo con un índice de refracción superior al de la pantalla, denominada en este caso revestimiento (podéis ver la figura siguiente), se consigue que al inyectar una señal por una parte, aparezca por la otra gracias al fenómeno físico de la refracción total. El diámetro del núcleo es de pocos micrómetros, y el revestimiento es de una décima de milímetro. Para mejorar la solidez mecánica, los dos cilindros van protegidos con una cubierta plástica.

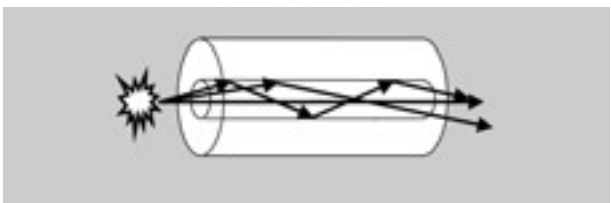



Componentes de la fibra óptica.

Esta disposición de guía de ondas se conoce con el nombre de fibra óptica, aunque en la actualidad se utilizan longitudes de onda infrarrojas. Volveremos más adelante al respecto.

En este caso, puesto que el revestimiento no es conductor, no hay peligro de cortocircuito –como ocurría en las guías de microondas. Por lo tanto, el diámetro de la guía no es importante desde el punto de vista del fenómeno de atenuación descrito en la guía de microondas. Sin embargo, sí que lo es por otra razón: la distorsión. De hecho, el fenómeno no es exactamente el de la distorsión tal y como lo hemos explicado. Dejaremos, sin embargo, las explicaciones para más adelante.

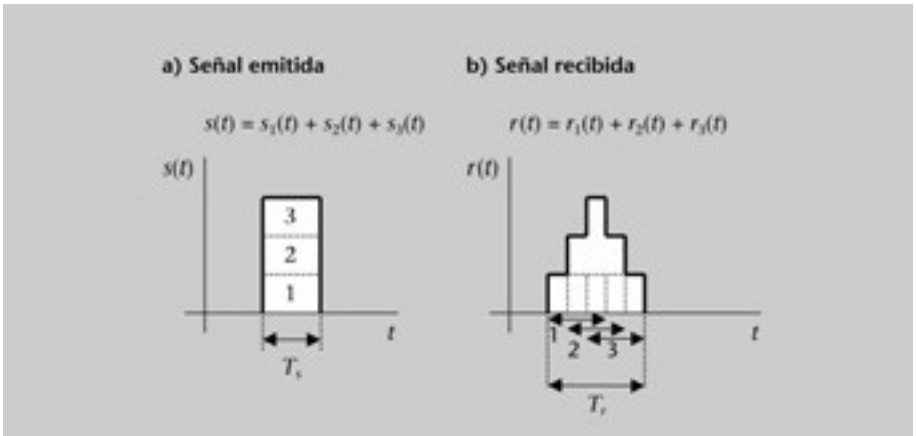
Observemos en la figura siguiente qué pasa cuando se hace entrar luz dentro de una fibra óptica:



Caminos de la luz en función del ángulo de incidencia.

En la imagen se han representado tres haces de luz posibles: un haz coaxial a la fibra, y los otros dos con un ángulo ligeramente abierto. Estos dos últimos, por refracción total, van rebotando por la fibra hasta que llegan a su destino. Dado que recorren caminos de longitud diferente, los haces llegarán también en momentos distintos.

Imaginemos que encendemos y apagamos la luz de la izquierda de la figura anterior. A la derecha la luz total será la composición de los tres haces dibujados. Puesto que recorren distancias diferentes, cada uno individualmente se apagará y se encenderá en momentos ligeramente distintos. Podemos representar gráficamente este comportamiento de la manera siguiente:




Recomposición de los diferentes haces de luz.

Si se observa el gráfico con atención, se aprecia que la señal recompuesta es claramente de duración más larga, y de que la potencia está más repartida. Ha habido una deformación de la señal que no se explica en términos de espectro o de transferencia y, por lo tanto, no es una distorsión propiamente dicha. Además, este fenómeno empeorará a medida que crezca la distancia hasta llegar a un punto en el que lo que se reciba sea totalmente inútil.

Si encadenamos datos (bits) uno tras otro, se acabarán solapando. Se producirá el fenómeno conocido como interferencia intersimbólica (ISI).

Usaremos el concepto de *interferencia intersimbólica* como concepto genérico, y restringiremos el uso de la palabra *distorsión* para los efectos producidos por la amplificación distinta de los armónicos.

La distorsión es un tipo particular de interferencia intersimbólica, ya que provoca un ensanchamiento o solapamiento de los símbolos (bits) causado por el comportamiento frecuencial del sistema (medio de transmisión).

Con respecto a la robustez frente al ruido, las fibras ópticas están blindadas. El ruido de interferencia y el impulsivo, que son los más importantes en los medios conductores, son casi inexistentes en fibras. El eco es lo único que puede aparecer en las conexiones entre fibra y fibra y, en situaciones en las que este problema pueda ser importante, las fibras se sueldan, de manera que la inhomogeneidad se reduce al mínimo, y al mismo tiempo el eco.

Podemos distinguir los dos tipos de fibras ópticas siguientes:

a) Fibras ópticas multimodo

Los diferentes haces que salen de la fuente de luz son contables. Dicho de otra manera, la fuente de luz no genera un haz disperso y continuo dentro de la fibra, sino que la luz entra sólo en un número entero de ángulos diferentes. Cada uno de los ángulos posibles se denomina modo.

Las fibras ópticas que admiten más de un modo se denominan fibras ópticas multimodo.

En una fibra óptica multimodo cabe aproximadamente una treintena de haces, lo cual limita su alcance:

- Alcance: 1 10 km;
- *BW*: 10 100 MHz.

Estas fibras se hacen de vidrio ($SiO_2$) o bien de plástico (material más utilizado por razón de su bajo coste). Normalmente, la fuente de luz que se utiliza es un LED (*Light-Emitting Diode),* también por su coste bajo, que suele emitir luz infrarroja.

b) Fibras ópticas monomodo

Podemos evitar la ISI de manera radical usando un solo modo, y lo conseguiremos con un diámetro de núcleo suficientemente pequeño.

Los tipos de fibras que únicamente admiten un solo modo se denominan fibras ópticas monomodo.

El alcance de las fibras ópticas monomodo sólo está limitado por la transparencia del dieléctrico a la luz inyectada.

En los años setenta se descubrió que el cristal de óxido de silicio ($SiO_2$) tiene una transparencia extraordinaria a ciertas frecuencias del infrarrojo. Estas frecuencias son tres ($\lambda$ = 850 nm, 1.300 nm y 1.500 nm*),* de las cuales la que tiene mejores características de transmisión es la de longitud de onda más larga (1.500 nm).

La luz dentro de una fibra óptica monomodo no se puede introducir con un LED, porque un LED emite los fotones en todas direcciones (en todos los modos).

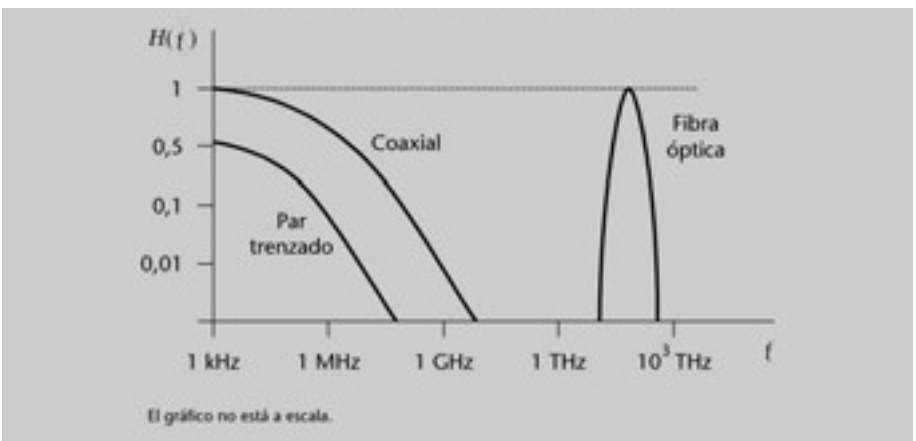
Si tomáramos un solo modo, la potencia sería una fracción del total, que sería irrisoria. Es necesaria una fuente de luz capaz de generar bastante potencia en un solo modo. Esta fuente existe, y se denomina láser (palabra que proviene de las iniciales en inglés de *Light Amplification by Stimulated Emission of Radiation* amplificación de luz por emisión estimulada de radiación).

Tanto las características de pureza del vidrio de la fibra como el hecho de necesitar láser en lugar de LED encarecen el sistema. Por ello sólo se usan fibras monomodo donde no es suficiente con las multimodo; es decir, para las largas distancias y los anchos de banda grandes:

- Alcance: 10 100 km;
- BW > GHz.

El límite en el ancho de banda lo causa más la electrónica necesaria que las limitaciones de la fibra monomodo.

Para acabar, mostramos una figura que resume las funciones de transferencia típicas de los medios de transmisión guiados que hemos visto:




Funciones de transferencia (para 1 km).

Capítulo III

# Enlace de datos

*José M. Barceló Ordinas*
*Llorenç Cerdà Alabern*

## Introducción

Tanto en las redes de gran alcance como en las de área local hemos hablado de conexiones punto a punto. Se trata del enlace que une directamente entre sí dos entidades sobre el cual se transmiten datos. También tienen sentido los enlaces punto a punto por sí mismos y no como elementos de una red, por ejemplo cuando se pretende conectar un ordenador y una impresora.

La conexión punto a punto se hace con medios de transmisión que, como ya sabemos, en general no son perfectos y están sometidos a diferentes fenómenos físicos que acaban produciendo errores de transmisión. En otras palabras, los bits que se reciben al otro extremo de una conexión no son exactamente los mismos que se han transmitido desde el otro.

El nivel de enlace es el encargado de detectar estos errores e intentar, de alguna manera, solucionarlos, corrigiéndolos o solicitando la retransmisión de las tramas erróneas recibidas. Veremos los diferentes mecanismos de retransmisión y a qué casos se adecuan.

El control de errores comporta un cierto procesamiento en la transmisión y la recepción. Por ello, se debe añadir un control de flujo, para poder abortar el proceso si la velocidad de llegada de tramas a un terminal es superior a la velocidad de procesamiento de dicho terminal.

## 1. Descripción del nivel de enlace

El nivel de enlace o DLC (Data Link Control) es el software que controla la transmisión de información sobre el flujo no fiable de bits del nivel físico. En este apartado profundizaremos en este concepto y en la unidad de transmisión de información del nivel de enlace: la trama.

Igual que en los otros niveles de la arquitectura de comunicaciones (excepto el nivel físico), el nivel de enlace consiste en dos programas o procesos que se ejecutan en computadores diferentes y que se comunican entre sí. Para que estos dos procesos se puedan comunicar es necesario establecer:

- un formato para la información que se intercambian;
- un conjunto de reglas de comportamiento.

El DLC está formado por el conjunto de protocolos necesarios para la transmisión de datos sobre un enlace de comunicación punto a punto. En este capítulo nos referiremos a cada uno de estos procesos como *DLC*, *entidad de nivel de enlace* o simplemente *nivel de enlace*.

Una de las funciones básicas del nivel de enlace es el control de errores, porque asumimos que el medio de transmisión que hay "por debajo" no es perfecto e introduce errores de transmisión. Para hacer el control de errores es necesario destinar una parte de los bits que se intercambian los DLC a la detección y la posterior gestión de los errores. Esto implica que el DLC debe ser capaz de distinguir los bits que forman parte de los datos de información de los que forman parte de los datos de control. Para conseguirlo, el DLC organiza los datos en unas estructuras denominadas *tramas* (*frames*, en inglés).

Las tramas son estructuras ordenadas de bits organizados en campos. Según la posición de los bits en la trama, el DLC puede reconocer el campo al que pertenecen. Hay campos con bits de información y campos con bits de control.

Se puede dar el caso de que el nivel de enlace reciba las tramas más rápidamente de lo que es capaz de procesarlas. En este caso se tiene que "frenar" el nivel de enlace remoto para evitar que la memoria temporal en la que están almacenadas las tramas pendientes de procesar se agote. Esta función se conoce como control de flujo.

Las tareas básicas del nivel de enlace son las siguientes:

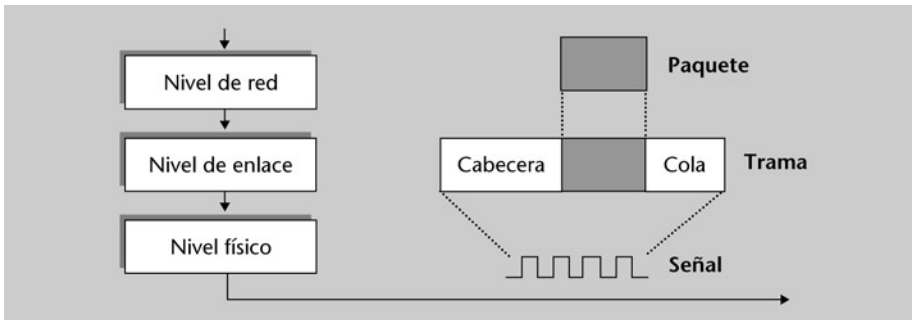
- La gestión de las tramas.
- El control de errores.
- El control de flujo.

## 1.1. Montaje de las tramas

Las tramas contienen bits de datos y bits de control. Esta información de control se estructura en campos, algunos de los cuales están antes de los datos, la cabecera, y otros detrás: la cola.

Qué campos hay, de qué tamaño son y su significado depende de las implementaciones del nivel de enlace. En general, habrá un campo donde esté la longitud de la trama, un campo con el tipo de datos que transporta y, como decíamos antes, la información adicional necesaria para hacer el control de errores.

En el contexto de las arquitecturas de protocolos como OSI, el nivel de enlace está por debajo del nivel de red. Esto quiere decir que los datos que van insertados en las tramas son paquetes de red.




Montaje de las tramas en el nivel de enlace.

## 1.2. Sincronismo de trama

Hasta ahora hemos hablado siempre del nivel físico como un medio capaz de transportar un flujo de bits. En muchos casos, ésta es una descripción correcta de

la visión que tiene el nivel de enlace del nivel físico. Hay, sin embargo, algunos medios físicos que tienen como unidad de transmisión el carácter, que se define como un bloque fijo de bits[1]. Este caso se conoce con el nombre de transmisión orientada a carácter.

De hecho, se habla de dos tipos de protocolos, uno para cada uno de los tipos de transmisión mencionados:

- Un protocolo orientado a carácter es un protocolo de nivel de enlace diseñado para ir sobre una transmisión orientada a carácter.
- Un protocolo orientado a bit es un protocolo de nivel de enlace diseñado para ir sobre una transmisión orientada a bit.

El sincronismo de trama (en inglés, *framing*) es el mecanismo que utiliza el nivel de enlace para determinar el inicio y el final de una trama dentro del flujo de bits o caracteres que llega del nivel físico.

Para poder determinar el inicio y el final de trama es necesario marcarlos de alguna manera. La solución adoptada depende del tipo de transmisión:

1) Para indicar el inicio de trama:
   - Si la transmisión es orientada a carácter, el inicio de trama se indica con un carácter especial denominado carácter de inicio de trama.
   - Si la transmisión es orientada a bit, se indica con una combinación especial de bits denominada indicador (*flag*) de inicio de trama.
2) Para indicar el final de trama hay dos técnicas:
   a) Utilizar un carácter especial denominado carácter de final de trama o una combinación especial de bits llamada indicador (*flag*) de final de trama, según si la transmisión es orientada a bit o a carácter.
   b) Utilizar un campo de longitud que indica la medida de la trama.

Algunos protocolos utilizan las dos técnicas conjuntamente para llevar a cabo el control de errores. De este modo, si el carácter de final de trama o el indicador, según el caso, no llega al final de la trama indicado por el campo de longitud, se detecta un error de delimitación de trama, o error de *framing*.

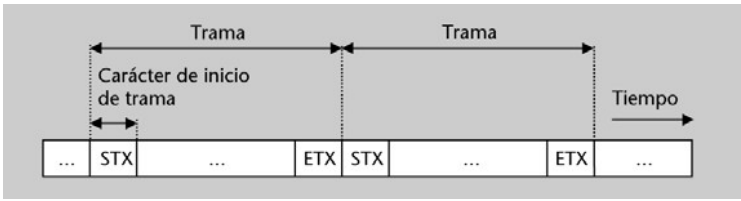
1. Normalmente cada carácter es de 8 bits.

**Ejemplos de sincronización de trama**

Presentamos un par de casos en los que se utiliza la sincronización de trama, según el tipo de transmisión del que se disponga:

**1) Sincronización de trama en una transmisión orientada a carácter**

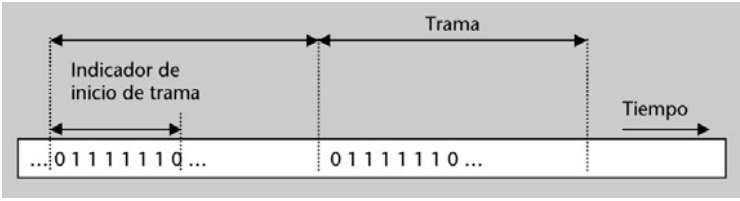
Tanto el código ASCII como el código EBCDIC tienen los caracteres de control STX (del inglés *Start of TeXt*) y ETX (del inglés *End of TeXt*). Algunos de los protocolos orientados a carácter más extendidos utilizan estos caracteres en el sincronismo de trama. La figura siguiente muestra cómo sería el sincronismo de trama con estos dos caracteres:




Sincronización de trama con los caracteres STX y ETX en una transmisión orientada a carácter.

**2) Sincronización de trama en una transmisión orientada a bit**

Como ejemplo de sincronización de trama, consideraremos la figura siguiente, que muestra los indicadores que utiliza el protocolo HDLC:




Sincronización de trama con los indicadores en el protocolo HDLC.

En el protocolo HDLC se define el indicador de final de trama con el mismo conjunto de bits que el de inicio de trama. El protocolo permite que, si hay dos tramas consecutivas, el indicador de final de trama sea también el de inicio de trama de la siguiente, y así se ahorra la transmisión de este indicador.

### *1.2.1. Mecanismo de transparencia*

Dado que los datos de información que transporta la trama son arbitrarios, si no se utiliza el campo de longitud se puede dar el caso de que, en una transmisión orientada a carácter, uno de los caracteres de datos coincida con el carácter

de final de trama. De la misma manera, en una transmisión orientada a bit puede darse el caso de que un conjunto de bits de datos coincida con el indicador de final de trama.

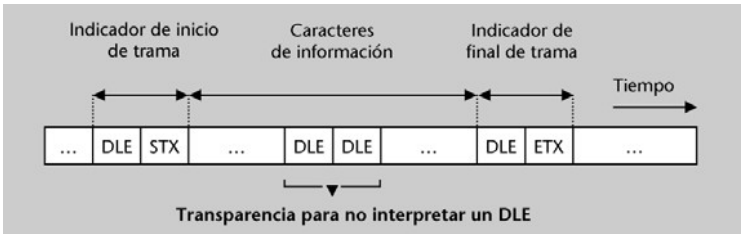
Cualquiera de las dos situaciones mencionadas en el párrafo anterior se interpretaría erróneamente como el final de trama. Para evitarlo se utiliza un mecanismo de transparencia.

**Ejemplos de mecanismos de transparencia**

Según los dos casos que acabamos de plantear, podemos encontrar mecanismos de transparencia como los que presentamos a continuación:

**1) Mecanismo de transparencia en una transmisión orientada a carácter**

El ejemplo de la figura siguiente muestra el mecanismo de transparencia del protocolo BSC. Para conseguir la transparencia se utiliza un tercer carácter, el DLE (del inglés, *Data Link Escape*).




Mecanismo de transparencia en el protocolo BSC (Inserción de caracteres o *character stuffing*).

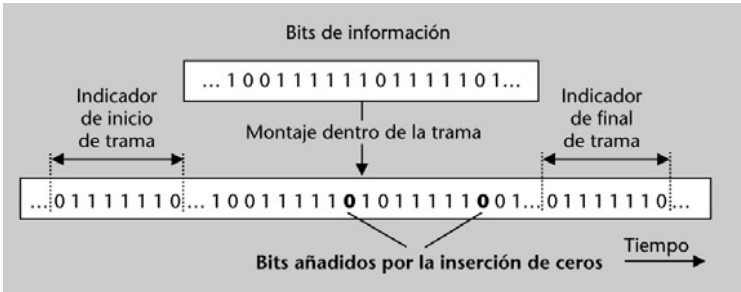
Tal y como muestra la figura, los caracteres STX y ETX van precedidos de un DLE. Además, cada vez que el transmisor encuentra un DLE entre los datos de información, independientemente del carácter siguiente, lo envía insertando previamente otro DLE. Esta técnica para conseguir la transparencia se conoce con el nombre de inserción de caracteres (en inglés, *character stuffing*).

Con respecto a la recepción, después de haber recibido los caracteres DLE-STX, se investiga cada uno de los caracteres de información recibidos. Si es un carácter DLE seguido de otro carácter DLE, se descarta uno de los caracteres DLE y se procesa el carácter siguiente (sin interpretar que iba precedido de un carácter DLE). Si es un carácter DLE seguido de un carácter ETX, se interpreta como final de trama. De esta manera:

a) El receptor elimina todos los caracteres de final de trama (DLE) sobrantes insertados por el transmisor.
b) Si resultara que dentro de los datos de información hubiese una secuencia DLE-ETX, la inserción del carácter DLE haría que el receptor no interpretara el carácter ETX como final de trama.

**2) Mecanismo de transparencia en una transmisión orientada a bit**

La figura siguiente muestra el mecanismo de transparencia del protocolo HDLC. En este caso el indicador de delimitación de trama es 01111110:




Mecanismo de transparencia en el protocolo HDLC (Inserción de ceros o *bit stuffing*).

La técnica consiste en añadir un 0 (los marcados en la figura) cuando hay cinco 1 seguidos, independientemente del bit que haya a continuación. La inserción de bits "extras" para conseguir la transparencia se conoce con el nombre de inserción de ceros o, en inglés, con el nombre de *bit stuffing*.

En recepción, si llegan cinco 1 seguidos de un 0, el 0 se elimina y se procesa el bit siguiente sin tener en cuenta los cinco 1 anteriores. De esta manera:

a) El receptor elimina todos los 0 "extras" insertados por el transmisor.
b) Si se produjese la coincidencia de que dentro de los bits de información hubiera el indicador 01111110, la inserción del bit 0 haría que el receptor no lo interpretara como el final de trama.

## 2. Detección de errores

Supongamos que el nivel de enlace es capaz de delimitar perfectamente las tramas del flujo de bits que recibe del nivel físico. Ahora nos queda el problema de detectar cuáles de estas tramas tienen uno o más bits erróneos. En este apartado veremos algunos conceptos básicos de la detección de errores.

Las tramas pueden tener una combinación de bits arbitraria. Por lo tanto, si hay error en un bit de la trama o más, la nueva combinación es otra posible trama. Así pues, sólo mirando los bits de la trama no es posible averiguar si hay alguno erróneo. Es necesario añadir bits extras que nos permitan la detección de errores.

Los bits extras se calculan a partir de los bits que hay que proteger, tal y como se explica a continuación. Observad, sin embargo, que al añadir estos bits estamos utilizando más bits de los estrictamente necesarios para transmitir el paquete. Por este motivo se dice que los códigos detectores de errores se basan en añadir redundancia a los bits de datos que se quieren proteger.

Supongamos que el tamaño de los datos que queremos proteger es de $P$ bits, y que añadimos $Q$ para poder hacer la detección de errores.

En este contexto se denomina código detector de errores, o simplemente código, a la transformación que se aplica a los $P$ bits que queremos proteger para convertirlos en la nueva combinación de $P + Q$ bits. Nos referiremos a esta nueva combinación como palabra código.

Esta transformación es biunívoca; es decir, a cada combinación determinada de los $P$ bits que hay que proteger le corresponde una sola combinación de $P + Q$ bits, y viceversa.

En este apartado nos referiremos a las tramas como *palabras código*, que es el término que se utiliza en el contexto de la detección de errores.

Hay que obsevar que el número de palabras código válidas es igual al número de combinaciones posibles de los bits de los datos que queremos proteger ($2^P$). En cambio, el número posible de palabras código es de $2^{P+Q}$. Las $2^{P+Q} - 2^Q$ combinaciones restantes son palabras código no válidas y no se utilizarán nunca en transmisión (porque no pueden resultar nunca de la transformación que aplica el código a los $P$ bits que deben protegerse).

Detectamos que hay un error cuando los bits erróneos transforman una palabra código válida en una no válida. Si como consecuencia de los bits erróneos resulta otra palabra código válida, no detectaremos el error.

Hay numerosos códigos detectores de error (los más utilizados se explicarán a continuación). Para compararlos, la medida ideal sería determinar la probabilidad de que se produjera un error no detectado en la transmisión de una trama. Desafortunadamente, esta probabilidad depende de las características del medio de transmisión y del código detector de errores, por todo lo cual resulta difícil de determinar.

Para medir empíricamente la probabilidad de una trama errónea no detectada tendríamos que contabilizar todas las tramas erróneas no detectadas y todas

las tramas erróneas que se producen en el enlace. La relación entre estos dos valores sería la probabilidad buscada.

A causa de las dificultades que comporta determinar la probabilidad de una trama errónea no detectada, consideraremos los tres parámetros para medir la robustez de un código detector de errores que se explican a continuación:

- La distancia mínima del código (distancia de Hamming del código).
- La capacidad de detección de ráfagas de error (*burst detecting capability*).
- La probabilidad de que una combinación arbitraria de bits sea aceptada como palabra válida.

Como veremos a continuación, para tomar estas medidas no debemos tener en cuenta el tipo de errores que introduce el medio de transmisión. Es decir, estas medidas dan idea de la facilidad que tiene un código para determinar ciertos tipos de errores. A la hora de elegir un código u otro habrá que tener en cuenta qué tipo de error introduce el medio de transmisión para elegir el código más adecuado, es decir, el que minimice la probabilidad de tener una trama errónea no detectada.

## 2.1. Distancia de Hamming

Para definir la distancia de Hamming de un código, primero es necesario introducir el concepto de *distancia de Hamming entre dos palabras código*.

La distancia de Hamming entre dos palabras código se define como el número de bits diferentes que hay entre estas palabras.

La distancia mínima de un código, o distancia de Hamming de un código, se define como la menor distancia que hay entre dos palabras válidas del código.

De esta definición se deduce que si la distancia de Hamming de un código vale $D_H$, cualquier combinación de $n$ bits erróneos que cumpla $n < D_H$ se detectará con probabilidad 1.

De la definición de la distancia de Hamming de un código deducimos también que un método exhaustivo para calcularla sería considerar todas las parejas posibles de palabras válidas, observar cuántos bits diferentes hay y tomar el mínimo.

En la práctica, generalmente no se aplica este método, sino que el cálculo se hace a partir de las propiedades del código.

Cuanto mayor es la distancia de Hamming, más bits erróneos debe haber para que se produzca un error no detectado y, por lo tanto, el código detector de errores será mejor.

## 2.2. Capacidad de detección de una ráfaga de error

Muchas veces los errores no se producen en bits aislados, sino que son originados por chispas que afectan a distintos bits consecutivos. Normalmente, sin embargo, una chispa no introduce errores en todos los bits que coinciden con su duración. Según las variaciones eléctricas de la intensidad de la chispa, algunos bits cambian, con lo cual se produce un error, y otros no cambian.

En una trama se define la ráfaga de error como el número de bits que hay entre el primer bit erróneo y el último, ambos incluidos.

La capacidad de detección de una ráfaga de error se define como el entero mayor, llamémosle *B*, tal que el código es capaz de detectar todas las ráfagas de error menores o iguales que *B*.

Evidentemente, cuanto mayor sea la capacidad de detección de ráfagas de error, mejor será el código detector de errores.

La capacidad de detección de ráfagas de error es especialmente importante cuando el medio de transmisión tiene tendencia a introducir los errores en forma de ráfagas. En este caso, cuanto mayor sea la capacidad de detección de ráfagas, menor será la probabilidad de tener una trama errónea no detectada.

## 2.3. Probabilidad de que una combinación arbitraria de bits sea aceptada como palabra válida

Hemos visto que si el número de bits erróneos de una trama no excede la distancia de Hamming ni la capacidad de detección de ráfagas, la trama errónea se detectará con probabilidad 1. En caso contrario, hay dos posibilidades:

a) La palabra código correspondiente a la trama errónea coincide con otra palabra código válida y, por lo tanto, no se detecta el error.
b) La palabra código resultante es una palabra no válida y se detecta el error.

El cálculo exacto de la probabilidad de que la trama errónea no sea detectada no es obvio. No obstante, podemos deducir de manera intuitiva un valor aproximado, haciendo la siguiente suposición: que la palabra código correspondiente a la trama errónea pasa a ser, con la misma probabilidad, cualquier otra palabra código. Esto equivale a suponer que se elige una combinación arbitraria de bits. Si esta combinación es una palabra código válida, no se detectará el error; si no lo es, el error se detectará.

Dado que una palabra código tiene una medida de $P + Q$ bits, cada una de las combinaciones arbitrarias posibles se puede recibir con una probabilidad de $1/2^{P+Q}$. Dado que hay $2^P$ palabras código válidas, la probabilidad de que una combinación arbitraria de bits sea aceptada como una palabra válida será $2^P/2^{P+Q} = 2^Q$.

La probabilidad de que una combinación arbitraria de bits sea aceptada como una palabra válida es $2^{-Q}$, donde $Q$ es el número de bits que añade el código detector de errores.

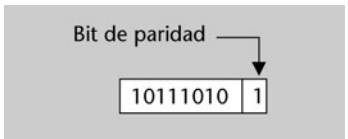
Cuanto mayor sea $Q$, menor será esta probabilidad y mejor será el código. Esto demuestra que, cuantos más bits añade el código detector de errores, más difícil es que se produzca un error no detectado.

## 2.4. Control de paridad

El código detector de errores más sencillo que hay es el denominado control de paridad. El control de paridad consiste en añadir un solo bit –denominado bit de paridad– al bloque de bits que se quiere proteger.

El valor del bit de paridad se elige de manera que la suma de los 1 de la palabra código sea un número par (paridad par). Si el criterio es que la suma sea un número impar, se habla de paridad impar.

El gráfico siguiente muestra un ejemplo de paridad par (la suma de todos los 1 tiene que ser un número par).




Código de paridad par.

### Paridad

Cuando se utiliza la paridad como código detector de errores, no debemos imaginar que toda la trama es protegida con un solo bit de paridad. La paridad normalmente se usa en transmisiones orientadas a carácter, y las tramas de estas transmisiones están formadas por más de un carácter, cada uno con su bit de paridad.

### Cálculo de la paridad par

La paridad par se suele calcular haciendo la operación OR exclusiva (XOR) de los bits que se quieren proteger.

| ***a*** | ***b*** | ***a* XOR *b*** |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Cálculo de la paridad par.

### El código detector de errores más habitual

La paridad es el código detector de errores que más se utiliza en transmisiones orientadas a carácter. Por ejemplo, la transmisión por el puerto serie de un PC es orientada a carácter. El dispositivo que controla el puerto serie (denominado *UART*, *Universal Asynchronous Receiver Transmitter*) añade automáticamente un bit de paridad a cada carácter transmitido. Con respecto a la recepción, el UART también controla automáticamente que el bit de paridad sea correcto; en caso contrario, se produce una condición de error.

Si en la transmisión de la palabra código se produce un solo error (un 1 pasa a valer 0 o un 0 pasa a valer 1), la paridad de la palabra código cambiará y no coincidirá con la del bit de paridad. Por lo tanto, se detectará el error. Sin embargo, si hay dos errores, la paridad será la misma y el error no se detectará. Deducimos, pues, que con el bit de paridad el código permite detectar un número impar de bits erróneos.

Ahora deduciremos cuánto valen los tres parámetros que hemos introducido para medir la robustez de un código:

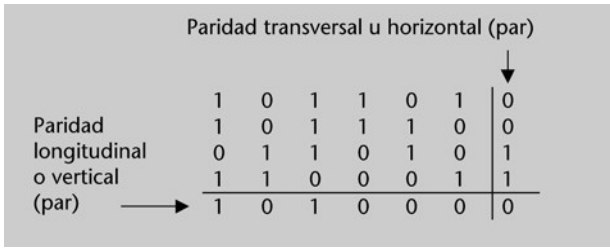
a) Para cualquier palabra válida, si se cambia un bit se obtiene una palabra no válida, y si se cambian dos se obtiene otra palabra válida. Deducimos, pues, que la diferencia mínima entre dos palabras código válidas es de dos bits y, por lo tanto, que la distancia de Hamming vale 2. En consecuencia, el código es capaz de detectar con probabilidad 1 todas las combinaciones de bits erróneos inferiores a 2 (es decir, el código detecta un bit erróneo, como habíamos visto anteriormente).
b) Dado que el código no detecta una ráfaga de error igual a 2 (dos bits consecutivos erróneos), la capacidad de detección de ráfagas vale 1.
c) Finalmente, la probabilidad de que una combinación arbitraria de bits sea aceptada como palabra válida es: $2^{-Q} = 2^{-1} = 0{,}5$, es decir, de los caracteres que tengan muchos bits erróneos, sólo se detectarán la mitad.

## 2.5. Códigos de paridad longitudinal y transversal

Una manera de mejorar la robustez del código detector de errores por medio de bit de paridad consiste en organizar los bits que hay que proteger en una matriz, y calcular la paridad de las filas y de las columnas.

La paridad de las filas se denomina paridad transversal (u horizontal), y la de las columnas, paridad longitudinal (o vertical).

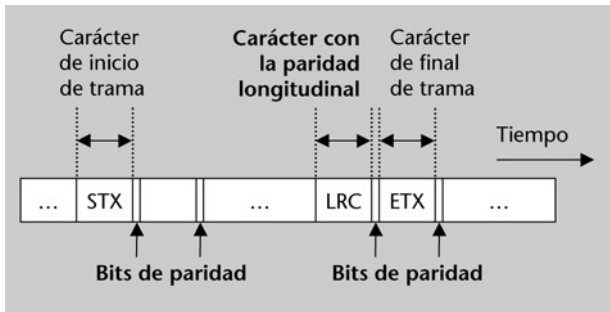
Así pues, la información se transmite organizada en bloques con su respectiva paridad longitudinal y transversal. La transmisión del bloque se hace por filas, de manera que los últimos bits transmitidos son los bits de la paridad longitudinal. La figura siguiente muestra uno de estos bloques:




Códigos de paridad longitudinal y transversal.

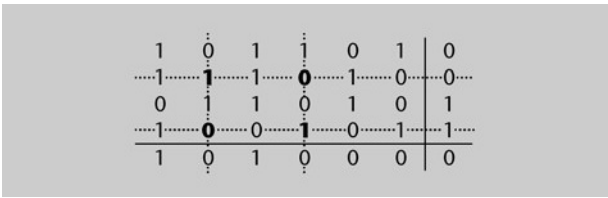
El código de paridad longitudinal y transversal se acostumbra a utilizar en transmisiones orientadas a carácter. El dispositivo transmisor añade automáticamente un bit de paridad a cada carácter, y de esta manera la trama queda formada por un conjunto de caracteres a los cuales se añade un carácter con la paridad longitudinal.

La figura siguiente nos muestra la disposición que acabamos de explicar de los bits de paridad dentro de la trama. A causa de este carácter extra, este código detector de errores se conoce también como LRC (*Longitudinal Redundancy Check*) o BCC (*Block Check Character*).




Transmisión de una trama con los bits de paridad y el carácter de paridad longitudinal LRC.

Para analizar este código investigaremos primero los casos en los que una combinación de errores no sería detectada. De la definición de paridad deducimos que este código detectará todas las combinaciones de bits erróneos que tengan un número impar de errores en alguna fila o columna. Es decir, no se detectarán las combinaciones de bits erróneos que tengan un número par de errores en todas las filas y columnas simultáneamente. El caso más sencillo es el que muestra la figura siguiente:



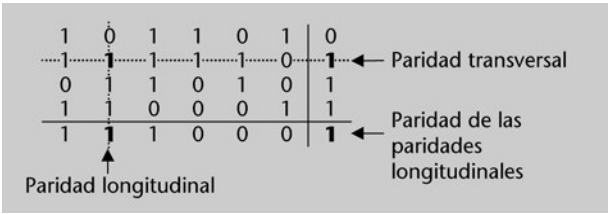

Combinación de errores que no sería detectada.

**Cálculo del carácter LRC**

El carácter LRC se suele calcular haciendo la operación XOR (paridad par) de los caracteres que se quieren proteger.

Deducimos el valor de los parámetros introducidos para medir la robustez de un código:

a) Si en un bloque cambia uno de los bits que hay que proteger, cambiarán, además, las paridades transversal, longitudinal y la paridad de las longitudinales: cambian 4 bits, y por lo tanto $D_H = 4$.




Bits que cambian entre dos palabras código válidas consecutivas.

b) Para determinar la capacidad de detección de ráfagas tenemos que encontrar la ráfaga mínima no detectada. A partir de la figura "Combinación de errores que no sería detectada" es fácil deducir que la ráfaga mínima no detectada se produce cuando los cuatro bits erróneos son adyacentes, y su tamaño es igual a la longitud de una fila más dos. Así pues, la capacidad de detección de ráfagas es la longitud de una fila más uno.

c) La probabilidad de que una combinación arbitraria de bits sea aceptada como palabra válida es $1/2^{\text{longitud fila+longitud columna}-1}$.

## 2.6. Códigos detectores de errores CRC

Como hemos comentado, los códigos detectores con bit de paridad están indicados para transmisiones orientadas a carácter. Para transmisiones orientadas a bit no son útiles, porque las tiras de bits donde se podría aplicar la paridad son mucho más largas y perderían efectividad. En lugar del bit de paridad se utilizan los denominados códigos CRC (*Ciclic Redundancy Check)* o *polinomiales*.

Los códigos detectores de errores polinomiales se basan en el cálculo de un número binario, lo que realmente es el CRC, resultado de una cierta operación matemática efectuada con los bits que se tienen que proteger. Este número se pone en el campo de control de errores de la trama. En recepción se repite el cálculo y se interpreta que hay o no hay error, en función de si coincide o no con el CRC recibido.

Para explicar el cálculo del CRC es conveniente representar las secuencias de bits como polinomios, tal y como se explica a continuación. Nos referiremos a los bits de una secuencia $s$ de $K$ bits como $s_{K-1}$, $s_{K-2}$, …, $s_0$. Definimos la representación polinomial $s(x)$ de la secuencia $s$ de la manera siguiente:

$$s(x) = s_{k-1}\, x^{k-1} + s_{k-2}\, x^{k-2} + \ldots + s_1 x + s_0$$

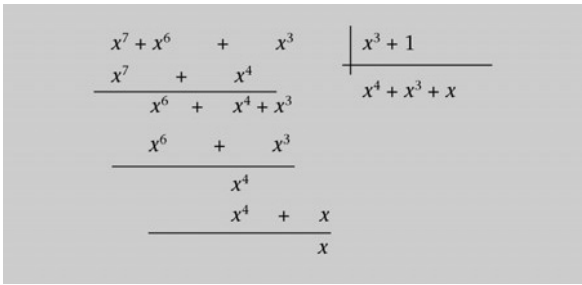
El objetivo de las potencias $x^j$ es distinguir el peso del bit $s_j$ dentro de la secuencia. Supongamos que la secuencia $s$ es la formada por los bits de la trama que queremos proteger. A los bits $s$ de la trama se añade una secuencia de $Q$ bits, llamémosla $c$, para la detección de errores. Estos bits se conocen como el *CRC de la trama*. La representación polinomial de la trama, llamémosla $t$, formada por los bits $s$ y el CRC será, pues:

$$t(x) = s(x) \cdot x^Q + c(x).$$

El cálculo del CRC se especifica en función de un polinomio generador $g(x)$, de manera que el polinomio $c(x)$ se define como el residuo de la división en módulo 2 entre los polinomios $s(x) \cdot x^Q$ y $g(x)$, es decir: $c(x) = \text{Residuo}[(s(x) \cdot x^Q) / g(x)]$. En módulo 2, el resto es igual a la suma, que a la vez es la operación XOR binaria.

**Ejemplo de cálculo del CRC**

Supongamos que la secuencia de bits que hay que proteger es "11001", con un CRC de tres bits, y que el polinomio generador es $g(x) = x^3 + 1$. Tenemos que $s(x) = x^4 + x^3 + 1$, $s(x) \cdot x^3 = x^7 + x^6 + x^3$, y la división en módulo 2 de $(s(x) \cdot x^Q) / g(x)$ es:




Se obtiene, pues, $c(x) = x$, y, por lo tanto, el CRC que habría que añadir sería "010".

Para saber si ha habido errores, con respecto a la recepción se repite la división: si el residuo coincide con el CRC, se entiende que la trama se ha recibido correctamente.

El grado del polinomio generador $g(x)$ no es arbitrario, sino que es determinado por el número de bits que se desea que tenga el CRC. El residuo de la división por un polinomio de grado $p$ es un polinomio de grado menor o igual a $p - 1$.

Si queremos que el CRC tenga $Q$ bits, es decir, que su representación polinomial tenga un grado menor o igual a $Q - 1$, tendremos que elegir un polinomio generador de grado $Q$. En otras palabras, el grado del generador debe ser igual al número de bits del CRC.

Las propiedades del código CRC dependen del polinomio generador. En general, sin embargo, se puede demostrar que si elegimos un polinomio generador adecuado de grado $Q$ (es decir, con un CRC de $Q$ bits):

a) La distancia de Hamming del código es mayor o igual a 4.
b) La capacidad de detección de ráfagas de error es mayor o igual a $Q$.
c) La probabilidad de que una combinación arbitraria de bits sea aceptada como palabra válida vale $2^{-Q}$.

**Polinomios generadores estandarizados**

Hay polinomios generadores muy utilizados que han sido estandarizados. Algunos de éstos, para CRC de 16 y 32 bits, son los siguientes:

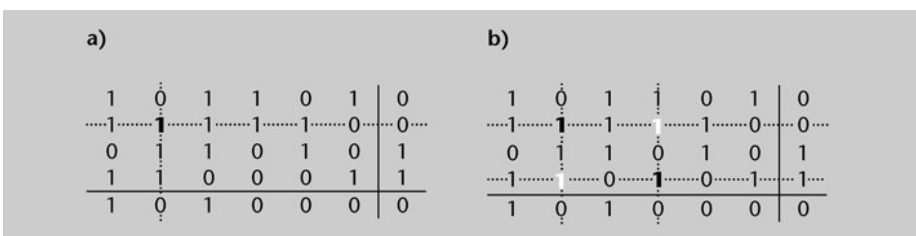
a) CRC-16: $g(x) = x^{16} + x^{15} + x^2 + 1$.

b) CRC-CCITT: $g(x) = x^{16} + x^{12} + x^5 + 1$.

c) CRC-32: $g(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1$.

## 2.7. Códigos correctores de errores

Hay códigos que, además de detectar errores, permiten corregir algunos bits erróneos. Por ejemplo, en un código de paridad longitudinal y transversal, cualquier error de un solo bit se puede corregir buscando la fila y la columna con la paridad cambiada (tal y como muestra el esquema a) de la figura siguiente). Sin embargo, si el error se produce en dos bits, el código ya no es capaz de corregir el error. Esto puede verse en el esquema b). En esta figura los bits erróneos están marcados en negro. Si el error se produjera en los bits marcados en blanco, se tendría el mismo error longitudinal y transversal. Así pues, el código de paridad longitudinal y transversal no es capaz de averiguar cuál de las dos posibilidades debería corregirse.




Corrección de bits con un código con paridad transversal y longitudinal.

**La técnica FEC**

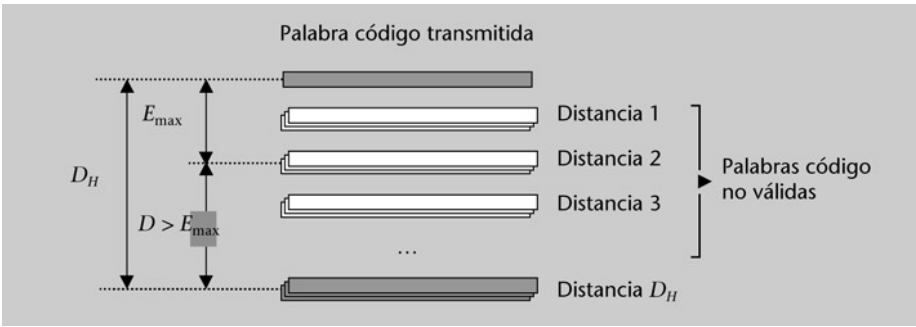
La técnica de utilizar un código corrector de errores para recuperar errores (en lugar de solicitar la retransmisión de la trama) se conoce en inglés como *forward error correction* (FEC).

Con la finalidad de hacer un análisis más formal de los códigos correctores se utiliza el concepto de *distancia de Hamming*, que ya hemos introducido. En caso de error, la corrección consiste en suponer que la palabra código transmitida es

la palabra código válida más próxima a la palabra recibida, según el concepto de *distancia* (criterio de la distancia mínima). Por lo tanto, será la que tenga menos bits de diferencia.

**Corrección de un bit**

Un código con paridad transversal y longitudinal es capaz de corregir el error en un bit (a), pero no en dos (b).




Código corrector según el criterio de las distancias mínimas.

La figura anterior es una representación gráfica de la idea que justo acabamos de exponer. En esta figura podemos ver una posible palabra código transmitida, y después encontramos agrupadas todas las palabras con el número de bits de diferencia (o sea, que distan): 1, 2… hasta las palabras válidas más próximas a $D_H$ bits de distancia. Si se recibe una de las tramas que se encuentra a una distancia 1, 2, …, $E_{máx}$, es decir, una de las palabras que no tienen otra palabra válida más próxima que la transmitida, el código corregirá el error.

Para saber cuántos bits es capaz de corregir el código con probabilidad 1, supongamos que, en la figura, $D_H$ es la distancia mínima entre dos palabras válidas (la distancia de Hamming del código).

Sea $E_{máx}$ el número de bits erróneos; de la figura deducimos que el criterio de distancia mínima corregirá el error si $E_{máx} < D$, donde $D = D_H - E_{máx}$.

En definitiva, si la distancia de Hamming de un código es $D_H$, utilizando el criterio de la distancia mínima se puede corregir cualquier combinación de $E_{máx}$ bits erróneos que cumpla $E$máx $< D_H/2$.

**Código con paridad transversal y longitudinal**

Si se utiliza un código con paridad transversal y longitudinal, la distancia de Hamming vale $D_H = 4$. Por lo tanto, se pueden corregir $E_{máx} < 2$, es decir, 1 bit, tal y como habíamos deducido al principio de este subapartado. Un código con $D_H = 5$ puede corregir *E*máx < 2,5; es decir, 2 bits.

## 3. Control de errores

En caso de que se reciba una trama con errores, el nivel de enlace puede adoptar una de las soluciones siguientes:

1) Descartar la trama.
2) Intentar corregir los errores (si se utiliza un código corrector adecuado).
3) Solicitar la retransmisión.

La solución que se adopte depende de factores virtuales al uso del enlace o al tipo de aplicación. Más adelante se discuten estos factores. Aquí sólo apuntaremos que para aplicar la tercera solución es necesario que la comunicación sea bidireccional (*half-duplex* o *full-duplex*).

La primera solución sólo es factible en aplicaciones que toleren un cierto grado de error en la información recibida. La ventaja de la segunda solución respecto de la tercera es que se evita el retardo que implica solicitar la retransmisión de las tramas. Por el contrario, los códigos correctores de errores necesitan añadir mucha redundancia (bits extras) y, por lo tanto, la transm isión es más ineficiente (la relación entre los bits de información y control disminuye). Más adelante se profundizará en el concepto de eficiencia.

Un caso típico donde es preferible un código corrector de errores antes que un código detector acompañado de retransmisiones es en comunicaciones vía satélite. Conviene más pagar en ineficiencia por el incremento de bits redundantes que en tiempo de retransmisiones, porque la distancia que hay que recorrer es muy grande.

Los motivos anteriores explican por qué, cuando se necesita hacer una recuperación de errores, se adopta la tercera solución siempre que se puede. Los pro-

tocolos que usan esta técnica se conocen como protocolos ARQ, del inglés *automatic repeat request*. En concreto, los protocolos que veremos son los siguientes:

- *Stop & Wait*
- Retransmisión selectiva
- *Go back N*

Para estudiar estos protocolos supondremos:

1) Un nivel físico genérico en el cual:
   - Las tramas pueden llegar con error o se pueden perder. Diremos que la trama se pierde cuando los errores se producen de manera tal que impiden que el receptor pueda delimitar e interpretar la recepción de la trama.
   - Las tramas llegan en el mismo orden en el que se han transmitido.
2) Un nivel por encima del nivel de enlace, en el cual:
   - El transmisor siempre tiene datos de información a punto para transmitir. El nivel de enlace, sin embargo, tiene la libertad de aceptarlos en instantes arbitrarios.
   - El receptor siempre está dispuesto a aceptar datos del nivel de enlace.

Además de describir los protocolos, los evaluaremos. La evaluación de los protocolos consiste en el cálculo de los parámetros de más interés, como la eficiencia. Para la descripción y evaluación de estos protocolos es muy útil el uso de un diagrama de tiempo. Así pues, antes de empezar con la descripción de los protocolos, explicaremos esta herramienta.

## 3.1. Diagramas de tiempo

Los diagramas de tiempo que utilizamos en este libro son una representación espaciotemporal de la transmisión de tramas entre las dos entidades de nivel de enlace. Este tipo de diagramas también se utiliza para el estudio de la comunicación entre entidades en muchos otros contextos.

Como ejemplo de este tipo de diagramas, la figura siguiente muestra la transmisión de una trama denominada $I_K$. La interpretación del índice *K* es la siguiente: si enumeramos la secuencia de tramas transmitidas con los números (1, 2, 3, …), $I_K$ representa la trama *K* de la secuencia transmitida.

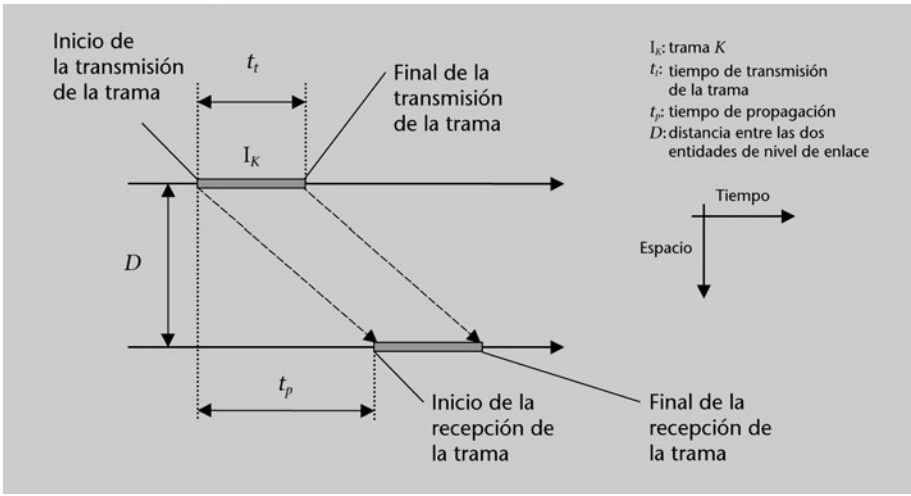



Diagrama de tiempo de la transmisión de una trama.

**Velocidad de propagación**

En el vacío, una señal electromagnética se propaga a una velocidad de $vp \cong 3 \cdot 10^8$ m/s. En cualquier otro medio la velocidad de propagación es inferior.

En la figura anterior hay dos ejes temporales; sobre cada uno se representan los acontecimientos que tienen lugar en cada una de las estaciones: en el eje superior, los que se llevan a cabo en la estación que transmite la trama, y en el inferior, los que se realizan en la estación que la recibe.

En los diagramas de tiempo que representaremos haremos estas suposiciones:

- Asumiremos que el tiempo de procesamiento es cero. Es decir, que cuando una entidad de nivel de enlace tiene una trama lista para transmitir, el nivel físico la empieza a transmitir inmediatamente.
- Supondremos que la velocidad de transmisión de la línea es fija y la designaremos por *vt* (medido en bits por segundo o bps).
- Supondremos (si no se indica lo contrario) que el tamaño de las tramas, *L*, es fijo y se mide en bits.
- Supondremos que la distancia entre las estaciones, *D*, y la velocidad de propagación en el medio, $v_p$, son fijas.

En función de $v_t$ y $L$, el tiempo de transmisión de una trama ($t_t$) viene dado por:

Ecuación 1

$$t_t\,(\mathrm{s}) = \frac{L(\mathrm{bit})}{v_t\,(\mathrm{bps})}$$

En función de $D$ y $v_p$, el tiempo de propagación ($t_p$) se puede obtener a partir de:

Ecuación 2

$$t_p\,(\mathrm{s}) = \frac{D(\mathrm{m})}{v_p\,(\mathrm{m/s})}$$

De ahora en adelante utilizaremos una versión simplificada como el diagrama de tiempo de la figura anterior, en la que indicaremos sólo el tiempo de transmisión de la trama transmitida, el tiempo de propagación y el instante de llegada del último bit de la trama, tal y como se muestra en la figura siguiente:

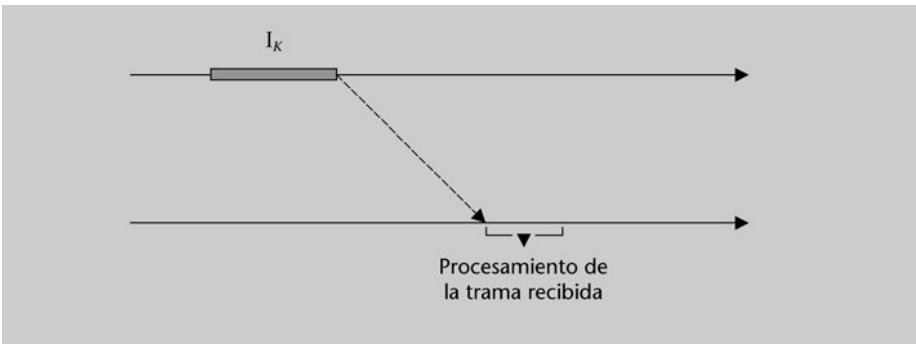



Diagrama de tiempo simplificado.

Es preciso observar que el receptor sólo puede empezar a procesar la trama recibida (comprobar si hay errores) después de haber recibido el último bit.

### 3.2. Evaluación de los protocolos: eficiencia

El principal criterio de evaluación de los protocolos ARQ es la eficiencia:

La eficiencia cuantifica la pérdida de capacidad de transmisión de información. Es decir, se trata de la relación entre la velocidad media a la que se transmiten los bits de información y la velocidad máxima a la que se podrían transmitir.

La velocidad de transmisión máxima ($v_t$) es la velocidad de transmisión de la línea; es decir, la velocidad a la cual el nivel físico es capaz de transmitir los bits:

Ecuación 3
$$v_t = \frac{1}{\text{tiempo de transmisión de un bit}}$$

La velocidad que nos interesa, sin embargo, es la velocidad media a la que se puede transmitir la información, teniendo en cuenta todos los factores que reduzcan la velocidad de transmisión máxima. Esta velocidad la denominaremos velocidad de transmisión eficaz o velocidad de transmisión efectiva ($v_{ef}$):

Ecuación 4
$$v_{ef} = \frac{\text{bits de información}}{\text{tiempo de transmisión}}$$

La eficiencia se define como la relación entre la velocidad efectiva y la velocidad de transmisión; es decir:

Ecuación 5
$$E = \frac{v_{ef}}{v_t} = \frac{\text{bits de información}}{\text{tiempo de transmisión}} / \frac{1}{\text{tiempo de transmisión de un bit}}$$

Dado que:

bits de información × tiempo de transmisión de un bit =
= tiempo que se han transmitido los bits de información,

la ecuación 5 para la eficiencia se puede expresar también como se muestra a continuación:

Ecuación 6 $$E = \frac{\text{tiempo transmitiendo información}}{\text{tiempo de transmisión}}$$

A veces es interesante tener la eficiencia en función de los bits transmitidos, en lugar del tiempo invertido en la transmisión de éstos. Teniendo en cuenta la igualdad siguiente:

$$\frac{\text{tiempo de transmisión}}{\text{tiempo de transmisión de un bit}} = \text{Número de bits que se podrían haber transmitido en el tiempo de transmisión.}$$

La ecuación 5 también se puede expresar de esta manera:

Ecuación 7 $$E = \frac{\text{bits de información}}{\text{bits que caben en el tiempo de transmisión}}$$

### 3.3. Protocolo *Stop & Wait*

El protocolo *Stop* & *Wait*, que podemos traducir por 'para y espera', es el protocolo ARQ más sencillo.

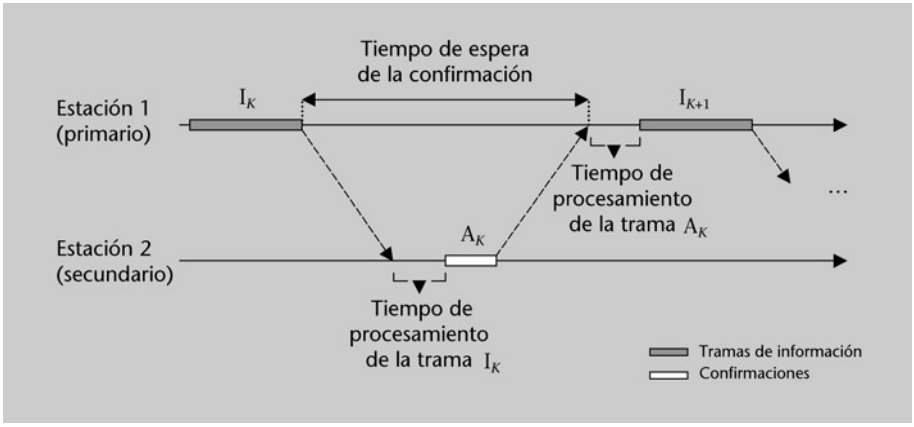
El principio de funcionamiento del protocolo *Stop* & *Wait* es "no transmitir una trama nueva hasta que no se tiene la certeza de la recepción correcta de la trama anterior".

Para conseguirlo se definen dos tipos de tramas:

a) Las tramas de información o tramas, que llevan la información que se intercambian las entidades del nivel superior.
b) Las tramas de confirmación o confirmaciones, que forman parte del protocolo del nivel de enlace y que no llevan información de los niveles superiores.

La figura siguiente muestra el funcionamiento de este protocolo. En esta figura la estación 1 envía tramas de información a la estación 2, y la estación 2 las

confirma. Para distinguir la estación transmisora de tramas de información de la estación transmisora de las confirmaciones, nos referiremos a las mismas como primario y secundario, respectivamente.




Protocolo *Stop & Walt*.

El diagrama de la figura debe interpretarse de la manera siguiente: cuando el secundario recibe completamente la trama $I_K$, comprueba si la ha recibido correctamente y, en caso afirmativo, envía la confirmación $A_K$.

Observad que tanto el secundario como el primario invierten un cierto tiempo en el procesamiento de las tramas recibidas. Esto implica que, aunque el tiempo de propagación es constante, a causa del tiempo de procesamiento de las tramas en el secundario, el tiempo de espera de las confirmaciones es variable.

Para simplificarlo, supondremos generalmente que el tiempo de procesamiento de las tramas es cero, excepto cuando sea necesario tenerlo en cuenta.

El tamaño de las tramas de confirmación es normalmente mucho menor que el de las tramas de información, dado que no llevan datos del nivel superior. Además, hay que tener en cuenta que el código detector de errores también tiene que aplicarse a las confirmaciones para asegurar la recepción correcta.

En este subapartado supondremos que el código detector de errores es lo bastante robusto como para poder detectar cualquier error que pueda haber en la transmisión de una trama.

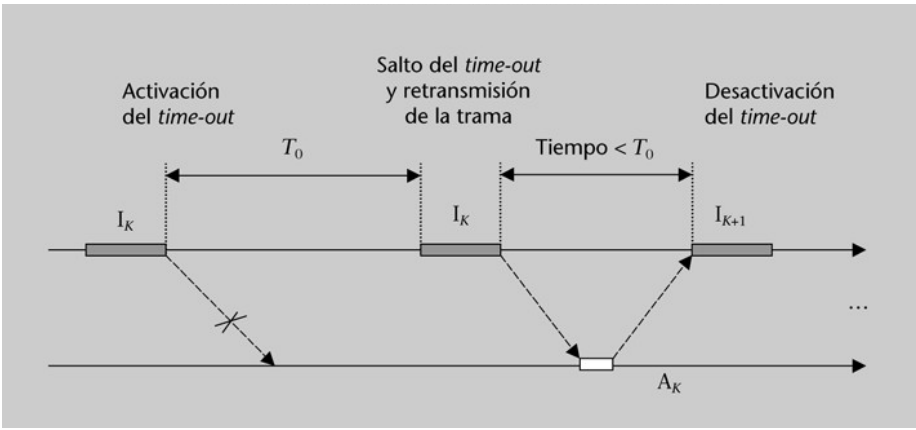
En caso de que haya error, el secundario solicita la retransmisión de la trama de información. Esto se puede hacer mediante las retransmisiones implícitas y las explícitas.

### *3.3.1. Protocolo* Stop & Wait *con retransmisiones implícitas*

En este caso las reglas que siguen el primario y el secundario son las siguientes:

1) El secundario envía confirmaciones positivas, A, de las tramas que recibe sin errores.
2) El primario, después de enviar una trama $I_K$, espera un tiempo $T_0$ (activa un temporizador o *time-out*) para recibir la confirmación positiva, $A_K$.
   a) Si agotado el tiempo $T_0$ la confirmación no ha llegado, retransmite la trama $I_K$ y vuelve a activar el *time-out*.
   b) Si recibe la confirmación positiva, $A_K$, desactiva el *time-out*, acepta nuevos datos del nivel superior, monta una nueva trama $I_K$+1 y repite este proceso.

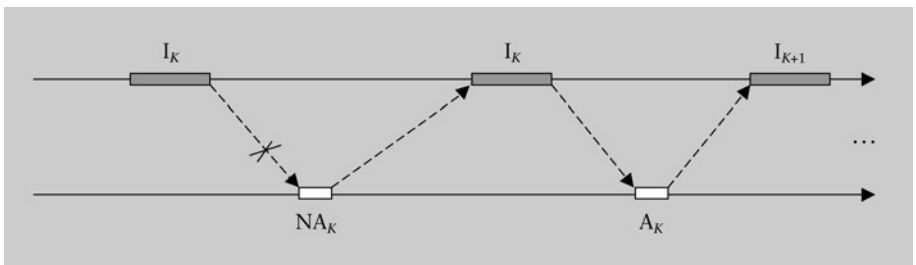
La figura siguiente muestra la recepción de una trama errónea y su retransmisión implícita. Evidentemente, el valor $T_0$ del *time-out* debe fijarse de manera que el primario tenga tiempo de recibir la confirmación (es decir, $T_0 > ta + 2tp$, donde *ta* es el tiempo de transmisión de las confirmaciones y *tp* es el tiempo de propagación). Si el valor del *time-out* fuera demasiado pequeño, se produciría la retransmisión innecesaria de las tramas.




Retransmisión implícita de una trama errónea.

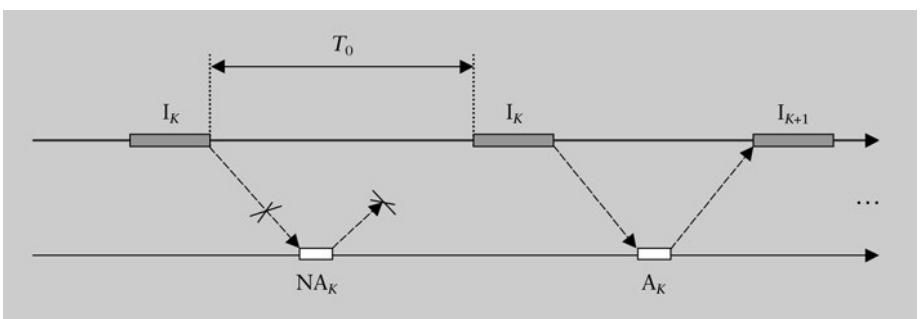
### *3.3.2. Protocolo* Stop & Wait *con retransmisiones explícitas*

Las reglas en este caso son las mismas que en la retransmisión implícita, con la diferencia de que el secundario también envía confirmaciones negativas ($NA_K$) en el caso de que reciba una trama con errores:




Retransmisión explícita de una trama errónea.

En general, la confirmación negativa permite una retransmisión más rápida de la trama errónea porque no hay que esperar a que se agote el *time-out*. Sin embargo, en la transmisión también es necesario utilizar un *time-out*, ya que las confirmaciones pueden llegar con errores o se pueden perder[2]. Esta condición se refleja en la figura siguiente:




Necesidad del *time-out* en una representación explicita.

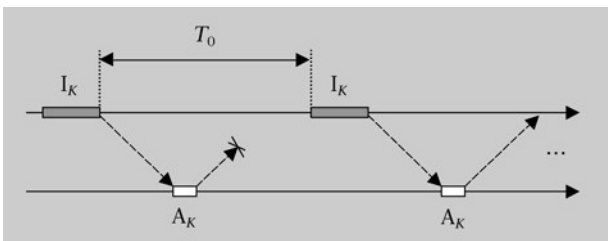
2. Las confirmaciones pueden perderse en caso de que los errores no permitan reconocer la recepción de la trama.

### *3.3.3. Necesidad de numerar las tramas y las confirmaciones*

Sabemos que las tramas llevan un campo de control que sirve para detectar posibles errores. En este subapartado veremos que, para el funcionamiento correcto del protocolo, es necesario numerar las tramas de información y las confirmaciones (positivas o negativas). Así pues, otro campo de control de las tramas es un número de secuencia que permite identificarlas.

**Necesidad de numerar las tramas de información**

La figura siguiente evidencia la necesidad de numerar las tramas de información:



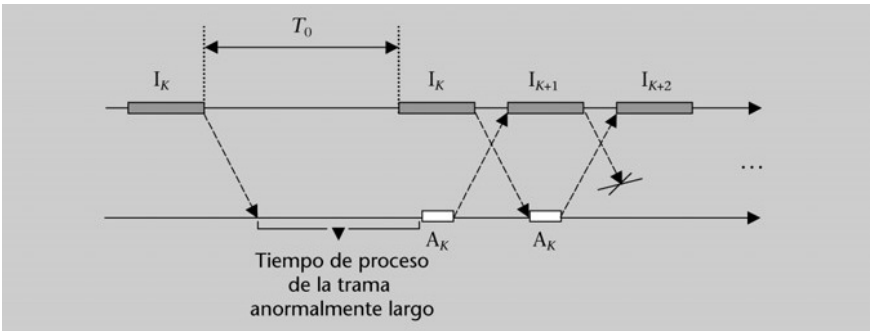

Necesidad de numerar las tramas de información.

En este caso, se pierde la confirmación de una trama. Dado que el primario no recibe la confirmación, salta el *time-out* y retransmite la misma trama $I_K$. Si la trama no estuviera numerada, el secundario podría detectar la recepción duplicada de la trama. Gracias al número de secuencia el secundario detecta esta condición, descarta la trama duplicada y vuelve a enviar la confirmación para que el primario continúe la transmisión de la trama de información siguiente (la trama $I_{K+1}$).

Hasta ahora, en los diagramas de tiempo hemos considerado que el tiempo de procesamiento de las tramas era igual a cero. Ahora tendremos en cuenta que, en una situación real, este tiempo puede ser variable. Incluso puede tener valores anormalmente grandes. Este hecho puede responder a múltiples causas.

**Necesidad de numerar las confirmaciones**

La figura siguiente es un ejemplo que evidencia la necesidad de numerar las confirmaciones:

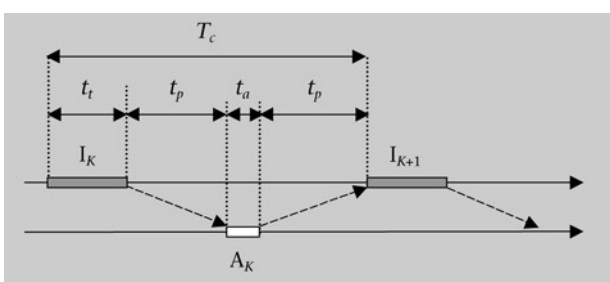



Necesidad de numerar las confirmaciones.

En el caso que presentamos, la confirmación $A_K$ llega al cabo de un tiempo anormalmente largo (a causa de un tiempo de procesamiento excesivo en el secundario), y provoca que antes de que llegue la confirmación salte el *time-out* y la trama $I_K$ se retransmita. Por este motivo, llegan dos confirmaciones de la trama $I_K$. Si éstas no se numeraran, la segunda confirmación se interpretaría como la confirmación de la trama $I_K$+1. En caso de que la trama $I_{K+1}$ se perdiera (como indica la figura), el primario no la retransmitiría.

### *3.3.4. Cálculo de la eficiencia del protocolo Stop & Wait*

En el caso del protocolo *Stop & Wait* la pérdida de capacidad de transmisión se debe al tiempo que se pierde en la espera de confirmaciones y retransmisiones de tramas. Hay también otros factores que afectan a la eficiencia, como los campos de control en las tramas de información. A la hora de calcular la eficiencia de un protocolo siempre debemos considerar todos los datos que tengamos al alcance y que resulten de una reducción de la capacidad de transmisión. Con el objetivo de calcular la eficiencia consideraremos el diagrama de tiempo siguiente:




Evaluación del protocolo *Stop & Walt*.

Recordemos que $t_p$ es el tiempo de propagación. Supondremos que todas las tramas tienen el mismo tamaño, igual a $L$ bits (si el tamaño fuera variable, tomaríamos como $L$ su valor medio). El tiempo necesario para la transmisión de la trama lo da la ecuación 1. De la misma manera, si las confirmaciones son de $M$ bits, el tiempo de transmisión de una confirmación ($t_a$) es:

Ecuación 8

$$t_a\,(\text{s}) = \frac{M(\text{bit})}{v_t(\text{bps})}$$

Según esto, el ciclo indicado como $T_c$ en la figura se repetirá en la transmisión de cada trama. Si se aplica la ecuación 6, la eficiencia del protocolo vendrá dada por la relación entre el tiempo de transmisión de la trama ($t_t$) y el tiempo invertido para la transmisión de la trama ($T_c$). De la figura anterior deducimos que $T_c$ es $T_c = t_t + t_p + ta + t_p$; así:

Ecuación 9

$$E = \frac{t_t}{T_c} = \frac{t_t}{t_t + t_a + 2t_p}$$

A menudo, el tamaño de las tramas de información es mucho mayor que el de las confirmaciones. Entonces $t_t >> t_a$ y, por lo tanto, la ecuación 9 se puede reescribir de la manera siguiente:

Ecuación 10

$$E \cong \frac{t_t}{t_t + 2t_p} = \frac{1}{1 + 2t_p/t_t} = \frac{1}{1 + 2a}$$

En función de la eficiencia podemos calcular la velocidad efectiva ($v_{ef}$):

Ecuación 11

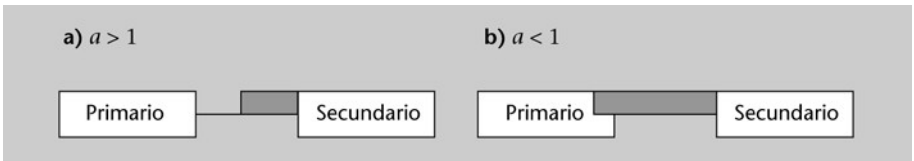
$$v_{ef} = v_t \cdot E = \frac{v_t}{1 + 2a}$$

El parámetro $a$, según la medida de la trama, tiene la interpretación física siguiente:

$$a = \frac{t_p}{t_t} = \frac{t_p}{L/v_t} = \frac{t_p \cdot v_t}{L} = \frac{\text{número de bits que caben en el enlace}}{\text{tamaño de la trama}}$$

Es decir, podemos interpretar *a* como el número de tramas que caben en el enlace. Por lo tanto, se da uno de estos casos:

- Si $a > 1$, cuando el primer bit de la trama llega al secundario, el primario ya ha acabado de transmitir la trama.
- Si $a < 1$, cuando el primer bit de la trama llega al secundario, el primario todavía no ha acabado de transmitir la trama.




Interpretación del parámetro *a*.

Hay que tener en cuenta las observaciones siguientes:

a) La eficiencia normalmente se da en tanto por ciento, es decir, multiplicando el resultado de la ecuación 9 por 100.
b) En el caso de que haya errores, la eficiencia es inferior porque hay que considerar las retransmisiones. Para evitar confusiones hablaremos de la eficiencia máxima para referirnos a la eficiencia que podemos conseguir en el mejor de los casos, es decir, sin errores.
c) Tal y como se ha comentado antes, deben tenerse en cuenta todos los datos que afecten a la eficiencia y que tengamos al alcance, por ejemplo los campos de control de las tramas de información. Supongamos que sabemos que de los $L$ bits de las tramas de información hay $L_d$ de información y $L_c$ de control.

Aplicando la ecuación 7, la eficiencia debida a los bits de control de la trama (llamémosla $E_t$) se obtendría a partir de la ecuación siguiente:

Ecuación 12

$$E_t = \frac{L_d}{L}$$

Evidentemente, la eficiencia total (*Etot*) es el producto de las eficiencias parciales:

Ecuación 13

$$E_{tot} = E_p \cdot Et$$

## 3.4. Protocolos de transmisión continua

Acabamos de ver que el *Stop & Wait* puede llegar a ser muy ineficiente, a causa de que el primario no puede transmitir durante el tiempo de espera de las confirmaciones.

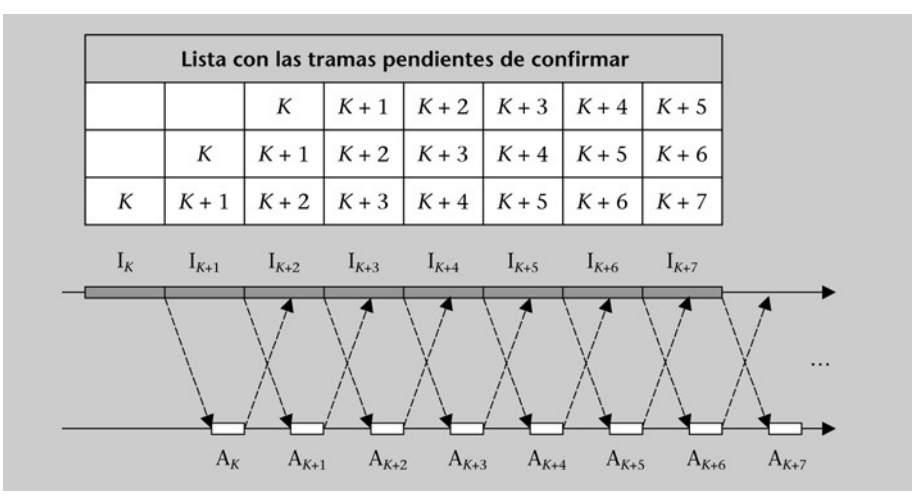
En los protocolos de transmisión continua se deja que el primario transmita continuamente tramas de información mientras se esperan las confirmaciones.

Evidentemente, esto sólo es posible si el enlace es *full-duplex*, porque el primario y el secundario tienen que poder ocupar simultáneamente el canal. En el caso del protocolo *Stop & Wait* esto no es necesario porque el primario y el secundario ocupan el canal alternativamente.

En caso de error, el primario se puede ver obligado a retransmitir una trama anterior a la última transmitida.

Así pues, en un protocolo de transmisión continua es necesario que el primario guarde en una lista las tramas que se han enviado y que están pendientes de confirmar, para poder retransmitirlas en caso de error. Esta lista se denomina lista de transmisión.

La figura siguiente muestra el principio de funcionamiento de un protocolo de transmisión continua. Cada trama transmitida se guarda en una lista y se




Protocolo de transmisión continua.

extrae cuando llega su confirmación. Se puede observar que el primario no se queda nunca parado; es decir, en ausencia de errores la eficiencia de un protocolo de transmisión continua es del 100%.

En caso de error se utiliza una de las dos estrategias que se explican a continuación: la retransmisión selectiva y el protocolo *Go back N*.

### *3.4.1. Retransmisión selectiva*

En la estrategia de retransmisión selectiva el secundario sólo solicita la retransmisión de las tramas erróneas. En función del convenio que se utiliza para solicitar la retransmisión, hay dos versiones de este protocolo: la implícita y la explícita (análogas a las que hay en el protocolo *Stop* & *Wait*).

*Retransmisión selectiva implícita*

Las reglas que debe cumplir una retransmisión selectiva implícita son las siguientes:

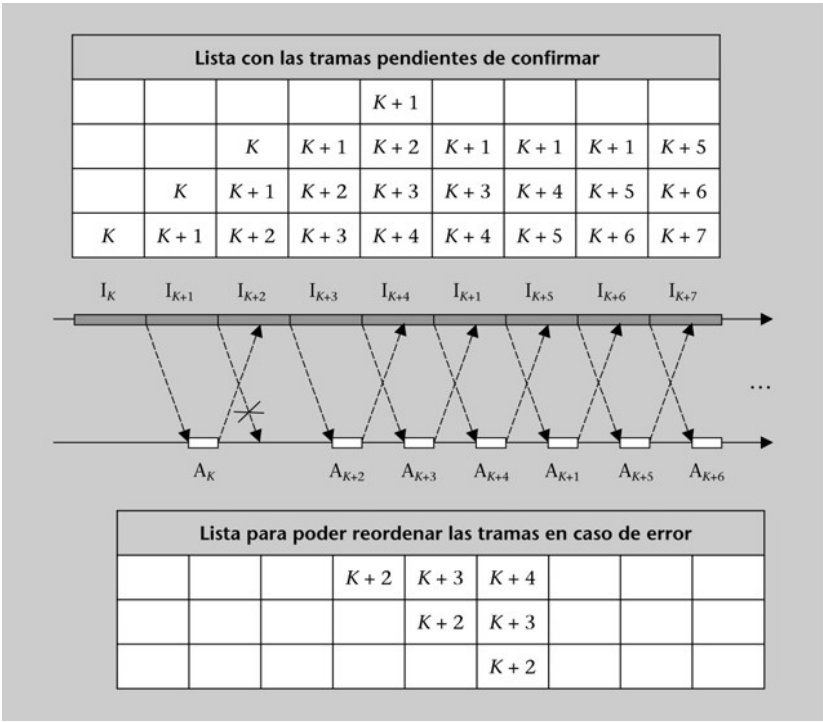
a) El secundario sólo envía confirmaciones positivas.
b) El secundario envía una confirmación por cada trama de información recibida.
c) Si el primario recibe dos confirmaciones no consecutivas, deduce que la confirmación que falta corresponde a una trama errónea y la retransmite.
d) Como en el protocolo *Stop* & *Wait*, el primario tiene que activar un *time-out* para la retransmisión de las tramas pendientes de confirmar. Esto es así porque una trama puede llegar con errores en sucesivas retransmisiones, pero el primario, con la no secuencia de las confirmaciones, sólo detectará el primer error.

Tal y como ya se ha comentado, el nivel de enlace debe entregar los datos de información que llevan las tramas no tan sólo sin errores, sino también en orden.

Si en una retransmisión selectiva se produce un error, el secundario tiene que memorizar las tramas que llegan después de la trama con errores en una lista para poder entregarlas en orden. Ésta se conoce como la lista de recepción.

En la figura siguiente se muestra una condición de error en una retransmisión selectiva implícita y la lista de recepción que es necesario que el secundario elabore para poder reordenar las tramas:

En la lista de recepción también podríamos considerar la posición ocupada por la trama en fase de recepción.




Condición de error en una retransmisión selectiva implícita.

- La trama $I_{K+1}$ llega con errores y, por lo tanto, el secundario no envía la confirmación correspondiente.
- Cuando la confirmación $A_{K+2}$ llega al primario, éste se da cuenta del error en la trama $I_{K+1}$ porque no ha recibido la confirmación, y retransmite la copia que guarda en la lista de tramas no confirmadas.

*Retransmisión selectiva explícita*

Las reglas que debe cumplir una retransmisión selectiva explícita son las siguientes:

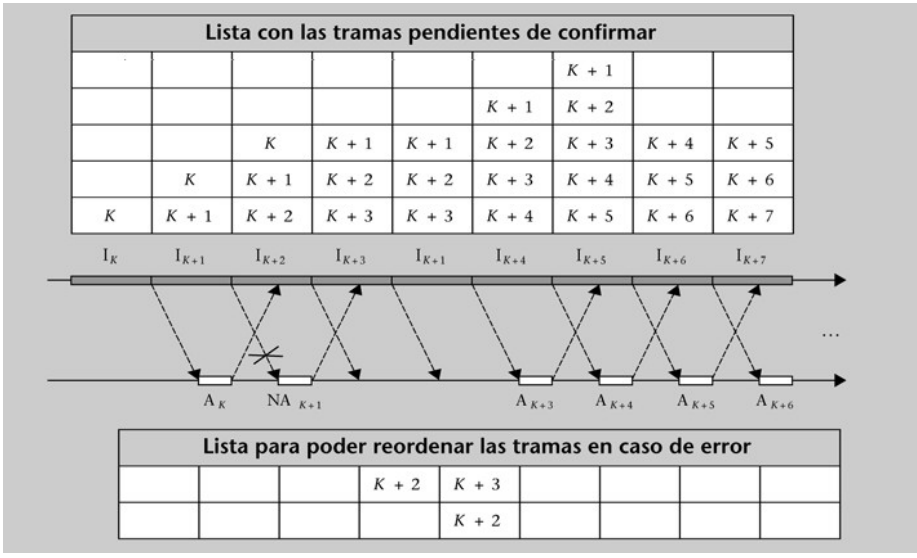
a) El secundario envía confirmaciones positivas cuando la trama es correcta y negativas cuando recibe una trama con errores o fuera de secuencia.

b) Si el primario recibe una confirmación negativa $NA_K$, debe retransmitir la trama $I_K$ e interpreta que todas las tramas anteriores a $I_K$ se han recibido correctamente: una confirmación $A_K$ confirma la trama $I_K$ y todas las anteriores.

   Una consecuencia de esta regla es que, mientras el secundario espera la retransmisión de una trama K, deja de enviar confirmaciones positivas o negativas de tramas posteriores a la K.

c) El secundario tiene que activar un *time-out* para la retransmisión de las confirmaciones negativas de las tramas no recibidas, ya que una confirmación negativa $NA_K$ podría perderse o llegar con errores, y el primario no retransmitiría la trama $I_K$.

La figura siguiente muestra el funcionamiento de la retransmisión selectiva explícita:




Condición de error en una retransmisión selectiva implícita.

- La trama $I_{K+1}$ llega con errores y, por lo tanto, el secundario envía la correspondiente confirmación negativa $NA_{K+1}$ y deja de transmitir confirmaciones.
- Cuando $NA_{K+1}$ llega al primario, éste retransmite la trama $I_{K+1}$.
- Cuando $I_{K+1}$ llega al secundario, éste envía $A_{K+3}$ y confirma todas las tramas recibidas correctamente hasta entonces.

### *3.4.2. Protocolo* Go back N

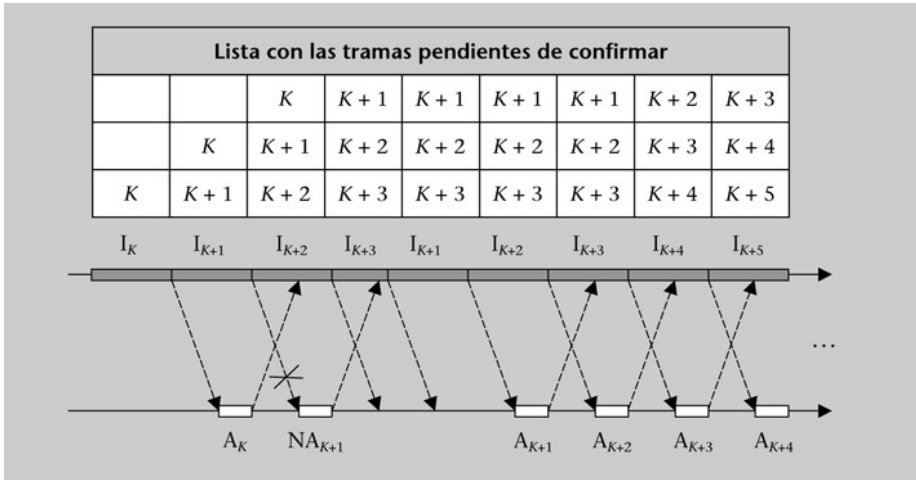
En muchos enlaces la probabilidad de error en una trama es relativamente baja, de manera que la retransmisión de tramas es un acontecimiento poco frecuente. Aprovechando esto, el protocolo *Go back N* simplifica el mecanismo de recepción de las tramas y elimina así el problema de la reordenación.

Para eliminar el problema de la reordenación, en caso de error el primario retransmite todas las tramas a partir de la trama errónea. De este modo, cuando el secundario detecta un error, lo comunica al primario y descarta todas las tramas que le llegan hasta la nueva recepción de la trama errónea, que irá seguida de las tramas descartadas.

El protocolo *Go back N*, en la práctica, sólo existe en la versión explícita. De esta forma, en caso de error el secundario descarta las tramas hasta que el primario retransmite la trama errónea y, por lo tanto, no tendría sentido enviar confirmaciones de las tramas descartadas. Así pues, las reglas del protocolo *Go back N* son las siguientes:

a) Cuando el secundario recibe una trama con errores (o fuera de secuencia), solicita la retransmisión de trama errónea (o de la trama que falta) enviando una confirmación negativa.
b) El primario siempre transmite las tramas en secuencia, de manera que si recibe una confirmación negativa $NA_K$, retransmite la trama $I_K$ y todas las posteriores, aunque ya las hubiera transmitido.
c) Una confirmación positiva A*K* confirma la trama $I_K$ y todas las anteriores. Así pues, la pérdida de una confirmación no tiene ningún efecto porque la confirmación siguiente ratifica la perdida. Una confirmación negativa de la trama *K* también ratifica todas las tramas anteriores a *K*.
d) El secundario activa un *time-out* para la retransmisión de las confirmaciones negativas de las tramas no recibidas.

La figura siguiente muestra el funcionamiento del protocolo *Go back N*:




Condición de error en un protocolo *Go back N*.

a) La trama $I_{K+1}$ llega con errores y, por lo tanto, el secundario envía la correspondiente confirmación negativa $NA_{K+1}$.
b) Cuando $NA_{K+1}$. llega al primario, éste aborta la transmisión en curso de la trama $I_{K+3}$ y empieza a retransmitir inmediatamente la trama $I_{K+1}$.

Dado que el secundario no tiene que reordenar las tramas, no necesita una lista para la reordenación.

## 4. Control de flujo

El control de flujo consiste en regular la velocidad de transmisión de las tramas. La necesidad del control de flujo se explica por los motivos siguientes:

1) Puede ocurrir que el mismo nivel de enlace no sea capaz de procesar las tramas a la velocidad con la que llegan.
2) En los protocolos de transmisión continua, el primario necesita una memoria donde almacenar la lista de tramas pendientes de confirmar. Si no impo-

nemos ninguna limitación a la velocidad de transmisión de las tramas, no seremos capaces de dimensionar la memoria necesaria.

3) Hasta ahora hemos supuesto que el nivel superior siempre está dispuesto a aceptar las tramas del nivel de enlace. En la práctica, sin embargo, es posible que esto no sea así. En este caso se podría agotar la memoria en la que el secundario almacena las tramas recibidas.

Típicamente, en el nivel de enlace el algoritmo de control de flujo está vinculado al algoritmo de control de errores. Por ejemplo, el protocolo *Stop & Wait* sirve sin modificaciones para hacer un control de flujo. Simplemente, el secundario tiene que retrasar las confirmaciones para conseguir disminuir la velocidad de transmisión del primario.

**Control de flujo X-ON/X-OFF**

En algunas transmisiones orientadas a carácter (donde la unidad de transmisión de nivel físico es el carácter), se utiliza el mecanismo de control de flujo denominado X-ON/X-OFF. En este mecanismo hay un carácter especial (X-OFF) que se envía para detener el equipo remoto. Cuando uno de los equipos recibe un carácter X-OFF, se para hasta recibir el carácter de activación X-ON.

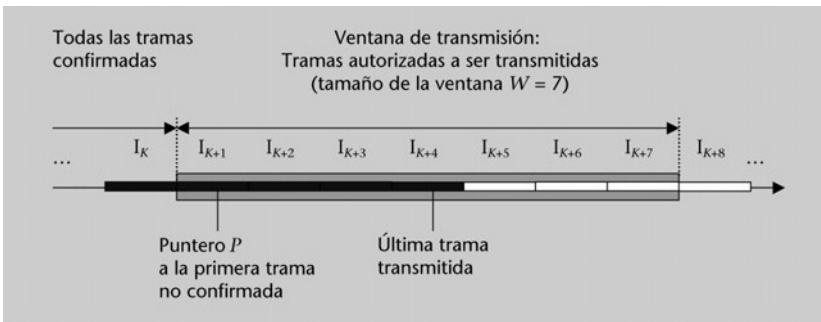
## 4.1. Protocolos de ventana

Los algoritmos de transmisión continua no tienen que esperar las confirmaciones para la transmisión de nuevas tramas. De esta forma, dejar de enviar confirmaciones no permite hacer un control de flujo. Para hacerlo, se limita el número de tramas que se pueden transmitir sin confirmar. Los protocolos resultantes se conocen como *protocolos de ventana*, porque esta limitación se puede interpretar como una ventana que se desplaza sobre el flujo de tramas del primario a medida que llegan las confirmaciones.

Los protocolos de ventana son protocolos de transmisión continua en los que el primario puede transmitir hasta $W$ tramas después de la trama $K$, una vez están confirmadas la trama $K$ *y* todas las anteriores. El parámetro $W$ se denomina tamaño de la ventana.

El transmisor mantiene un puntero *P* a la primera trama no confirmada (todas las tramas con identificador menor que *P* están confirmadas). Este puntero se actualiza a medida que llegan las confirmaciones.

El primario está autorizado a transmitir tramas con identificadores < *P* + *W*. Las tramas *P*, *P* + 1, …, *P* + *W* – 1 se conocen como ventana de transmisión. Por ejemplo, si en las condiciones de la figura siguiente no llegan más confirmaciones, después de la transmisión de la trama $I_{K+7}$ el primario se quedará parado:




Transmisión de tramas en un protocolo de ventana.

Para hacer el control de flujo en un protocolo de ventana, el secundario simplemente deja de enviar confirmaciones. En este caso, el primario agotará la ventana y se parará.

La ventana mínima que se puede definir es *W* = 1. En este caso, el primario no puede transmitir una nueva trama hasta que se confirme la anterior; es decir, un protocolo de ventana con una ventana de tamaño *W* = 1 es equivalente a un protocolo *Stop* & *Wait*.

## 4.2. Dimensionado de la lista de transmisión

Una propiedad importante de la ventana de transmisión es que permite dimensionar la memoria necesaria para almacenar la lista de tramas pendientes de confirmar del primario.

El razonamiento que seguiremos para llegar a esta conclusión es el siguiente: si todas las tramas con identificador menor que *P* (donde *P* es el puntero a la pri-

mera trama no confirmada) han sido confirmadas, podemos asegurar que el secundario no solicitará la retransmisión de ninguna de éstas y, en consecuencia, no hay que guardarlas en la lista. Dado que el primario no puede haber transmitido ninguna trama con identificador mayor o igual que $P + W$, y que en la lista sólo ha guardado las tramas transmitidas, deducimos que en la lista de tramas no confirmadas del primario de un protocolo de ventana puede haber como máximo $W$ tramas.

### 4.3. Ventana de recepción

Por analogía con la ventana de transmisión, la ventana de recepción se define como el número máximo de tramas que debe almacenar el secundario.

Cuando se define la ventana de recepción se asume que el nivel de enlace saca de la lista de recepción todas las tramas que no están pendientes de ser recibidas con un identificador anterior. De esta manera, se garantiza que los datos que llevan estas tramas se pasan sin errores y de manera secuenciada al nivel superior. Además, en la ventana de recepción se incluye la trama que está en fase de recepción.

Así pues, la ventana mínima en recepción es igual a 1 (para almacenar la trama en fase de recepción).

También deben almacenarse las tramas que se pueden tener que reordenar. Puesto que en los protocolos *Stop* & *Wait* y *Go back N* no existe el problema de la reordenación de las tramas, tienen una ventana de recepción que vale 1.

El problema de la reordenación sólo surge en el caso de la retransmisión selectiva. Para conocer cuál es el número máximo de tramas que se pueden tener que reordenar en el secundario, basaremos el razonamiento en la figura anterior. Dado que todas las tramas con identificadores menores o iguales que $K$ han sido confirmadas, tienen que haber sido recibidas sin errores por el secundario y, por lo tanto, éste las debe haber sacado de la lista de recepción. Así pues, resulta que las únicas tramas que puede tener que ordenar el secundario son las $K + 1 \ldots K + W$ tramas que el primario ha sido autorizado a transmitir (en total $W$ tramas).

Por lo tanto, en una retransmisión selectiva el tamaño de la ventana de recepción es igual que el tamaño de la ventana de transmisión.

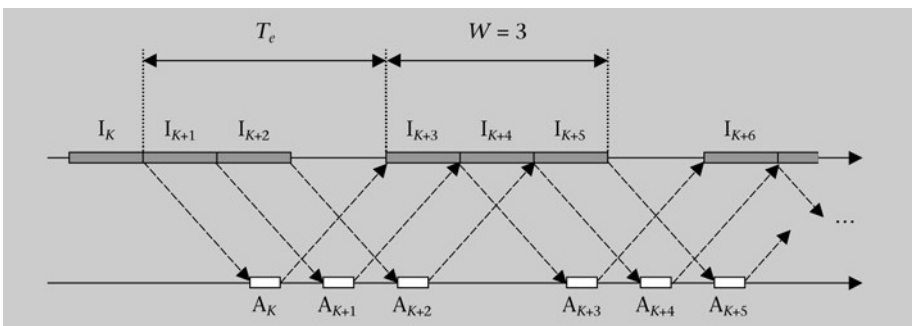
La tabla siguiente resume el tamaño de las ventanas en los tres protocolos:

| Protocolo | Ventana de transmisión | Ventana de recepción |
|---|---|---|
| *Stop & Wait* | 1 | 1 |
| *Go back N* | *W* | 1 |
| Retransmisión selectiva | *W* | *W* |

Tamaño de las ventanas de transmisión y recepción.

## 4.4. Ventana óptima

Hemos visto que la eficiencia de los protocolos de transmisión continua es del 100%, gracias al hecho de que el primario no permanece nunca parado esperando confirmaciones. En un protocolo de ventana, esta condición puede no darse; es decir, el transmisor se puede quedar parado si el tamaño de la ventana no es lo bastante grande.




Parada del primario en caso de ventana pequeña.

**Ventana de transmisión pequeña**

En la figura, la medida de la ventana es $W = 3$ (se pueden transmitir tres tramas sin confirmar). Es decir, después de transmitir una trama, el primario está autorizado a transmitir dos más antes de recibir la primera confirmación. En la figura se ve, sin embargo, que el tiempo de espera de la confirmación ($T_e$) es mayor que el tiempo de transmisión de las dos tramas y el primario se queda parado.

La ventana óptima ($W_{opt}$) es la ventana mínima que permite conseguir una eficiencia del protocolo del 100%.

En el ejemplo de la figura anterior hemos calculado que la ventana óptima (*Wopt*) vale lo siguiente:

$$W_{opt} = 1 + [T_e/t_t] = 1 + [(2t_p + t_a)/t_t],$$

donde $[x]$ es el entero mayor o igual que $x$.

## 4.5. Dimensionamiento del campo de los identificadores de trama

Hasta ahora hemos supuesto que las tramas van numeradas para que se puedan identificar, sin preocuparnos de las limitaciones de esta secuencia de números. En la trama habrá un campo, supongamos que de $n$ bits, destinado al número de secuencia. Dado que con $n$ bits podemos generar $2^n$ identificadores distintos, si numeramos de forma secuencial las tramas se repetirán cíclicamente los mismos identificadores cada $2^n$ tramas.

De esta forma, existe el peligro de que, al solicitar una retransmisión $A_K$, haya ambigüedad: se puede tratar de la trama $I_K$ actual o de la del ciclo anterior. Igualmente, es necesario evitar que el secundario interprete como tramas duplicadas las que llegan con el mismo identificador, pero que son de ciclos distintos.

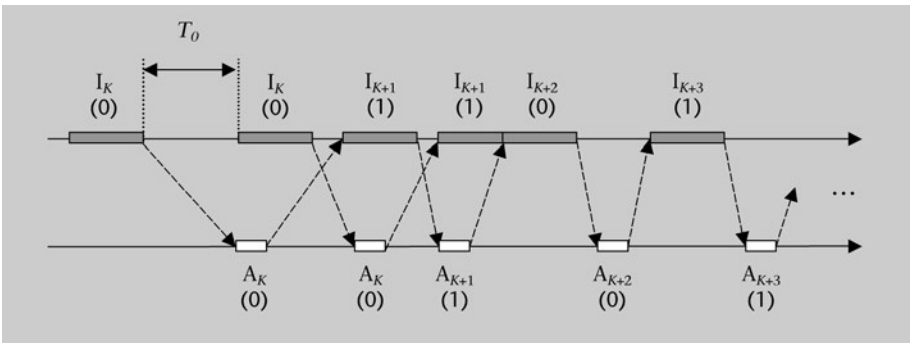
Para saber cuántos identificadores necesitamos como mínimo para evitar ambigüedades haremos una deducción informal para el caso del protocolo *Stop & Wait*. Supondremos las mismas condiciones que las que ya hemos enumerado antes: las tramas pueden llegar con retardos arbitrarios, pero llegan en el orden en el que se han transmitido.

Dado que en este protocolo no se rompe nunca la secuencia de los identificadores en la transmisión de las tramas de información, éstas siempre llegarán en orden. De este hecho se deduce que las confirmaciones también llegarán siempre en orden.

Supongamos que el primario ha recibido la confirmación de la trama $K$, pero no la de la $K + 1$. Puesto que la ventana vale 1 y las tramas de información llegan en orden, al secundario sólo pueden llegar tramas $I_K$ e $I_{K+1}$. Y dado que las con-

firmaciones llegan en orden, al primario no le puede llegar ninguna confirmación anterior a la *K*. Por consiguiente, sólo le pueden llegar las confirmaciones $A_K$ y $A_{K+1}$. Deducimos, pues, que en el caso del protocolo *Stop* & *Wait* únicamente puede haber ambigüedad entre dos tramas y, por lo tanto, sólo son necesarios dos identificadores distintos.

En la figura siguiente hay un ejemplo del funcionamiento del protocolo *Stop* & *Wait* con dos identificadores (entre paréntesis en la figura). En este caso, la confirmación de la trama I*K* tarda un tiempo anormalmente largo y se retransmite (salta el *time-out* $T_0$). Eso provoca que la trama $I_K$ y la confirmación $A_K$ lleguen duplicadas con el identificador[3] (0). Tanto el primario como el secundario detectan la duplicación porque la trama siguiente tiene el identificador (1). Cuando llega la confirmación de la trama $I_{K+1}$, el primario aborta la segunda transmisión de esta trama y empieza la transmisión de la siguiente ($I_{K+2}$).




Funcionamiento del protocolo *Stop & Walk* con dos identificadores.

La deducción del número de identificadores necesarios en el caso de los protocolos *Go back N* y retransmisión selectiva es más compleja y no la haremos. El resultado es que si el tamaño de la ventana es *W*, se necesitan *W* + 1 identificadores distintos en el protocolo *Go back N* y 2*W* en el de retransmisión selectiva.

El número de identificadores necesario, como se puede ver en la tabla siguiente, coincide con la suma del tamaño de las ventanas de transmisión y de recepción:

3. El identificador que se pondría en cada trama aparece entre paréntesis en la figura.

| Protocolo | Número de identificadores necesarios | Tamaño de la ventana máximo si se utilizan n bits para los identificadores |
|---|---|---|
| *Stop & Wait* | 2 | 1 |
| *Go back N* | $W + 1$ | $2^n - 1$ |
| Retransmisión selectiva | $2W$ | $2^n - 1$ |

Identificadores necesarios y ventana máxima en un protocolo de ventana.

Finalmente, el tamaño de la ventana máximo $W_{máx}$, según el número de bits ($n$) que se utilicen en el campo del identificador de trama, será de $W_{máx} + 1 = 2^n$ en el protocolo *Go back N* y $2W_{máx} = 2^n$ en el de retransmisión selectiva.

## 5. Otras consideraciones sobre el nivel de enlace

Hasta ahora, hemos visto las funciones más importantes asociadas al nivel de enlace, que son la sincronización de las tramas, la detección y el control de errores y el control de flujo. En este apartado se explican algunos detalles que también podemos encontrar en un protocolo de nivel de enlace.

El objetivo de este apartado es, además, dar ejemplos de los contextos donde se puede encontrar el nivel de enlace.

### 5.1. *Piggybacking*

Hasta este momento siempre hemos considerado que había una estación que transmitía tramas de información (el primario) y otra que las confirmaba (el secundario). Ahora bien, hay casos en los cuales las dos estaciones se intercambian tramas de información recíprocamente y, por lo tanto, actuarían como primario y secundario al mismo tiempo y tendrían que alternar las tramas de información con las confirmaciones.

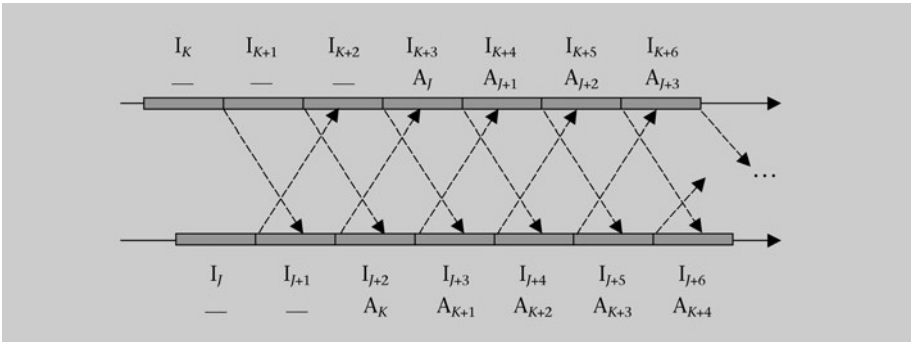
Las tramas de confirmación suelen tener un tamaño pequeño porque la única información relevante es la del identificador de la trama que confirman. La mayor parte de los bits de estas tramas la ocupan los campos de control (indicadores de sincronización, CRC, etc.).

No nos hemos preocupado de esto hasta ahora porque la eficiencia que hemos medido es la obtenida en el sentido primario-secundario. Sin embargo, si las dos estaciones envían información, nos interesará conseguir que la eficiencia en los dos sentidos sea tan alta como se pueda.

Una manera de aumentar la eficiencia en esta situación es incorporar las confirmaciones a las tramas de información (de este modo nos ahorramos la transmisión de los otros campos de control de las confirmaciones).

La manera de hacerlo consiste en destinar un campo de la trama de información al identificador de la trama que se quiere confirmar del primario contrario. Esta técnica se conoce como *piggybacking*.

La figura siguiente muestra el diagrama de tiempo que se obtendría utilizando el *piggybacking*:




Confirmación de las tramas con *piggybacking* en un protocolo de transmisión continua.

## 5.2. Sincronismo de protocolo

En la descripción que hemos hecho hasta ahora de los protocolos de recuperación de errores y control de flujo, hemos supuesto implícitamente que los protocolos estaban correctamente inicializados. Es decir, hemos supuesto que en los

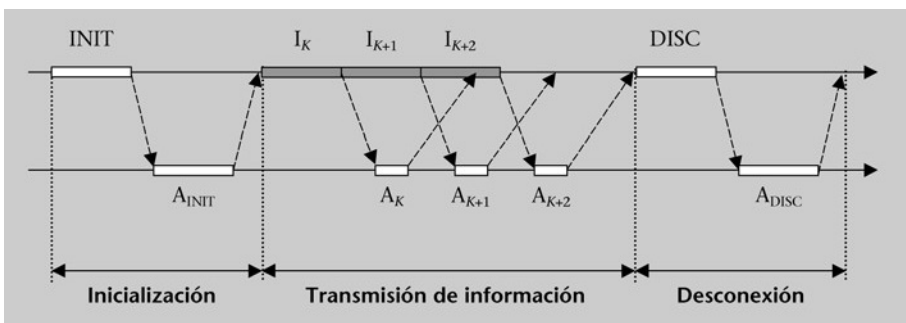
diagramas de tiempo que hemos utilizado no hay tramas de información ni confirmaciones anteriores a las mostradas que no puedan introducir ningún tipo de ambigüedad en los identificadores. En la práctica esto no es así. Antes de que las dos entidades de nivel de enlace puedan empezar a intercambiarse tramas de información, es necesaria una fase de inicialización.

Se podría pensar que sólo hay que hacer la inicialización cuando las dos entidades se ponen físicamente en marcha por primera vez. Esto no es posible porque, en un caso real, puede pasar que la ejecución de una de las entidades de nivel de enlace se quede temporalmente fuera de servicio, lo cual haría que se perdieran las tramas almacenadas en la memoria de transmisión y recepción, y se perdería la sincronización; es decir, se tendría que ejecutar de nuevo la fase de inicialización.

Además de la fase de inicialización también hay una fase de desconexión, en la que se acuerda la finalización del enlace. Esta fase permite liberar los recursos que se le dedican, como las memorias intermedias para almacenar las listas de transmisión y recepción.

Un protocolo que tiene una fase de inicialización y una fase de desconexión se conoce como protocolo orientado a la conexión (*connection oriented*). De otro modo, hablaríamos de protocolo no orientado a la conexión (*connectionless*).

La figura siguiente muestra la fase de inicialización y la de descomposición. La trama INIT solicita la inicialización, que es confirmada por la trama $A_{INIT}$. Las tramas DISC y $A_{DISC}$ funcionan de manera análoga para la desconexión. Estas tramas no están numeradas (por este motivo son conocidas como tramas no numeradas o *unnumbered frames*). En las tramas hay un campo de control que indica el tipo de trama, y que permite diferenciarlas:




Fases de inicilización y desconexión en un protocolo orientado a la conexión.

Dado que para poder hacer una recuperación automática de errores necesitamos numerar las tramas, estos protocolos deberán ser protocolos orientados a la conexión.

Sin embargo, no todos los protocolos de nivel de enlace son orientados a la conexión. Según el tipo de red o aplicación, es posible que no se utilice un protocolo de recuperación de errores en el nivel de enlace y que, por lo tanto, las tramas no deban numerarse. En este caso tendríamos un nivel de enlace no orientado a la conexión que haría la detección de errores, pero descartaría las tramas erróneas en lugar de intentar recuperarlas.

Hay motivos que pueden justificar un nivel de enlace no orientado a la conexión: en aplicaciones en tiempo real en las que se transmite voz o vídeo, por ejemplo, es posible que el retardo que se necesite para poder hacer la recuperación de errores no sea aceptable.

## 5.3. Importancia del nivel de enlace según el contexto

Hasta este momento hemos explicado las funciones que podemos encontrar en el nivel de enlace. Es importante entender, sin embargo, que el nivel de enlace no siempre efectúa todas las funciones explicadas. Esto depende mucho del contexto donde se encuentre el nivel de enlace.

**Recuperación de errores en el nivel de transporte**

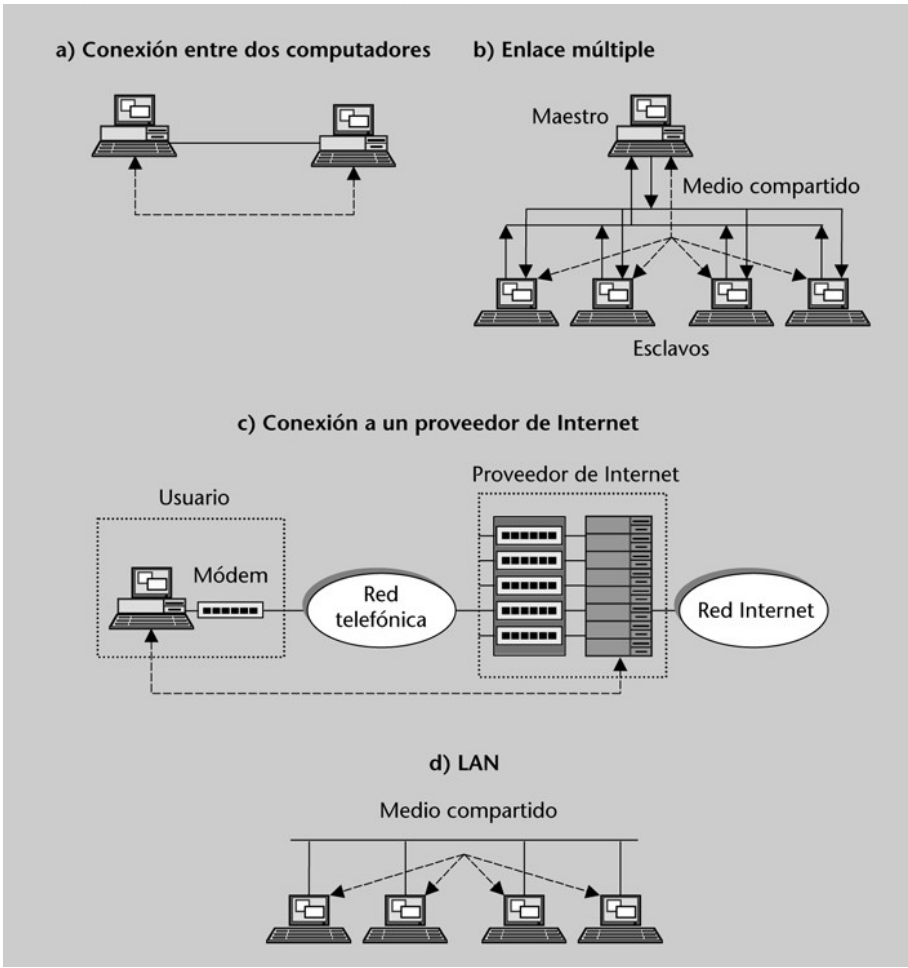
Hay redes en las que el nivel de enlace no lleva a cabo una recuperación de errores, sino que simplemente descarta las tramas erróneas. En este caso, si es necesario, la recuperación de errores la hacen los niveles superiores, normalmente el nivel de transporte.

**Notación**

En esta figura, las líneas discontinuas conectan los dispositivos entre los que se establece el protocolo de nivel de enlace.

A continuación, describiremos las diferentes situaciones representadas en la figura:

a) Comunicación punto a punto entre dos computadores; por ejemplo, la comunicación por el puerto serie entre dos PC para poder hacer una trans-




Ejemplos de contextos en los que se puede encontrar el nivel de enlace.

ferencia de ficheros. A causa del reducido número de elementos que intervienen en este caso, normalmente toda la arquitectura de comunicaciones estará integrada en el mismo programa. Evidentemente, aquí no habrá nivel de red y el nivel de enlace será responsable de la recuperación de errores.

b) Enlace multipunto. Esta situación es semejante al caso anterior, pero lo que entra en comunicación es una comunidad de computadores y lo hace a partir de un medio compartido. Para poder comunicarse sin interferirse

entre sí, uno de éstos (el maestro en la figura) se encarga de arbitrar las comunicaciones. Todas las comunicaciones son entre el maestro y otro de los computadores (los esclavos), o viceversa.

Esta condición es representada por las flechas que hay entre el maestro y cada uno de los esclavos (en los dos sentidos). En este entorno podemos interpretar que hay un enlace punto a punto entre el maestro y cada uno de los esclavos, y que el maestro selecciona alternativamente uno de los enlaces posibles según algún algoritmo de arbitraje.

c) El entorno de acceso a Internet es, actualmente, uno de los más habituales. Millones de usuarios lo utilizan para acceder a Internet. En este entorno el protocolo de nivel de enlace se establece entre el computador del usuario y el computador del proveedor de Internet. Aquí el usuario se conecta al proveedor mediante un módem. El proveedor dispone de una batería de módems para que múltiples usuarios puedan conectarse simultáneamente.

d) Red de área local (LAN). Una característica de estos tipos de redes es que la comunidad de computadores comparte el medio de transmisión, igual que en el caso b). La diferencia fundamental, sin embargo, es que ahora no hay un árbitro que selecciona uno de los enlaces posibles, sino que el algoritmo de acceso está distribuido entre los computadores que acceden al medio.

## 5.4. Multiplexación en el nivel de enlace

El concepto de *multiplexación* ya se ha introducido anteriormente. La idea de la multiplexación es utilizar un único medio para la transmisión de diferentes flujos de información. Obviamente, hay que definir un mecanismo que permita separar los distintos flujos con respecto a la recepción.

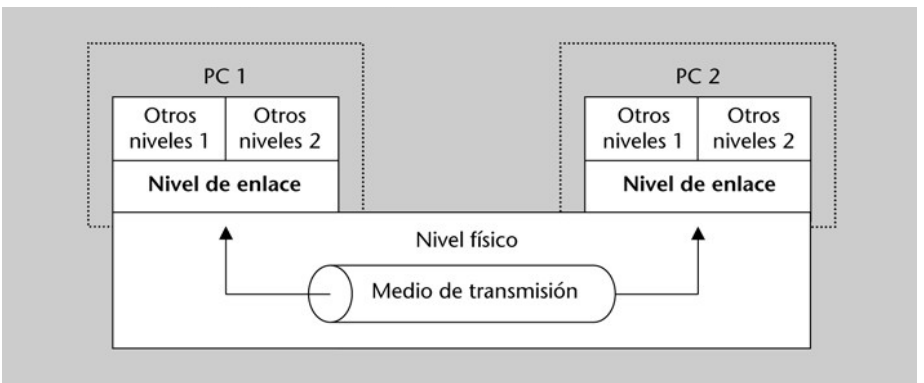
Ahora veremos que la multiplexación también se puede encontrar en un protocolo de nivel de enlace. De hecho, es habitual que esta técnica se utilice en cualquier nivel de la arquitectura de comunicaciones.

En primer lugar, veamos por qué puede ser útil tener multiplexación en el nivel de enlace. El primer ejemplo es el enlace multipunto, que ya hemos introducido (ved el esquema b) de la figura anterior). El hecho de tener el medio com-

partido podemos interpretarlo como si hubiera un único enlace donde se multiplexan las tramas de las distintas estaciones. En este caso, podríamos interpretar el algoritmo de arbitraje que utiliza el maestro como el mecanismo de multiplexación.

Dentro de este enlace compartido necesitaremos distinguir cada una de las estaciones. Esto se consigue con una dirección de nivel de enlace. Es decir, uno de los campos de control de las tramas es una dirección que identifica cada estación.

Otro ejemplo de multiplexación en el nivel de enlace es el que muestra la figura siguiente. En este ejemplo hay un nivel físico y de enlace común y, por encima, dos arquitecturas de comunicaciones (conjunto de protocolos) diferentes. En este caso el nivel de enlace lleva las unidades de información de la arquitectura 1 o 2 a su arquitectura par 1 o 2, respectivamente.




Ejemplo de multiplexación en el nivel de enlace.

Este caso no es extraño en la práctica, ya que hay numerosos protocolos de comunicaciones que puede interesar que convivan dentro de un mismo entorno y que compartan el mismo enlace; por ejemplo, en una red de área local.

Para poder distinguir la arquitectura que transmite las tramas y, por lo tanto, la arquitectura a la cual hay que entregarlas, también se utilizan direcciones de nivel de enlace.

## 6. Ejemplo: HDLC y ATM

Como ejemplo de un estándar de nivel de enlace, en este apartado describiremos algunas de las características del HDLC (*High Level Data Link Control*) y del ATM (*Asynchronomous Transfer Mode*). HDLC ha sido estandarizado por el organismo ISO y muchos otros protocolos de nivel de enlace son derivados de allí, mientras que ATM es un estándar de la ITU-T.

### 6.1. HDLC

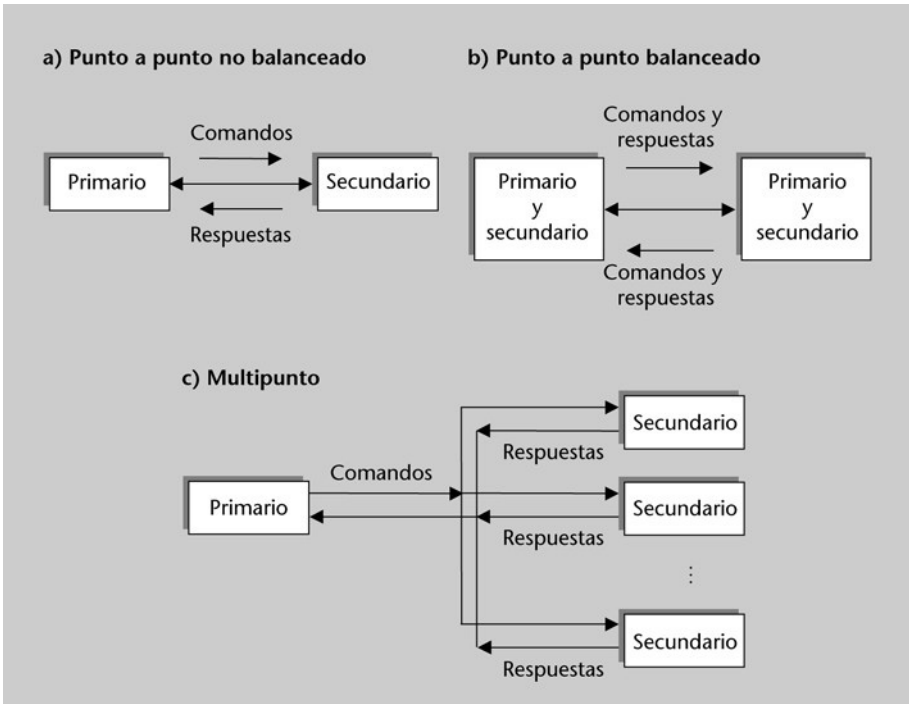
El HDLC ha sido diseñado como un protocolo de nivel de enlace de propósito general que se puede utilizar en numerosas situaciones. Por este motivo se han definido las tres configuraciones de la figura siguiente. En esta figura los nombres primario y secundario no hacen referencia a la estación que envía las tramas de información y las confirmaciones, como hasta ahora, sino que esta terminología se utiliza para distinguir ciertos privilegios que tiene el primario. El privilegio principal del HDLC es que puede hacer de árbitro en un enlace multipunto (lo que hemos denominado *maestro*).

**Ejemplos de protocolos derivados del HDLC**

Algunos protocolos que se han derivado del protocolo HDLC son los que mencionamos a continuación.

- LLC: se utiliza en las redes de área local (LAN).
- PPP: se utiliza en Internet.
- LAPM: es utilizado en el campo de la comunicación entre módems.
- LAPD: se utiliza como canal de señalización (canal D) en el acceso a la red telefónica digital ISDN.
- LAPB: se utiliza para conectar una estación a una red de gran alcance X.25.

En el HDLC las tramas que envía el primario se conocen como comandos, y las que envía el secundario, como respuestas. Si en la configuración todas las estaciones son primarios y secundarios al mismo tiempo, se dice que es una configuración balanceada; de lo contrario, se trata de una configuración no balanceada. Evidentemente, una configuración multipunto debe ser no balan-




Configuraciones posibles en el protocolo HDLC

ceada, porque sólo puede haber una estación que sea primario (que haga de árbitro).

El HDLC permite tres modos de operación distintos:

a) *Normal response mode* (NRM). Este modo se utiliza en configuraciones no balanceadas (casos a) y c) de la figura anterior). En este modo de operación los secundarios (o esclavos) sólo pueden transmitir tramas en respuesta a los comandos enviados por el primario (o maestro). Este modo se utiliza típicamente en el enlace entre un computador y uno o más periféricos.

b) *Asynchronous response mode* (ARM). Este modo se utiliza en configuraciones punto a punto no balanceadas (caso a) de la figura). A diferencia del modo NRM, el secundario puede transmitir sin que el primario lo solicite. Este modo es el menos usado en la actualidad.

c) *Asynchronous balanced mode* (ABM). Este modo se utiliza para configuraciones punto a punto balanceadas (caso b) de la figura). Los protocolos más usados derivados del HDLC, como el LAPM, el LAPD y el LAPB, son derivados de este modo.

Independientemente del modo, en el HDLC se definen tres tipos de tramas:

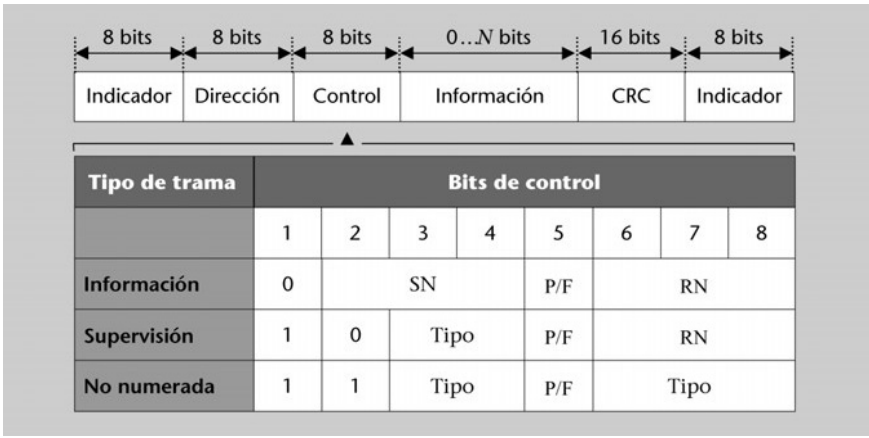
a) Tramas de información. Equivalen a las tramas de información a las que nos hemos referido hasta ahora. Para la recuperación de errores y control de flujo se utiliza *Go back N* o retransmisión selectiva (en el cual muchos de los derivados implementan sólo *Go back N*). Para las confirmaciones se utiliza la técnica de *piggybacking*.
b) Tramas de supervisión. Equivalen a las tramas de confirmación a las que nos hemos referido hasta ahora. Se utilizan para las confirmaciones cuando no se envían tramas de información.
c) Tramas no numeradas. Se usan en las fases de inicialización y desconexión y para enviar información de control suplementaria.

El HDLC es un protocolo con una estructura de las tramas como la que se muestra en la figura siguiente. Los indicadores de inicio y final de trama valen 01111110, y se utiliza la técnica de la inserción de ceros para la transparencia. La comprobación de errores se realiza con códigos CRC, concretamente con el polinomio generador CRC-CCITT.

El contenido del campo de dirección depende del modo de operación. Por ejemplo, en un enlace multipunto, si trabajamos con el modo NRM, sólo hay un primario y cada secundario tiene una dirección única. En este caso, en el campo de dirección se pone siempre la dirección del secundario. En el modo ABM, en cambio, se pone la dirección propia cuando se quiere enviar un pedido (la estación actúa como primario) y la dirección contraria cuando se envía una respuesta (la estación actúa como secundario).

Los bits de control son los siguientes:

- SN. Lleva el número de secuencia de una trama de información.
- RN. Lleva el número de secuencia de la trama confirmada.
- P/F (*Poll/Final* bit). Este bit tiene muchos usos, algunos de los cuales son los siguientes:
  - El primario pone este bit a 1 para solicitar una respuesta urgente de un secundario.
  - El secundario pone el bit a 1 para indicar el final de una respuesta.
- Tipo. El tercer y cuarto bits codifican el tipo de trama de supervisión, y el sexto, séptimo y octavo, el tipo de trama no numerada.




| Tipo de trama | Bits de control | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Información | 0 | SN | | | P/F | RN | | |
| Supervisión | 1 | 0 | Tipo | | P/F | RN | | |
| No numerada | 1 | 1 | Tipo | | P/F | Tipo | | |

Estructura de las tramas en el protocolo HDLC.

El estándar describe el tipo de trama y el comportamiento que se debe seguir en todas las situaciones posibles.

En este apartado sólo se han presentado los ingredientes básicos del estándar para que el estudiante se haga una idea de todos los puntos que involucra la estandarización de un protocolo de nivel de enlace.

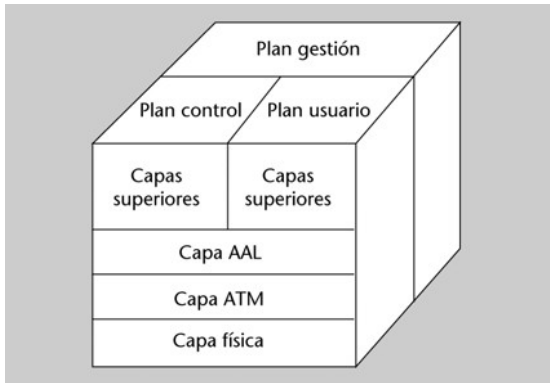
## 6.2. ATM

ATM es una tecnología de conmutación de paquetes que se basa en el uso de líneas de transmisión rápidas y fiables. Eso le permite conseguir altas velocidades de transmisión, no sólo porque las líneas lo permiten, sino también porque los protocolos intermedios (transporte, red, enlace) pueden ser mucho más simples que los otros que hemos visto, gracias a la fiabilidad de las líneas.

Vamos a describir la funcionalidad del nivel de enlace a ATM, para poder compararlo con el HDLC.

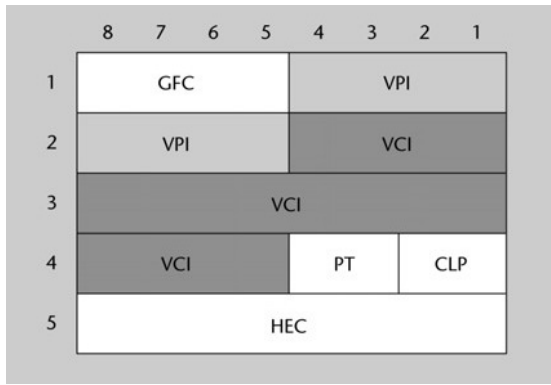
En la figura siguiente se puede ver la arquitectura de protocolos de ATM. Donde se llevan a cabo las tareas que hemos asociado al nivel de enlace en terminología OSI es en la 'capa ATM'.

La capa ATM define que la transmisión de datos sobre el medio físico se haga en paquetes o celdas de tamaño fijo. Este tamaño se fijó en 53 bytes, 48 de infor-




mación y 5 de cabecera. Un tamaño fijo permite un encaminamiento más eficiente, y un tamaño pequeño permite que las celdas con alta prioridad sufran retrasos muy pequeños en las colas.

En la figura siguiente se puede ver el formato de la celda ATM:




Los campos de la cabecera son:

- GFC: *generic flow control*. Este campo sólo se utiliza en la interfaz usuario-red, y no en las conexiones dentro de la red. Permite efectuar control de flujo en esta interfaz para diferentes calidades de servicio. En ATM las transmisiones pueden ser controladas o no controladas. Los diferentes valores de estos 4 bits permiten especificar diferentes tipos de control.
- VPI: *virtual path identifier*. Es un campo de encaminamiento.
- PT: *payload type*. Este campo indica el tipo de información que transporta la celda. El primer bit en 0 quiere decir que es información de usuario, y

entonces el segundo bit indica si se ha experimentado congestión o no. Si el primer bit vale 1 la celda transporta información de gestión de la red.

- CLP: *cell loss priority*. Se utiliza en caso de congestión. Si este bit vale 0, la celda no tendría que ser descartada, a no ser que no haya más remedio, mientras que un valor 1 indica que puede ser descartada.
- HEC: *header error control*. Éstos son 8 bits que sirven para proteger los 32 bits restantes de la cabecera. Eso quiere decir suficiente redundancia para no sólo detectar errores, sino también para intentar corregir.

Capítulo IV

# Tecnologías de red de área local

*José M. Barceló Ordinas*
*Jordi Íñigo Griera*
*Enric Peig Olivé*
*Jaume Abella i Fuentes*
*Guiomar Corral i Torruella*

## Introducción

Las redes de área local (LAN, local area network)surgen como una tecnología capaz de conectar ordenadores en un ámbito reducido, como por ejemplo, una oficina, un edificio o un departamento de una universidad. Actualmente, gracias a la creciente implantación de la informática doméstica y el coste reducido de los equipos, comienza a ser habitual encontrar redes locales incluso en los hogares.

El objetivo principal de una red de área local es permitir que unos recursos se compartan entre diversos ordenadores, tanto hardware especializado (impresoras, *scanners*, sistemas de *back-up*, etc.) como software (aplicaciones y datos asociados a los mismos que han de ser utilizados por diversos usuarios).

En este capítulo veremos los principios de funcionamiento de las redes de área local y las tareas que se deben llevar a cabo para conseguir que haya comunicación entre todas las estaciones de una red.

Presentaremos las diferentes topologías posibles, enumerando las ventajas e inconvenientes de cada una, y los diferentes protocolos de acceso al medio disponibles en la actualidad. Asimismo evaluaremos los parámetros más característicos de los diferentes protocolos para poder decidir cuál es el más adecuado a cada uno de los diversos escenarios posibles.

A continuación, describiremos el estándar Ethernet, con toda seguridad el más significativo en el campo de las redes locales.

Finalmente, veremos como se aplican los conceptos de conmutación en el diseño de redes de área local e introduciremos el concepto de VLAN, o LAN virtual

## 1. Conceptos básicos

Una red de área local es un sistema que permite la interconexión de ordenadores que están próximos físicamente. Entendemos por próximo todo lo que no sea cruzar una vía pública: una habitación, un edificio, un campus universitario, etc.

En el momento en que una red tiene que atravesar una calle, o una vía pública en general, una compañía de telecomunicaciones deberá establecer la comunicación, porque son las únicas autorizadas para pasar líneas por zonas públicas.

Como comentábamos anteriormente, la filosofía de diseño de las redes locales es la difusión de tramas con medio compartido. Eso quiere decir que las estaciones de la red están conectadas en el mismo medio de transmisión y, cuando alguna tiene información para alguna otra (o para varias), simplemente la pone en el medio. El resto de estaciones, que mientras no envían información están atentas a lo que hay en el medio, se quedan la trama si son destinatarias de la misma, o la ignoran si no lo son.

Últimamente, se ha aplicado a las redes de área local el concepto de conmutación que veíamos cuando describíamos las redes de gran alcance para conseguir mejores rendimientos de funcionamiento.

Otra mejora importante ha sido la aparición de las redes de área local sin hilos (*wireless* LAN). En estas redes, el enlace entre estaciones no se hace mediante cables, sino mediante enlaces radioeléctricos. Eso permite prestaciones tan interesantes como la movilidad o la ubicuidad, y también la disminución de costes a la hora de instalar una red local.

En una red local es imprescindible identificar los ordenadores que forman parte de la misma. Cuando un ordenador genera una trama para otro, además de los datos que quiere enviar, pone el identificador del ordenador (u ordenadores) destino y el suyo, para que quien reciba la trama pueda saber quién la ha enviado.

Para construir una red local, son necesarias básicamente dos cosas: hardware (tarjetas, cables, conectores) y un software que sea consciente de que hay diferentes máquinas conectadas y ofrezca los servicios necesarios para que las aplicaciones lo puedan aprovechar. Lo que es más lógico es que este software se integre en el sistema operativo y ofrezca a las aplicaciones la visión de la red como un recurso propio más.

Las primeras redes de área local sólo permitían que uno de los ordenadores de la red ofreciera recursos al resto (los servidores de red), los cuales sólo podían actuar de clientes de este servidor, sin capacidad de ofrecer nada. De un tiempo a esta parte, el software de red que hacen empresas como Novell, Microsoft o Apple permite que todas las estaciones puedan actuar de servidores y clientes a la vez.

Todas estas generalidades permiten ver que los parámetros principales que describen una LAN y que determinan sus características tanto a nivel de hardware como de software son:

- El medio de transmisión.
- La topología de la red.
- La política de acceso al medio.

Las posibles topologías de las redes y las diferentes políticas de acceso al medio que existen se verán en los siguientes apartados.

El medio de transmisión hace referencia a los elementos necesarios para que la información viaje entre las estaciones de la red. Hoy en día existen dos grandes tipos de LAN: las LAN cableadas y las LAN sin hilo. En LAN cableadas se utiliza cable coaxial, par trenzado o fibra óptica, con velocidades que van desde los 10 Mbps hasta los 10 Gbps.

Las LAN sin hilos utilizan transmisión por ondas, ya sean a frecuencias de radio, de infrarrojos o de microondas. Actualmente hay que destacar las redes que funcionan en los 2,4 GHz y 5 GHz. Estas dos subbandas (dentro de las microondas) son de uso libre, regulado, respectivamente, en todos los países del mundo.

– *Uso libre* quiere decir que no hay que pedir permisos para utilizar transmisores en estas frecuencias, como sí que pasa en la mayor parte del espectro radioeléctrico.
– *Regulado* se refiere al hecho de que los transmisores no pueden exceder de ciertas limitaciones, básicamente de potencia transmitida[1].

Eso deja el campo libre a la implementación de LAN que aprovechen estas bandas espectrales: en particular el IEEE 802. 11 WiFi y el IEEE 802. 15 Bluetooth. Las redes sin hilo tienen un campo de aplicación muy amplio:

- LAN para equipos portátiles. En LAN donde los terminales son ordenadores portátiles, que se mueven dentro de los despachos o de un área reducida.

1. Son las que se conocen como bandas ISM, o *Industrial, Scientific and Medical Bands*.

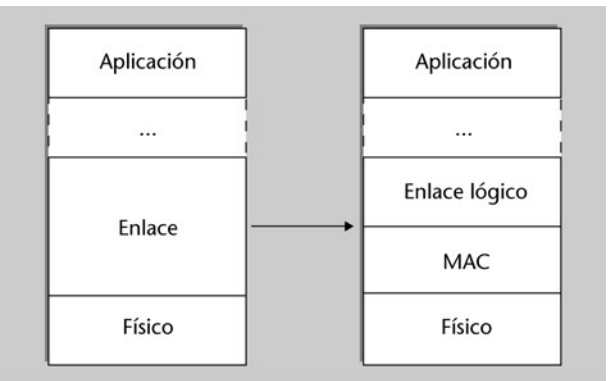
- LAN de bajo coste. Aunque los equipos no sean móviles, en ciertos ámbitos la instalación del cableado de una LAN (cableado estructurado o no) puede ser costosa, tanto con respecto al precio del mismo cable como, sobre todo, por las obras y reformas que hay que hacer en el edificio. Las redes sin hilo obvian este coste. Pequeñas oficinas, edificios antiguos o, incluso, hogares particulares son los principales destinatarios de estas redes.
- Interconexión de sistemas personales. La interconexión de ordenadores portátiles, teclados, ratones, impresoras, teléfonos móviles, etc. se puede solucionar también utilizando este tipo de tecnología.

## 1.1.Arquitectura de protocolos en las LAN

Una LAN es un sistema distribuido y, por lo tanto, puede ser diseñada con la filosofía de la separación de tareas por niveles, tal como se describe en el modelo OSI. Si repasamos este modelo, podemos deducir que para una red local nos hacen falta los niveles físico, enlace y aplicación. Los del medio (red y transporte) no porque están pensados para redes de gran alcance.

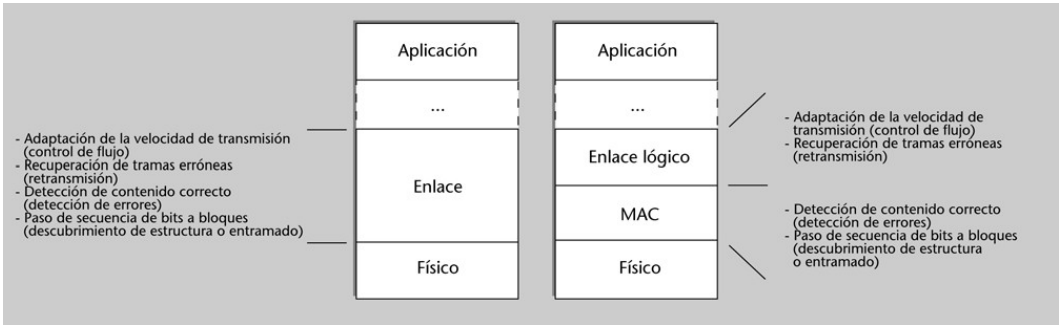
El nivel de enlace tal como está definido en el modelo OSI sirve para enlaces punto a punto, pero tiene una carencia importante en entornos donde varias estaciones comparten el medio de transmisión. En esta situación, hay que establecer quién, cuándo y cómo puede acceder al medio y quién no.

Incorporar esta funcionalidad se hace desdoblando el nivel de enlace en dos subniveles: MAC (*Medium Acces Control*) y LLC (*Logical Link Control*).



El subnivel MAC incorpora, pues, un protocolo que ejecutarán todos los terminales de manera distribuida, y que tiene que ofrecer un reparto eficiente (y justo) de la capacidad disponible del medio de transmisión en todos los terminales participantes de la LAN.

La figura muestra cómo se reparten las tareas asignadas al nivel de enlace original entre los dos nuevos subniveles:




Toda la funcionalidad "baja" del nivel de enlace ha pasado al subnivel MAC. Esta migración no es arbitraria. El nivel MAC necesita datos estructurados para añadir información de control, necesita las direcciones para poder saber si la trama que detecta en el medio va destinada a ello o no y necesita la detección de errores para saber si lo que recibe es correcto.

Como el medio de transmisión es compartido por todos los terminales, todos ellos "escuchan" todas las tramas que viajan, las descodifican, comprueban que sean correctas y miran si la dirección es la suya. Si las tramas no son correctas o bien no están dirigidas a ellos, simplemente las ignoran.

Si la trama es errónea, el terminal no tiene que hacer nada. No puede pedir la retransmisión, ya que nos encontramos en un absurdo: ¿cuál de todos los terminales es el que tiene que pedir la retransmisión? ninguno de ellos sabe quién es el receptor legítimo, ya que la trama es errónea.

Por eso todos los subniveles relacionados con el entramado o estructuración del flujo de bits pasan a formar parte del nivel más inferior que los necesita: el nivel MAC.

La literatura de redes usualmente cambia los nombres de los niveles bajos cuando el modelo OSI se aplica sobre terminales de una LAN. El modelo resultante para las LAN mantiene siete niveles (y no ocho como se desprende de lo que se ha explicado), por-

que el nivel MAC y el nivel físico se unen manteniendo el nombre de *nivel MAC*, y se deja como nivel de enlace el subnivel LLC. Eso se justifica en el hecho de que muchos protocolos MAC (los derivados de CSMA, como veremos más adelante) requieren funcionalidad adicional en la pura transmisión y recepción de bits para poder funcionar.

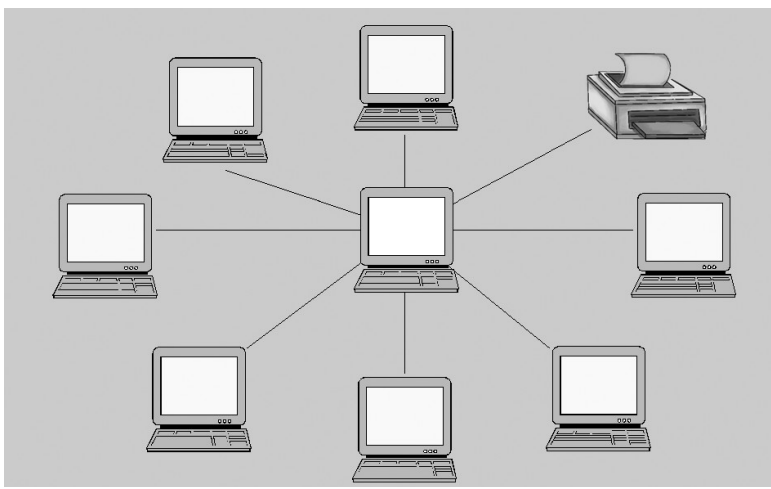
## 2. Topologías

La topología lógica de una red local define la manera como se conectan las estaciones que la forman al medio de transmisión. Dicho de otra manera, la topología es la forma que toma el medio que comparten las estaciones. Como hemos dicho, en el punto anterior, hay redes cableadas y redes sin hilos. En las primeras, el medio es tangible, por lo tanto se le puede dar una forma, mientras que en las redes sin hilos el medio de transmisión es el aire o el vacío, y por lo tanto parece difícil poder hablar de topología. Lo haremos, sin embargo, más adelante.

Básicamente hay tres topologías lógicas posibles:

- Estrella
- Bus
- Anillo

### 2.1. Topología en estrella



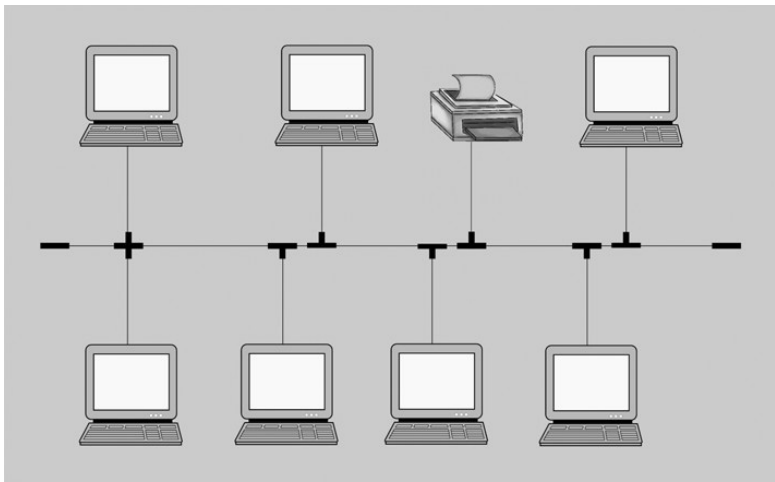
La topología en estrella consiste en conectar cada ordenador a un punto central, que puede ser tan sencillo como una simple unión física de los cables.

Cuando un ordenador pone una trama en la red, ésta aparece inmediatamente en las entradas de todo el resto.

Aunque se han definido estándares para este tipo de redes, actualmente ya casi no hay, ya que no aportan ninguna ventaja sobre el resto y sí muchos inconvenientes.

## 2.2. Topología en bus

La topología en bus consiste en un cable al cual se unen todas las estaciones de la red.



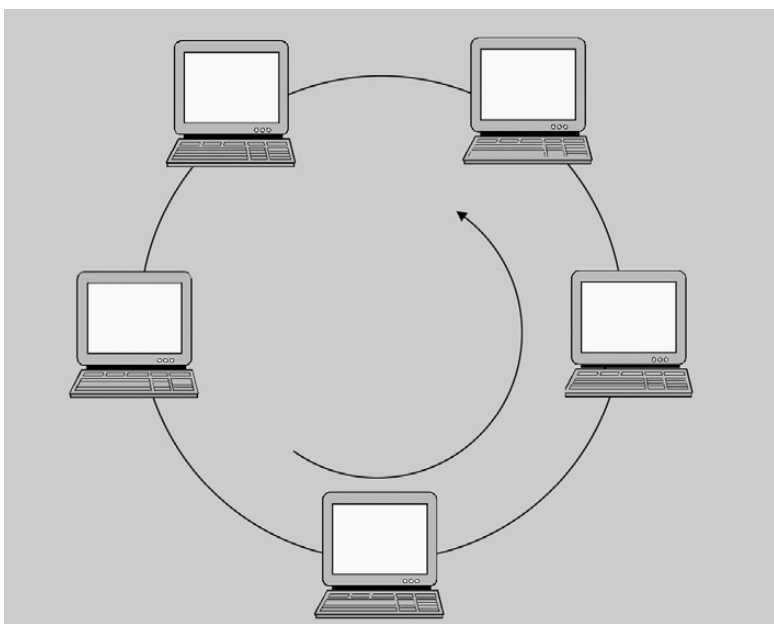
Todos los ordenadores están pendientes de si en el cable hay actividad. En el momento en que un ordenador pone una trama, todos los ordenadores la recogen y miran si son el destinatario. Si es así, se la quedan y, si no, la descartan.

Las primeras redes en bus utilizaban un cable coaxial grueso, conectores tipos BNC, y los ordenadores se conectaban con un dispositivo llamado *transceptor* (*transceiver*), que era exterior. Posteriormente, apareció una nueva versión, con un cable más fino (*thinethernet*) y con unos transceptores más pequeños, de manera que se podían integrar en el adaptador de red y así no se veían.

Los caprichos de la electrónica exigen que el cable esté "tapado" en los dos extremos, para que los bits no se "pierdan". Eso se hace con una resistencia de carga.
Cuando un ordenador pone una trama en el cable, ésta lo recurre completamente en las dos direcciones hasta los extremos donde es absorbida por los tapones.

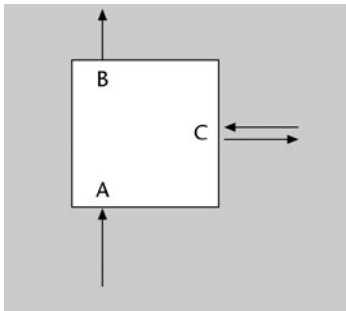
## 2.3. Topología en anillo

La topología en anillo consiste en conectar cada ordenador a dos más, de manera que se forme un anillo. Cuando un ordenador quiere enviar una trama a otro, ésta tiene que pasar por todos los que hay en medio: la circulación por el anillo es unidireccional.



El dispositivo que conecta el ordenador al anillo es el repetidor, un circuito con tres conexiones:

- Conexión de entrada de tramas desde el anillo al ordenador (A).
- Conexión de salida de tramas desde el ordenador al anillo (B).
- Conexión bidireccional, por la cual pasan todas las tramas que entran y salen del ordenador (C).




El repetidor tiene tres modos de trabajo:

- Modo escucha: el repetidor coge las tramas que le llegan por A y las pone simultáneamente en B y C, para que continúen por el anillo y para que el ordenador reciba una copia y lo analice. Si es el destinatario, se la queda y, en caso contrario, la descarta.
- Modo transmisión: el ordenador envía información a la red. Pone una trama en C, de manera que atraviesa el repetidor y sale por B hacia el ordenador siguiente del anillo.
- Modo cortocircuito: las tramas que llegan por A se ponen directamente en B sin dar una copia al ordenador. Este modo sirve para que el anillo continúe funcionando si el ordenador correspondiente no está en marcha.

## 2.4. Topologías lógicas y topologías físicas

La manera como se implementan realmente las topologías lógicas es lo que denominamos una topología física.

Las topologías en bus y en anillo comportan un serio problema de cableado a la hora de implementarlas.

Aunque es relativamente sencillo montar una red en bus o en anillo, es muy complicado mantenerla y ampliarla: cuando falla un cable o una conexión, la red entera deja de funcionar, y no hay nada que permita localizar el fallo: hay que comprobar la red entera, y eso muchas veces es complicado porque los cables pueden pasar por falsos techos o canaladuras escondidas.

Este problema hizo pensar en un nuevo diseño de red más controlable: el cableado estructurado. Se trata de hacer una preinstalación de red similar a la de

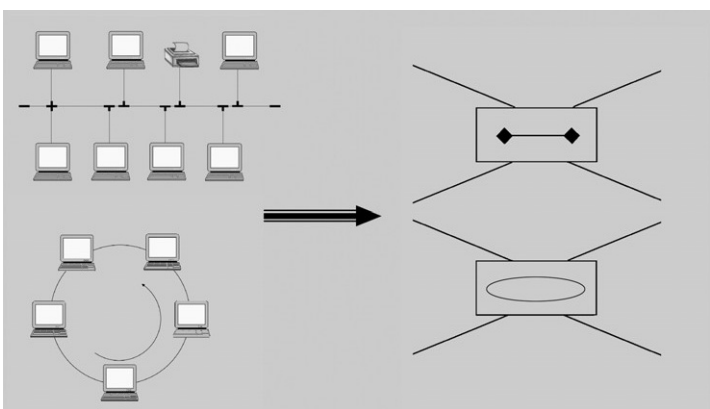
las redes telefónicas. A cada punto de trabajo se hacen llegar dos líneas: una para el teléfono y otra para los datos. Todos los cables llegan a una habitación, donde se establecen las conexiones: los cables de teléfono se envían hacia la centralita y los de los datos, hacia un dispositivo que permite la interconexión en red local, que, como veremos más adelante en este capítulo, puede ser un concentrador (*hub*) o un conmutador (*switch*).

El cableado estructurado ha sido estandarizado por el EIA/TIA-568 (*Electronics Industries Association/Telecomunication Industry Association*), que recoge la descripción de los cables, los conectores, los armarios, etc. para que todas las instalaciones sean homologables.

Normalmente se hace un cableado doble:

- Cableado horizontal: en cada planta (si hay que cablear varios) se ponen cables desde un armario hasta los puntos terminales.
- Cableado vertical: desde cada armario de planta hasta una habitación del edificio donde están los dispositivos de red, los encaminadores (*routers*) hacia el exterior, la centralita telefónica, etc.

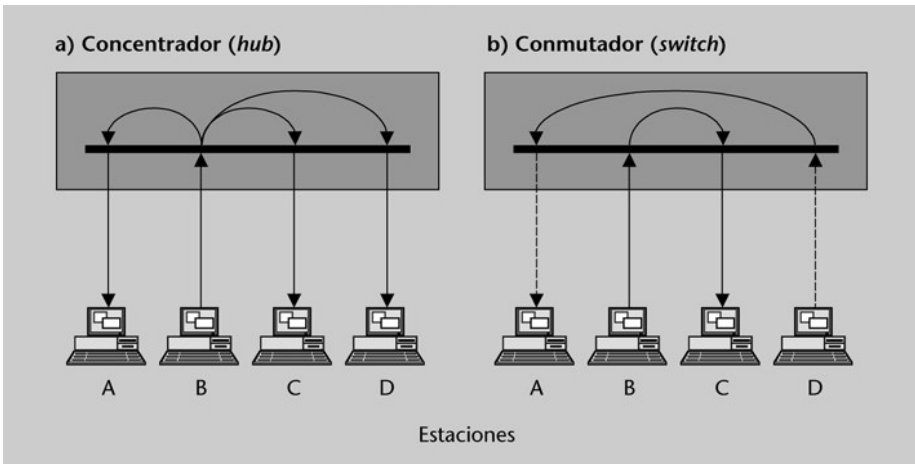
Se puede tener la tentación de pensar que un sistema de cableado estructurado es una topología lógica en estrella, porque todas las conexiones van a un punto único. Pero eso no es exactamente correcto. Los dispositivos llamados concentradores pueden ser de dos tipos: los que se comportan como un bus, o los que crean un anillo entre las estaciones. Así, las redes tienen forma de estrella, pero se comportan como un bus o un anillo. Por eso desde la aparición de los sistemas de cableado estructurado se habla de topología física en estrella y topología lógica en bus o en anillo.



En el fondo, es un bus (o un anillo) pero con las conexiones desde el bus (o desde el anillo) hasta las estaciones exageradamente alargadas.

Los costes de instalación de un sistema de cableado estructurado son muy altos, pero el mantenimiento es mucho más simple y barato: si falla un cable, sólo falla un puesto de trabajo, no toda la red, y, si falla toda la red, es que se ha estropeado el concentrador. Tanto un caso como el otro son muy rápidos de solucionar.

El hecho de que el concentrador sea un aparato electrónico que se comporta como un bus pronto hizo pensar que se podía aprovechar para hacerlo más sofisticado. Esta sofisticación vino por la incorporación de los conceptos de conmutación a las redes locales. Como hemos dicho, en un bus las tramas que una estación envía llegan a todo el resto, y sólo su auténtica destinataria coge las tramas. Si ahora tenemos un equipo que recibe una trama por una de sus entradas, puede tomar la decisión de no reenviarla a todo el resto sino a su único destinatario. Al fin y al cabo, puede identificar todas las estaciones porque las tiene conectadas a entradas (puertos) diferentes. Éste es el mecanismo de funcionamiento de los conmutadores o *switch*.




Topología en estrella.

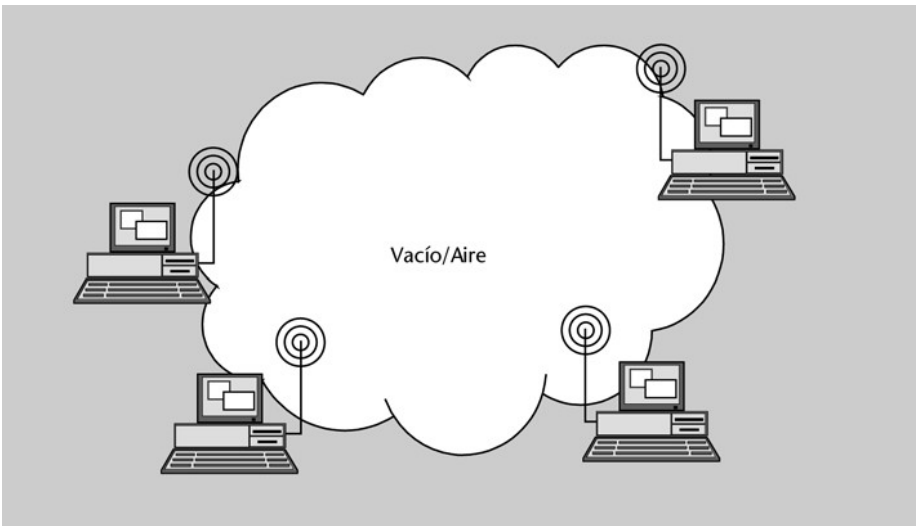
- En el concentrador, la estación B transmite una trama con destino a la estación C. Pero todas las estaciones la reciben. La estación C tendrá que

mirar la dirección MAC de destino para averiguar si la trama es para ella. El resto de estaciones descartarán la trama al ver la dirección MAC de destino.

- En el conmutador, la estación B transmite una trama con destino a la estación C. Ahora es el conmutador el que filtra la dirección MAC y sólo retransmite la trama por un puerto de salida, el de la estación C. Simultáneamente, la estación D se puede comunicar con la estación A.

## 2.5. Pseudotopología sin hilos

Una vez vistas las diferentes topologías que se pueden implementar con los medios guiados (cables, fibras ópticas), podemos deducir que una red sin hilos donde se utilizan ondas para los enlaces es equivalente a un bus, porque las ondas que envía una estación son emitidas en todas direcciones, de manera que son recibidas por todas las otras estaciones. Todas las tramas que no llevan la dirección MAC propia son descartadas por las estaciones receptoras.




No obstante, tratamos el "bus sin hilo" como un caso aparte, y se habla de pseudotopología (no es una topología diferenciada) para resaltar sus aspectos diferenciales:

- Visibilidad parcial. En un bus, los dos terminales más lejanos tienen que transmitir con bastante potencia para recibir la señal el uno del otro. En cambio, en redes sin hilos dos terminales pueden no tener conexión directa entre ellos, pero sí con un tercer terminal intermedio. Eso puede ser por dos razones: o bien por excesiva lejanía de los terminales extremos, o bien por sombras de señal producidas por obstáculos.
- Movilidad. Los equipos son móviles, por lo tanto los retrasos, la potencia, la relación señalruido, etc. no son factores estables durante el transcurso de una comunicación.
- *Half duplex*. En la actualidad no es posible hacer radiotransmisores *full duplex* (dentro de una misma frecuencia). Eso impide implementar algunos de los protocolos como CSMA/CD sobradamente utilizados en cableados en bus.
- Seguridad. Una red sin hilo no está limitada a un conjunto de puntos de conexión. El acceso está disponible dentro de un área, o mejor dicho, de un volumen con unas dimensiones imprecisas definidas por el radio de acción de las antenas instaladas en cada uno de los terminales. Cualquier terminal ajeno a la LAN que pase por el radio de acción de estas antenas podrá "escuchar" el tráfico de la LAN.

Eso exige que estas redes utilicen procedimientos criptográficos para proteger el tráfico de los legítimos usuarios contra ataques malévolos.

El IEEE 802. 1 x o el IPSec son estándares basados en criptografía (cifrado, integridad y autenticación) que ofrecen un nivel de seguridad suficiente para proteger los datos en tránsito contra la copia y manipulación por terceras partes.

Todos estos factores son determinantes a la hora de diseñar los protocolos para las LAN sin hilo.

## 3. Control de acceso al medio

Hemos visto que, independientemente de la topología que utilice una LAN, todas las estaciones comparten un único medio de transmisión que sólo puede utilizar una estación en cada momento. Esto se debe a que la señal que envía esta estación se difunde por todo el medio, lo ocupa e impide, por lo tanto, que ninguna otra estación pueda transmitir al mismo tiempo.

La estación que transmite debe tener la certeza de que es la única que lo hace en aquel momento y, además, tiene que dar a otras estaciones la posibilidad de enviar datos. No puede, por lo tanto, ocupar el medio indefinidamente.

El algoritmo de control de acceso al medio se encarga de repartir de manera ordenada, equitativa y eficiente el uso de este recurso único (el medio de transmisión) entre un conjunto de estaciones usuarias.

El control de acceso al medio (MAC) es un mecanismo que decide qué estación puede acceder en cada momento al medio para transmitir una trama de información.

El MAC proporciona un servicio de acceso múltiple a los usuarios de la red.

Como ya hemos mencionado, la unidad de información es la trama MAC. Puesto que una LAN, por definición, es una red de difusión, las tramas se propagan por el medio y todas las estaciones las reciben. Por lo tanto, es necesario identificar las estaciones de origen y de destino y transmitir esta información junto con los datos que nos proporciona el nivel superior. A partir de la dirección MAC de destino, las estaciones pueden deducir si la trama es para ellas y, además, consultando la dirección MAC de origen, pueden saber quién les envía la información. En caso de que una estación reciba una trama con una dirección MAC de destino que no coincida con la suya, se limita a eliminarla.

Se ha propuesto una gran cantidad de protocolos de control de acceso al medio para todo tipo de redes, tanto para redes cableadas como para redes inalámbricas. Actualmente, el acceso al medio es un tema de estudio muy importante en las comunicaciones móviles, en las que un usuario con un terminal de datos puede moverse continuamente.

Los protocolos de control de acceso al medio básicos se pueden clasificar en tres grandes grupos:

- Control de acceso al medio estático.
- Control de acceso al medio dinámico.
  - Acceso dinámico por control centralizado.
  - Acceso dinámico por control distribuido.
- Control de acceso al medio aleatorio.

Cada uno de estos accesos al medio tiene ventajas e inconvenientes, y se aplica a redes muy distintas. Los protocolos que más han triunfado en las LAN cableadas han sido uno por control distribuido (redes *token ring* o redes en anillo con paso de testigo) y otro de tipo aleatorio (redes Ethernet con topología en bus o estrella y acceso al medio CSMA/CD). En redes inalámbricas se utiliza una combinación de control de acceso al medio estático (FDMA/CDMA) y aleatorio (CSMA/CA). Todos estos protocolos se explican a continuación.

## 3.1. Control de acceso al medio estático

El control de acceso al medio estático, también denominado *acceso al medio por reserva*, consiste en mecanismos que se basan en el reparto del medio entre los diferentes usuarios, bien de una porción de tiempo, o bien de una porción del ancho de banda. Dos de los mecanismos principales (TDMA y FDMA) utilizan algunas de las ideas aplicadas a redes de gran alcance (multiplexación por división en el tiempo y multiplexación por división en frecuencia).

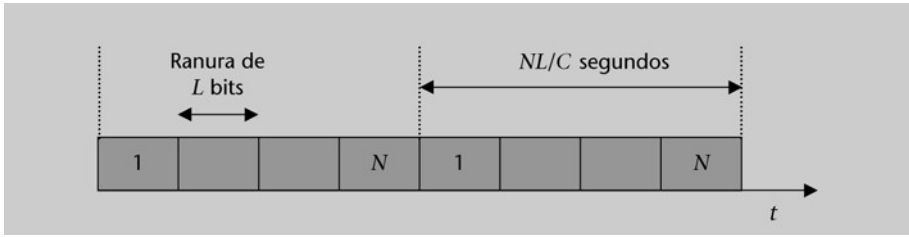
### *3.1.1. Acceso múltiple por división en el tiempo*

El acceso múltiple por división en el tiempo (TDMA [*Time Division Multiple Acces*]) consiste en dividir el canal (medio de transmisión) en ranuras de tiempo (*slots*) que se asignan estáticamente a las estaciones (usuarios).

Cada estación transmite en su ranura de tiempo. Supongamos que la red tiene los parámetros siguientes: $N$ usuarios, una capacidad del canal de $C$ bits por segundo (b/s o bps) y una longitud de las tramas de $L$ bits.

Si la duración de una ranura de tiempo se corresponde con la longitud de una trama MAC, tendremos lo siguiente.

- Tiempo de transmisión de una trama MAC: $t_t = L/C$ segundos (en este caso coincide con la duración de una ranura).
- Tiempo entre la transmisión de dos tramas consecutivas que pertenecen a la misma estación: $(N\text{-}1)L/C$ segundos.

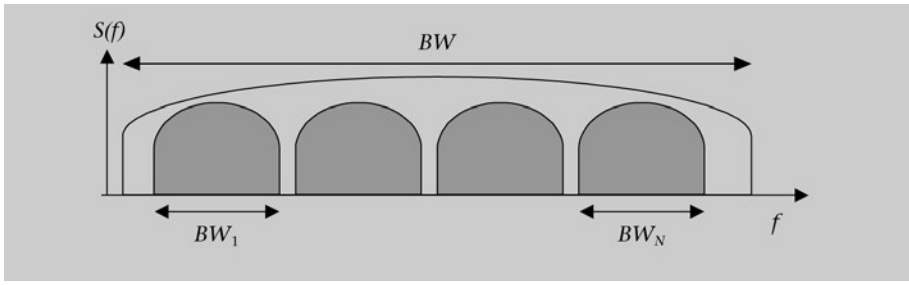



Acceso múltiple por división en el tiempo.

### *3.1.2. Acceso múltiple por división en frecuencia*

Supongamos que la red tiene los parámetros siguientes: $N$ usuarios y un ancho de banda del canal de $BW$ Hz.

En un acceso múltiple por división en frecuencia (FDMA [*Frequency Division Multiple Acces*]) se divide el ancho de banda $BW$ del medio de transmisión en $N$ subcanales. Se asigna un subcanal a cada estación para que transmita su información. Entonces, tenemos un ancho de banda de un subcanal de $BWi \approx BW/N$ (de hecho, un poco menos, ya que entre subcanal y subcanal hay un ancho de banda de guarda para que no se superpongan).

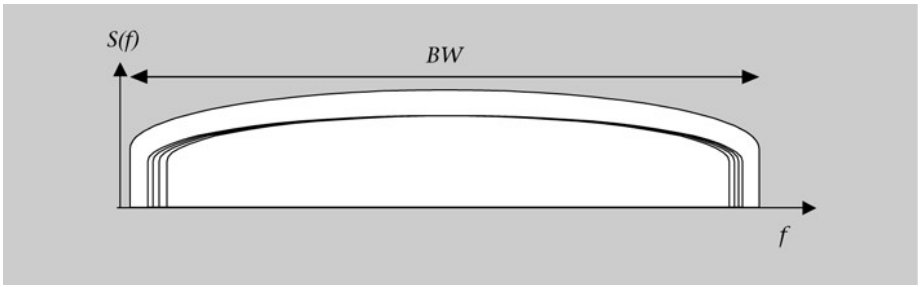



Acceso múltiple por división en frecuencia.

### *3.1.3. Acceso múltiple por distinción de códigos*

El CDMA (*Code Division Multiple Acces*) fue introducido por primera vez en sistemas militares por su robustez frente a distintos tipos de interferencia.

El funcionamiento del CDMA se basa en que todos los terminales transmitan dentro de la misma banda (100% de solapamiento entre terminales) y simultáneamente. Es decir, como un FDMA pero con todos los canales superpuestos.




Acceso múltiple por distinción de códigos.

A partir de esta descripción, el CDMA no podría funcionar[2] si todos los terminales transmitieran con la misma potencia, ya que la señal resultante es una mezcla indescifrable.

Sin embargo, el CDMA permite al terminal receptor disminuir la "potencia equivalente" de todos los terminales, a excepción del transmisor deseado, por un proceso matemático sencillo. Esto hace que la señal recibida resalte sobre el resto, de manera que las señales "no deseadas" se puedan tratar simplemente como ruido residual.

El procedimiento se basa en aumentar la redundancia de la señal transmitida para que sea "fácilmente deducible": los unos y los ceros de la secuencia de bits que se transmite se sustituyen por una secuencia de *n* bits para los unos, y la secuencia inversa para los ceros, diferente para cada canal. Esta secuencia es el código.

2. En el FDMA, se podía dar uso simultáneo del medio porque cada terminal utilizaba una banda de espectro diferente del resto.

*Ejemplo*

Para facilitar la explicación, supondremos que la secuencia está formada por señales positivas y negativas. Por ejemplo:

$$1\text{: } +1;\ +1;\ -1;\ +1$$
$$0\text{: } -1;\ -1;\ +1;\ -1$$

En este ejemplo, la transmisión resultante se debe transmitir cuatro veces más rápidamente que la secuencia original. Por lo tanto, para una misma modulación, el ancho de banda será cuatro veces mayor.

Siguiendo con el ejemplo, lo que tiene que hacer el receptor es aplicar el "código" multiplicándolo a la señal recibida. De esta forma, en cada caso:

$$(+1) \cdot (+1) + (+1) \cdot (+1) + (-1) \cdot (-1) + (+1) \cdot (+1) = +4 \Rightarrow \text{“1”}$$
$$(+1) \cdot (-1) + (+1) \cdot (-1) + (-1) \cdot (+1) + (+1) \cdot (-1) = -4 \Rightarrow \text{“0”}$$

Si el receptor recibe la transmisión de otro terminal, la señal estará codificada con otro código y, por lo tanto, el resultado estará atenuado respecto de la señal que tiene que recibir. Por ejemplo, si el código de este otro terminal fuera el siguiente:

$$1\text{: } +1;\ -1;\ -1;\ -1$$
$$0\text{: } -1;\ +1;\ +1;\ +1$$

Al descodificarlo, el primer terminal obtendría lo siguiente:

$$(+1) \cdot (+1) + (+1) \cdot (-1) + (-1) \cdot (-1) + (+1) \cdot (-1) = 0 \Rightarrow \text{nada}$$
$$(+1) \cdot (-1) + (+1) \cdot (+1) + (-1) \cdot (+1) + (+1) \cdot (+1) = 0 \Rightarrow \text{nada}$$

Este código generaría una interferencia nula sobre la señal anterior. Esto, sin embargo, sólo ocurre en códigos ortogonales, y no todos los códigos pueden ser ortogonales entre sí. En estos casos, las señales contribuyen a la señal final de manera atenuada. Por ello decimos que el procedimiento CDMA atenúa los canales no deseados y/o amplifica el canal deseado. El factor de amplificación viene dado por el número $n$ de símbolos que generamos por cada bit. Cuanto mayor sea este

valor, mejor será la relación señal-ruido (entendiendo como ruido el resto de los sistemas). Sin embargo, también será mayor (*n* veces) el ancho de banda necesario. En LAN y en telefonía se utilizan códigos de centenares de bits.

Utilizar secuencias fijas para cada bit no es la única manera de añadir redundancia. Una forma común de implementar el CDMA es generar secuencias seudoaleatorias a partir de una semilla (*seed*). En este caso, el terminal transmisor y el receptor deben conocer la semilla para generar el mismo código cada uno. Por lo tanto, los códigos no son repetitivos como se ha explicado antes, sino diferentes. Eso sí, la secuencia es conocida por el receptor legítimo, que será el único capaz no sólo de descodificarla, sino también de detectar la señal entre el ruido. Esta propiedad es muy interesante para mantener la confidencialidad de la transmisión, sobre todo si se tiene en cuenta que el CDMA es un procedimiento básicamente pensado para radiotransmisión.

### *3.1.4. Valoración del control de acceso al medio estático*

Los mecanismos estáticos presentan el inconveniente principal de que si una estación no tiene información para transmitir, el resto de las estaciones no puede aprovechar el canal, de manera que se están desperdiciando los recursos de la red.

**Desaprovechamiento de los recursos de la red**

Presentamos algunos ejemplos de las desventajas de los esquemas que acabamos de describir:

a) Hemos visto que, en un sistema TDMA, una estación puede transmitir tramas cada $(N-1)$ ranuras de tiempo. Si una estación no tiene tramas para transmitir, pero hay otra que sí las tiene, habrá ranuras vacías que la estación que tiene información para transmitir no podrá utilizar.
b) En el caso de un sistema FDMA, ocurrirá lo mismo con los subcanales de ancho de banda $BW/N$ Hz.

Pensad qué pasaría si en una autopista de unos pocos carriles no se permitiera que los coches cambiaran de un carril a otro. Podría ocurrir que en un carril hubiera muchos coches, mientras que otro iría prácticamente vacío. Esta autopista sería más ineficiente que otra en la que se permitiese que los coches cambiaran de carril cuando el que se les había asignado estuviese muy ocupado.

Los mecanismos estáticos ofrecen un rendimiento elevado si la red está muy cargada; es decir, si todas las estaciones tienen siempre información para trans-

mitir. Ésta no es la situación de una LAN, donde el flujo de tráfico es mayoritariamente de datos.

**Datos, voz e imagen**

Algunos autores hablan de *datos* para referirse a la transmisión de cualquier tipo de información, independientemente de su aplicación (voz, vídeo, imágenes, texto, etc.).

Ahora bien, la transmisión de estas aplicaciones requiere diferentes tipos de restricciones temporales. Por ejemplo, la voz y el vídeo se transmiten en tiempo real, mientras que las imágenes (fotografías) y el texto no necesitan ser transmitidos en tiempo real. Por este motivo muchos autores diferencian transmisión de aplicaciones en tiempo real (en general, voz y vídeo), y transmisión de aplicaciones en tiempo no real (comúnmente, aplicaciones de datos).

Las aplicaciones de datos producen tráfico a ráfagas. Lo denominamos *a ráfagas* porque las estaciones deben transmitir una trama tras otra (una racha de tramas) durante un tiempo, y después tienen que dejar de transmitir, o hacerlo a una velocidad más baja. Esto quiere decir que las estaciones acceden al medio durante intervalos cortos y esporádicos de tiempo. Con mecanismos de asignación estáticos como el TDMA o el FDMA y este tipo de tráfico a ráfagas, la red no puede obtener rendimientos altos. Los rendimientos serían mejores si el tráfico fuera continuo.

Nos interesan mecanismos que ofrezcan buenos rendimientos con tráfico de datos.

Un segundo inconveniente es la dificultad de asignar más canales si hay nuevos usuarios (estaciones) que se quieran unir a la red. Si hay $N$ estaciones ($N$ canales), ¿qué pasa si queremos añadir una estación nueva al sistema?

Nos interesan mecanismos en los que sea sencillo conectar y desconectar dinámicamente estaciones nuevas a la red sin que esto afecte mucho a su funcionamiento.

Sin embargo, no todo son inconvenientes en los mecanismos estáticos. Hay aplicaciones que no producen tráfico a ráfagas, sino que, al contrario, pueden producir un tráfico constante; es decir, información cada periodo de tiempo. Además, son aplicaciones en las que puede ser necesario que el retardo sea constante para todas las tramas; es decir, que las tramas no experimenten retardos muy diferentes. Estas características son las que ofrece precisamente un TDMA.

Para tráfico de voz y vídeo en tiempo real, un mecanismo estático de acceso al medio puede ofrecer muy buenos rendimientos.

Estas técnicas se utilizan poco en LAN cableadas. Sin embargo, por ejemplo, son técnicas que se utilizan en telefonía móvil (*Global System for Mobile Communications* o GSM).

## 3.2. Control de acceso al medio dinámico

De la lectura del apartado anterior se desprende que interesa encontrar mecanismos de acceso adecuados para el tráfico a ráfagas; es decir, que las estaciones puedan acceder al medio de manera que si una no tiene tráfico no ocupe el canal. Este tipo de control se puede obtener tanto mediante mecanismos de acceso dinámicos, como aleatorios.

### *3.2.1. Acceso dinámico por control centralizado*

El acceso dinámico por control centralizado usa técnicas de encuesta (*polling*) con una topología en estrella. Un nodo central se encarga de averiguar las necesidades de cada estación. El acceso al medio se efectúa mediante peticiones y respuestas entre las estaciones y el centro. Las estaciones envían peticiones antes de transmitir las tramas. El concentrador hace un turno rotatorio (*round robin*) entre sus puertos de entrada y da acceso con una respuesta a una de las estaciones que haya efectuado una petición.

**Turno rotatorio**
Un turno rotatorio, o *round robin*, es un algoritmo de planificación por el cual se controla el orden de transmisión siguiendo un orden numérico; por ejemplo, el del número de puerto (primero transmite el puerto 1, después el puerto 2, etc. y cuando se llega al último puerto se vuelve a empezar por el primero).

### *3.2.2. Acceso dinámico por control distribuido*

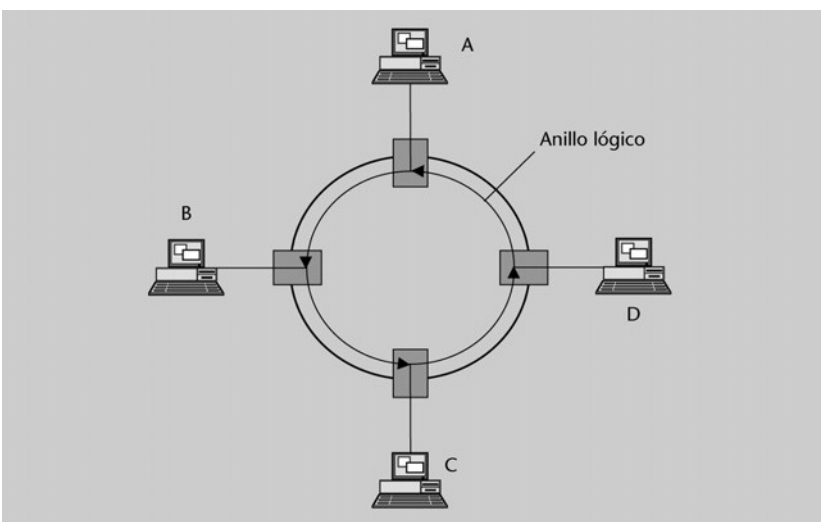
El acceso dinámico por control distribuido es más efectivo con topologías en bus o en anillo, en las que no hay ningún nodo central y las estaciones están distribuidas a lo largo del medio de transmisión.

Una de las políticas de acceso al medio distribuido que más se utilizan es la política de acceso por paso de testigo. Este tipo de mecanismo tiene las características siguientes:

a) Se puede utilizar tanto en topologías en bus como en anillo.
b) Se basa en la generación de una trama especial conocida como testigo (*token*).
c) La estación que tiene el testigo puede transmitir tramas. Si la transmisión es no exhaustiva, sólo puede transmitir una trama; si es exhaustiva, puede transmitir más de una.
d) Cuando una estación acaba de transmitir, libera el testigo y la estación siguiente que quiera transmitir lo recoge.
e) Las estaciones que quieran transmitir y no tengan el testigo deberán almacenar las tramas a punto para transmitir en una memoria intermedia y esperar que les llegue su turno; es decir, que les llegue el testigo.

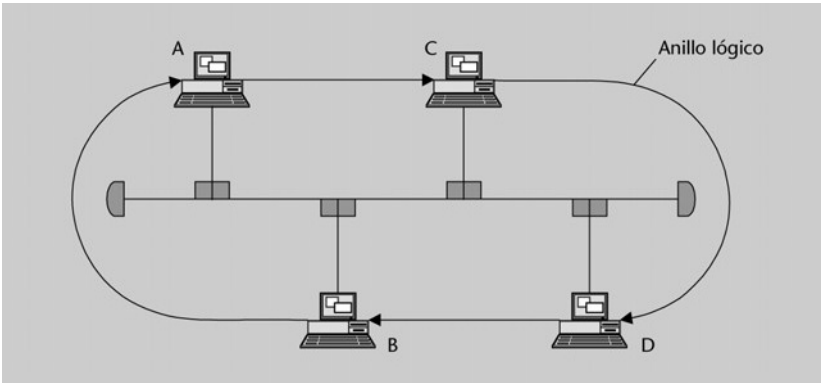
Ahora hay que decidir a quién se pasa el testigo una vez se ha transmitido una trama y, por lo tanto, a quién se le da permiso para transmitir. La solución más simple es formar un anillo lógico entre las estaciones; es decir, imponer un orden o una numeración lógica, de manera que cuando una estación acabe de transmitir pase el testigo a otra estación, que será siempre la misma.

Si asumimos una topología física en anillo, es muy sencillo elegir este orden. El anillo lógico coincide con el anillo físico. Por ejemplo, en la figura siguiente se puede ver cómo es posible imponer el mismo orden en el que están situadas las estaciones para pasar el testigo: A → B → C → D → A.




Topología física en anillo con paso de testigo.

En cambio, en una topología en bus la ordenación física de las estaciones no tiene por qué coincidir con la ordenación lógica del paso del testigo. Por ejemplo, si la ordenación física es A → B → C → D, hemos impuesto una ordenación lógica del paso de testigo A → C → D → B → A.




Topología física en bus con paso de testigo.

**Estándares para anillo y bus con paso de testigo**

El estándar para anillo con paso de testigo (*token ring*) es el IEEE-802.5 y el estándar para bus con paso de testigo (*token bus*) es el IEEE-802.4. Hay que aclarar que el bus con paso de testigo no triunfó nunca comercialmente. El IEEE estandarizó diferentes topologías con acceso al medio por paso de testigo, pero después la industria fue por otros caminos. IBM apostó por un MAC basado en la topología en anillo con paso de testigo y ninguna otra empresa consiguió comercializar con éxito el bus con paso de testigo. Si tenéis curiosidad por saber cómo funciona un *token bus*, podéis consultar el estándar IEEE-802.4 en alguno de los libros de la bibliografía.

## 3.3. Control de acceso al medio aleatorio

En los MAC estáticos, cada terminal tenía acceso todo el tiempo a una fracción *n*-ésima de la capacidad del medio. En los MAC dinámicos, el terminal tenía acceso durante una fracción *m*-ésima de tiempo a toda la capacidad del medio. La aparente simetría entre una solución y la otra no es tal –y de aquí viene la mejora de los MAC dinámicos–, ya que en los MAC dinámicos, *m* son los terminales

que quieren transmitir (no incluye los que escuchan) y, en cambio, *n* incluye todos los terminales de la red.

Los MAC aleatorios tienen un enfoque más radical: cada terminal tiene acceso todo el tiempo a toda la capacidad del medio.

En esta afirmación, aparentemente hay un contrasentido: cuando dos o más terminales pretendan disfrutar de estas prestaciones, el medio no lo podrá soportar.

Si suponemos, sin embargo, que cada terminal sólo transmite de manera esporádica, el sistema tiene una eficiencia máxima, ya que no hay que aplicar ningún algoritmo. Dicho de otra manera, los MAC aleatorios no implementan ningún algoritmo.

Sin embargo, cuando hay dos terminales que deciden transmitir, los dos lo harán independientemente uno de otro, de manera que puede pasar que las señales emitidas lleguen mezcladas a todos los terminales de la red y que, por lo tanto, sean intratables. Esto es lo que se conoce como *colisión*.

Cuando se produce una colisión, las tramas afectadas no son recuperables. Es necesario, por lo tanto, un procedimiento superior que controle esta eventualidad y actúe en consecuencia. Habitualmente, este procedimiento superior es un protocolo de enlace que recupera las tramas perdidas retransmitiéndolas.

Así pues, la eficiencia de los MAC aleatorios está fuertemente vinculada al tipo de tráfico que circula por la red: cuanta más probabilidad de colisión, menos eficiencia.

Estudiaremos ahora los diferentes MAC aleatorios desde este punto de vista. Hay cinco fundamentales:

a) Aloha puro o simplemente Aloha. No aplica ningún algoritmo de control de acceso al medio, ni tampoco de control de colisiones.
b) Aloha segmentado. Minimiza la probabilidad de colisión forzando la transmisión en momentos determinados mediante un reloj central.
c) CSMA (*Carrier Sense Multiple Access,* 'acceso múltiple por escucha de portadora'). Minimiza la probabilidad de colisión al comprobar antes de transmitir si hay alguna estación que esté transmitiendo.
d) CSMA/CD (*Carrier Sense Múltiple Access with Detection Collision*, 'acceso múltiple por escucha de portadora con detección de colisiones'). Complementa las prestaciones del CSMA y minimiza el impacto de las colisiones en el ren-

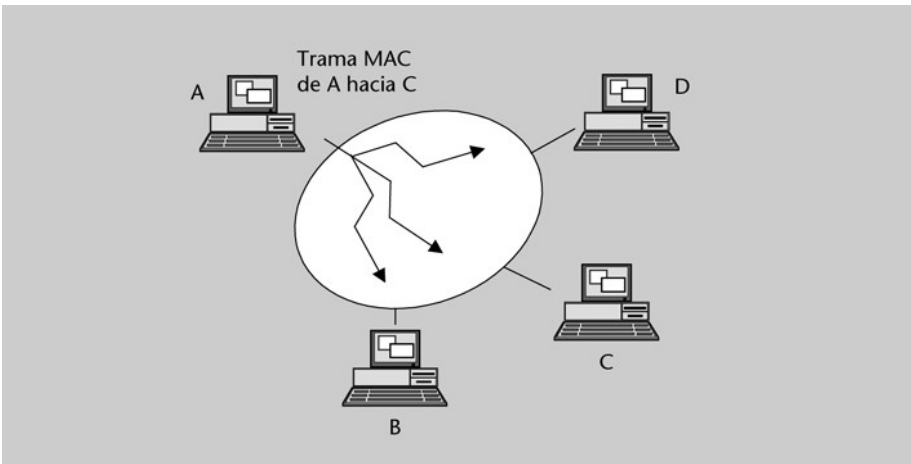
dimiento de la red acortando su duración. CSMA/CD es el único MAC aleatorio que no puede funcionar en medios no guiados. El resto sirve tanto para medios guiados como no guiados.

e) CSMA/CA (*Carrier Sense Múltiple Access with Collision Avoidance*, 'acceso múltiple por escucha de portadora con eliminación de colisiones'). El CSMA/CA es una mejora respecto del CSMA, no respecto del CSMA/CD.

De estos cinco MAC aleatorios, sólo son "comercialmente útiles" el CSMA/CD y el CSMA/CA. De todos modos, a continuación veremos el funcionamiento de los cinco.

### *3.3.1. Algoritmo Aloha puro*

El Aloha es un algoritmo experimental que dio inicio a todo el resto. En pocas palabras, este algoritmo permite que cada terminal envíe los datos cuando tenga necesidad, sin ningún tipo de espera. Los datos llegarán a su destino si ningún otro terminal decide transmitir más o menos en el mismo momento.



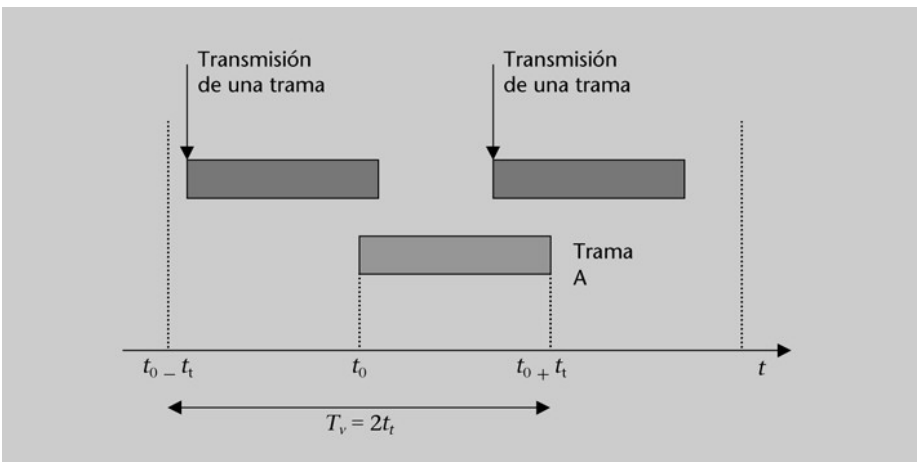

Estaciones con Aloha puro como algoritmo de acceso al medio.

Como ya hemos comentado, dos tramas transmitidas por diferentes estaciones pueden chocar. Observad que las estaciones transmiten sus tramas sin preocuparse por si el medio está ocupado o no por una trama de otra estación.

Asumimos que una estación transmite una trama. Definimos el tiempo de vulnerabilidad (*Tv*), o ventana de colisiones, como el tiempo en el que una trama es susceptible de experimentar colisiones.

Tenemos que deducir cuál es este tiempo de vulnerabilidad. La estación A transmite una trama en el instante $t_0$. La trama tiene una duración $t_t$. La estación A no sabe si alguna estación ha transmitido una trama antes de $t_0$ o la transmitirá después de $t_0$.

Esto quiere decir que si en el intervalo $[t_0 - t_t, t_0]$ hay alguna estación que ha empezado a transmitir una trama, o bien si alguna estación empieza a transmitir una trama en el intervalo $[t_0, t_0 + t_t]$, se producirá una colisión. Por lo tanto, el tiempo de vulnerabilidad valdrá: $\Delta v = (t_0 + t_t) - (t_0 - t_t) = 2t_t$.




Tiempo de vulnerabilidad con un Aloha puro.

Ahora evaluaremos el rendimiento de este algoritmo de acceso al medio. En primer lugar es necesario definir los parámetros siguientes:

- Número de estaciones (N).
- Duración de la trama ($t_t$).
- Número medio de tramas transmitidas por todas las estaciones juntas (λ).
- Número total de tramas transmitidas por todas las estaciones juntas (incluidas las tramas nuevas y las tramas retransmitidas a causa de las colisiones) por tiempo de trama (*G*). Se puede concluir que si λ es el número medio

de tramas por unidad de tiempo que transmiten todas las estaciones, se puede calcular $G$ multiplicando $\lambda$ por la duración de la trama: $G = \lambda t_t$.

- Número de tramas transmitidas con éxito (de tramas que llegan correctamente a su destino por el canal) por tiempo de trama ($S$). Este parámetro también se denomina *throughput* por tiempo de trama o rendimiento. Es un parámetro normalizado a la capacidad del canal y, por lo tanto, inferior a 1.
- Número de retransmisiones por tiempo de trama ($RT_x$).

A partir de las definiciones anteriores podemos deducir lo siguiente:

a) $0 < S < 1$ (normalizado a la capacidad del canal).

b) $G \geq S$.

c) $G = S + RT_x$ (tramas con éxito más tramas retransmitidas).

Para analizar este mecanismo de acceso al medio y averiguar el rendimiento ($S$), utilizaremos un modelo matemático que simplifique el original: asumimos que el número de estaciones es infinito y que las estaciones pueden transmitir en cualquier momento, independientemente unas de otras. Una vez hemos hecho esta suposición, la generación de tramas se puede modelar con un proceso de Poisson.

Recordemos los procesos de Poisson. Supongamos que colocamos $n$ puntos en el eje temporal $t$, en el que $t \in (-\infty,\infty)$. La probabilidad de que un número $k$ de estos puntos esté en un intervalo $T$ es la siguiente:

$$Pr[k] = (\lambda T)^k \cdot \frac{e^{-\lambda T}}{k!}$$

donde:

$$\lambda = \frac{n}{T}, \text{ con } n \to \infty \text{ y } T \to \infty$$

En nuestro caso, podemos calcular la probabilidad de que todas las estaciones transmitan $k$ tramas en un intervalo de tiempo $T$. El intervalo que nos interesa es el de vulnerabilidad, $T_v$. Si durante este intervalo se transmite más de una trama, habrá colisiones.

La probabilidad de que se transmitan $k$ tramas en el intervalo $T_v$, sabiendo que las estaciones transmiten una media $\lambda$ de tramas, es la siguiente:

$$Pr[k] = (\lambda T_v)^k \cdot \frac{e^{-\lambda T_v}}{k!} = (\lambda 2t_t)^k = \frac{e^{-\lambda 2t_t}}{k!} = (2G)^k \cdot \frac{e^{-2G}}{k!}$$

Asumimos que una estación transmite una trama en un instante arbitrario $t0$. La probabilidad de que se produzcan colisiones es $Pr[k > 0] = 1\text{-}Pr[k = 0]$. Por lo tanto, el número de retransmisiones por tiempo de trama lo darán las tramas que, entre todas las tramas transmitidas ($G$), hayan experimentado colisiones:

$$RTx = G\ Pr\ [k > 0]\ G\left(1 - Pr[k = 0\ ]\right) = G\ (1-(2G)\ \frac{e^{-2G}}{o!}\Big) = G(1-e^{-2G})$$

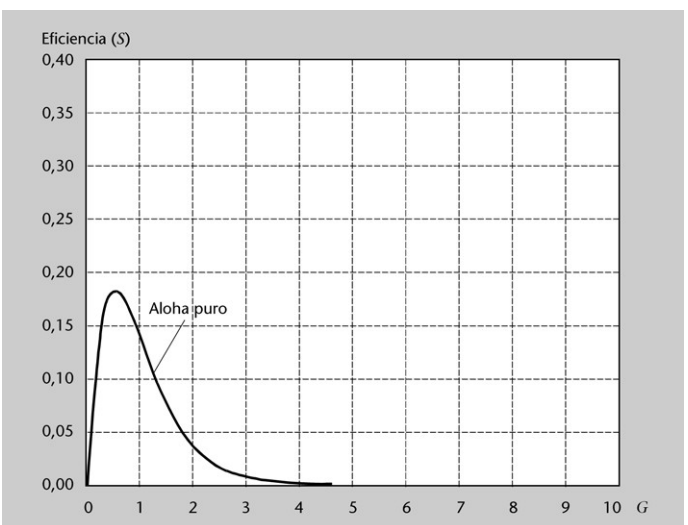
En esta fórmula ya podemos observar que la probabilidad de que no se produzcan colisiones vale $Pr[k = 0] = e^{-2G}$. Aislamos de $G = S + RT_x$ para obtener el *throughput* por tiempo de trama, $S = G - RT_x$:

$$S = G - RT_x = G - G(1 - e^{-2G}) = Ge^{-2G}$$

A continuación se muestra una gráfica de esta fórmula. $S$ tiene un máximo en $G = 0{,}5$:

$$S_{máx} = S|_{G = 0,\ 5} = 0{,}5\ e^{-1} = \frac{1}{2e} = 0{,}1839$$

Es decir, el rendimiento máximo que se puede conseguir con un algoritmo Aloha puro es de 18,39%:

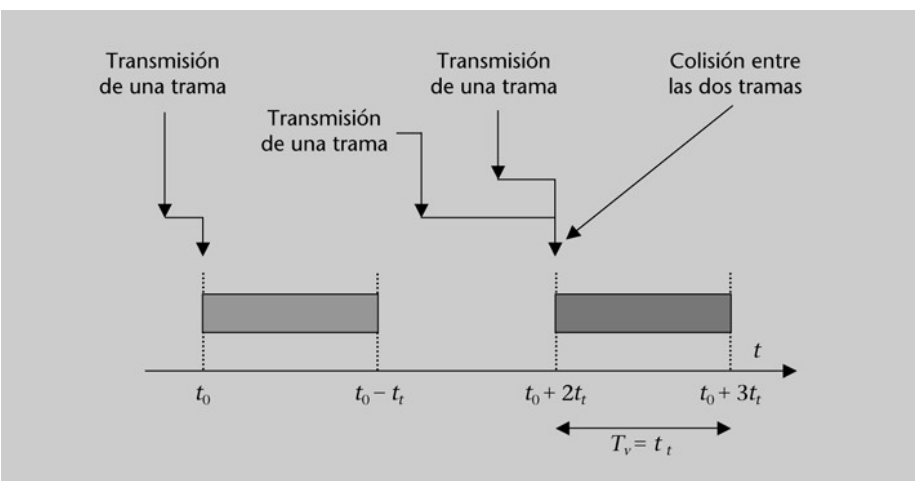


Esta fórmula permite hacer un cálculo aproximado de la eficiencia del algoritmo Aloha puro con la hipótesis de que el número de estaciones tiende a ser infinito y que las estaciones transmiten independientemente unas de otras. También permiten comprobar que el algoritmo Aloha tiene un problema de estabilidad que veremos más adelante.

Si el número de estaciones es pequeño, el modelo descrito no se cumple, y habría que utilizar otros tipos de modelos matemáticos que no estudiaremos en esta volumen. No obstante, sí que podemos deducir qué pasa si sólo hay una estación que transmite tramas. Si se trata de una sola estación no habrá colisiones. Esto quiere decir que la eficiencia se calculará como si fuera un protocolo *Stop & Wait*. El tiempo de ciclo dependerá de la longitud de la trama, del tiempo de propagación entre la estación transmisora y la receptora y de la duración del reconocimiento.

### *3.3.2. Algoritmo Aloha segmentado*

El sistema Aloha segmentado se propuso con el fin de aumentar el rendimiento del Aloha puro. En este sistema el tiempo se divide en ranuras de tiempo que tienen una longitud igual a la duración de una trama ($t_t$ segundos). Una manera de conseguir la sincronización es que una de las estaciones emita un tono al principio del intervalo de sincronización. Todas las estaciones que tienen una trama preparada para transmitir deben esperar el comienzo de este intervalo de sincronización.




Mecanismo Aloha segmentado.

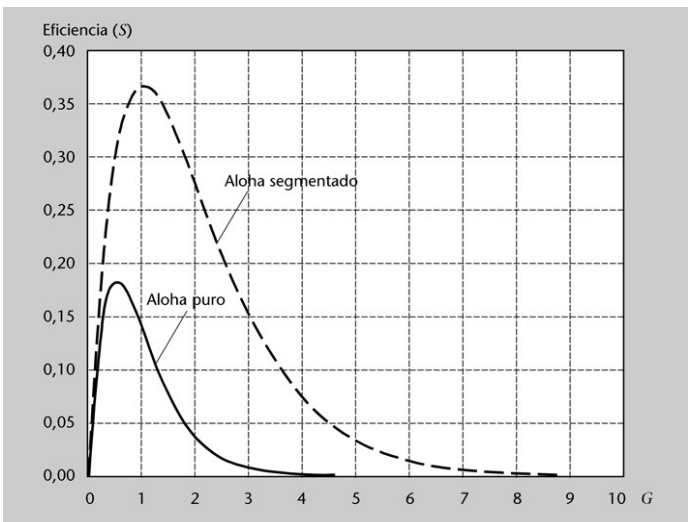
Ahora sólo se pueden producir colisiones en el tiempo de ranura, de manera que la vulnerabilidad valdrá $T_v = t_t$. Si hacemos el mismo tipo de análisis que en el algoritmo Aloha puro, podemos deducir que el *throughput* por tiempo de trama vale:

$$S = G - RT_x = G - G(1 - e^{-G}) = Ge^{-G}$$

y que tiene un máximo en $G = 1$:

$$S_{máx} = S_{G-1} = e^{-1} = \frac{1}{e} = 0{,}367837$$

Es decir, hemos aumentado (duplicado) el rendimiento, que ha pasado del 18,39% al 36,78%:




### *3.3.3. Problemas de estabilidad de los algoritmos Aloha puro y Aloha segmentado*

Si nos fijamos en la gráfica de la eficiencia, podemos observar que además de tener un bajo rendimiento, los dos mecanismos tienen un problema de estabilidad: pueden llegar a saturar el canal. Ahora estudiaremos cómo se produce este efecto.

En la gráfica de la eficiencia podéis ver que si $G$ aumenta, el *throughput* tiende a 0. Tomemos como ejemplo el algoritmo Aloha segmentado. El punto de trabajo ideal de este mecanismo sería $G = 1$ (en el algoritmo Aloha puro, $G = 0{,}5$). Si se produce una colisión, $G$ crece. La trama se tiene que retransmitir, de manera que $G$ aumen-

ta por encima de $G = 1$. Como ahora $G > 1$, nos encontramos en la zona de la curva en la que el *throughput* tiende a 0 (esta zona se denomina *zona inestable del algoritmo*). Se producen más colisiones, de manera que $G$ continúa creciendo hasta que el *throughput* llega a 0. Este efecto se denomina *saturación del canal*.

### *3.3.4. Acceso múltiple por escucha de portadora*

Uno de los motivos del bajo rendimiento de los algoritmos Aloha es que las estaciones no son conscientes de si el medio de transmisión está ocupado o no. Una manera de solucionar este problema es escuchar el medio electrónicamente y, si hay una trama en curso (hay señal en el medio; es decir, hay portadora), esperar y no transmitir. Este tipo de mecanismos se denominan mecanismos de acceso múltiple por escucha de portadora (CSMA).

Cuando una estación quiere transmitir, comprueba que no haya ninguna otra utilizando el medio. Si no hay ninguna estación transmitiendo, pone su trama; en caso contrario, hay que elaborar más el algoritmo. En este sentido, se presentan tres alternativas:

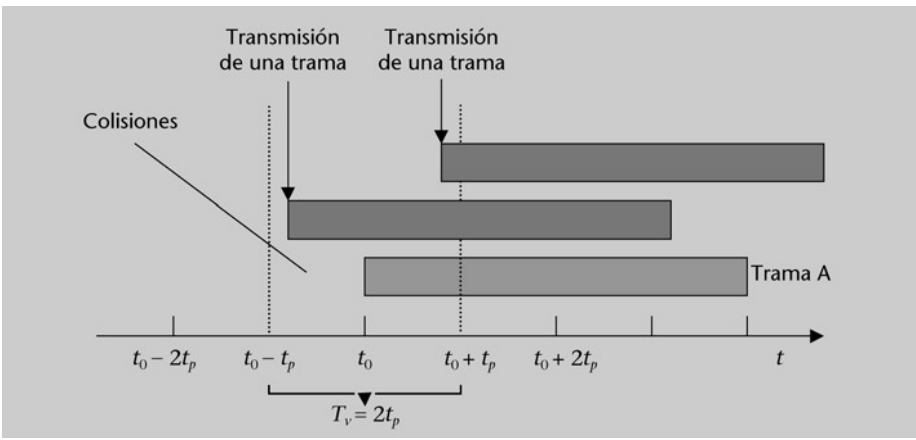
- Algoritmo 1-persistente. La estación comprueba que el medio esté libre. Si lo está, transmite inmediatamente; si está ocupado, espera hasta que esté libre y, entonces, transmite inmediatamente.
  Este algoritmo presenta la desventaja de que si dos estaciones quieren transmitir en el momento en el que una tercera ya lo está haciendo detectarán simultáneamente el medio vacío, decidirán transmitir al mismo tiempo y colisionarán. Ciertamente, esto es un inconveniente cuando hay cargas altas. Para cargas moderadas o bajas éste es el mejor algoritmo.
- Algoritmo no persistente. La estación comprueba que el medio esté libre. Si lo está, transmite inmediatamente; si está ocupado, espera un tiempo aleatorio y vuelve a intentar hacer la transmisión.
  Este algoritmo tiene la ventaja de que el tiempo aleatorio hace disminuir la probabilidad de que se produzcan colisiones. Y tiene la desventaja de que el medio de transmisión puede estar en desuso durante el tiempo aleatorio, ya que una estación puede haber dejado de transmitir y la otra todavía tiene que esperar este tiempo aleatorio. Por lo tanto, es ineficiente para cargas bajas (el canal se puede utilizar poco), mientras que para cargas altas es mejor que el algoritmo 1-persistente, ya que el número de colisiones disminuye. También puede aplicar un mecanismo de *back-off* con el fin de aliviar las colisiones.

- Algoritmo p-persistente. Si el medio está libre se transmite con probabilidad $p$. Con probabilidad $1 - p$ se espera la ranura siguiente, en la que se repite el proceso. Si el medio está ocupado, se escucha hasta que esté libre y se vuelve a poner en marcha el proceso.
  Es decir, siempre que el canal está libre se tira una moneda: con probabilidad $p$ se transmite y con probabilidad $1 - p$ se vuelve a intentar en la ranura siguiente. Si $p = 1$, tenemos un 1-persistente.

También se pueden producir colisiones aunque las estaciones se encuentren en estado de escucha del medio para detectar si está ocupado. La razón de este hecho es el tiempo de propagación que hay entre las estaciones.

Dos estaciones A y B separadas por una distancia $D$ km tardarán $t_p = D/v_p$ segundos en saber lo que hace la otra. Es decir, si la estación A empieza a transmitir, la estación B, que está escuchando el medio, oirá el primer bit de la trama de A $t_p$ segundos más tarde. Esto significa que durante $t_p$ segundos el medio está ocupado por una trama de A y la estación B lo ve libre. Si B transmite durante este tiempo, se puede producir una colisión.

Como pasaba en el algoritmo Aloha, una estación no percibe lo que ya ha pasado ni lo que pasará. Si la estación A se encuentra en estado de escucha y puede transmitir una trama, no sabe si $t_p$ segundos antes alguien ha empezado transmitir otra, ya que ve el medio libre. La figura siguiente muestra esta situación. El tiempo de vulnerabilidad ha disminuido de $2t_t$ a $2t_p$:

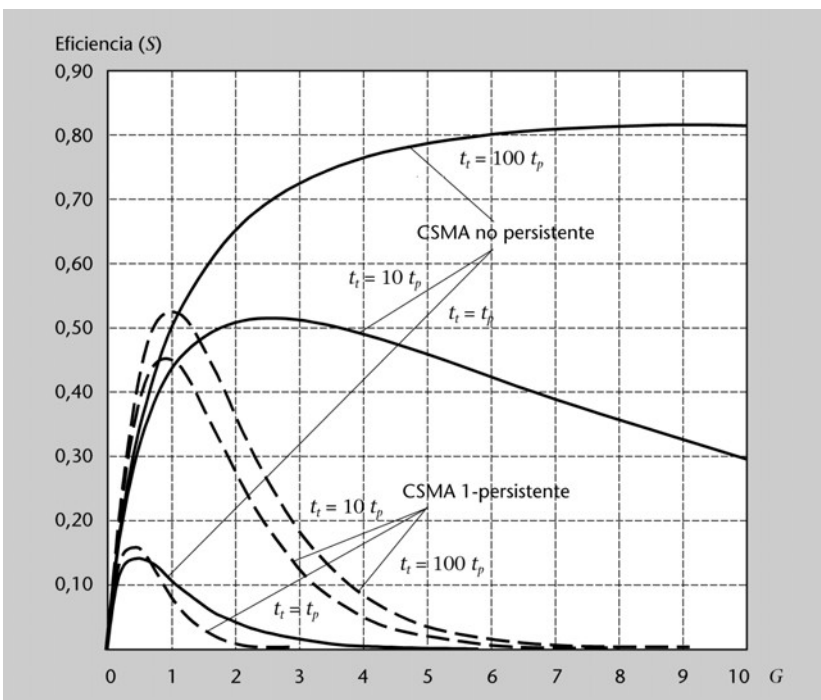



Tiempo de vulnerabilidad en un mecanismo CSMA.

El tiempo de vulnerabilidad vale $T_v = 2t_p$, donde $t_p = D_{max}/v_p$ es el tiempo de propagación determinado por la distancia entre las dos estaciones más alejadas entre sí.

Si observamos la mejora ofrecida por el CSMA respecto del Aloha, podemos ver que el algoritmo Aloha puro tenía un tiempo de vulnerabilidad de $T_v = 2t_t$ y el algoritmo Aloha segmentado, de $Tv = t_t$, mientras que el algoritmo CSMA tiene uno de $T_v = 2t_p$.

Para evaluar el rendimiento de un CSMA, se tienen que utilizar herramientas matemáticas bastante complejas. Nos limitaremos a ver una serie de gráficas en las que se pueda observar el rendimiento de los diferentes algoritmos, sin entrar en las fórmulas que nos permitan calcular estas curvas.




Eficiencia (*S*) en función del tráfico ofrecido (*G*) para unos mecanismos CSMA no persistente y CSMA 1-persistente.

La figura anterior muestra la eficiencia (*S*) en función del tráfico ofrecido (*G*) para los algoritmos CSMA no persistente y CSMA 1-persistente para diferentes valo-

res de *a*. Se puede observar lo siguiente:

- El CSMA 1-persistente obtiene rendimientos muy pequeños (< 0,2) cuando $t_t \approx t_p$.
- El CSMA 1-persistente obtiene rendimientos medios (~ 0,5) cuando $t_t \cong 10\ t_p$ o $t_t \approx 100\ t_p$.
- El CSMA no persistente obtiene rendimientos pequeños (< 0,2) cuando $t_t \approx t_p$.
- El CSMA no persistente obtiene rendimientos medios (< 0,5) cuando $t_t \cong 10t_p$.
- El CSMA no persistente obtiene rendimientos altos (> 0,8), cuando $a = 0{,}01 \rightarrow t_t \approx 100t_p$ a medida que *G* crece.

Un CSMA 1-persistente es mejor que un CSMA no persistente cuando el tráfico ofrecido (*G*) es pequeño, mientras que un CSMA 1-persistente es peor que un CSMA no persistente cuando el tráfico ofrecido (*G*) crece.

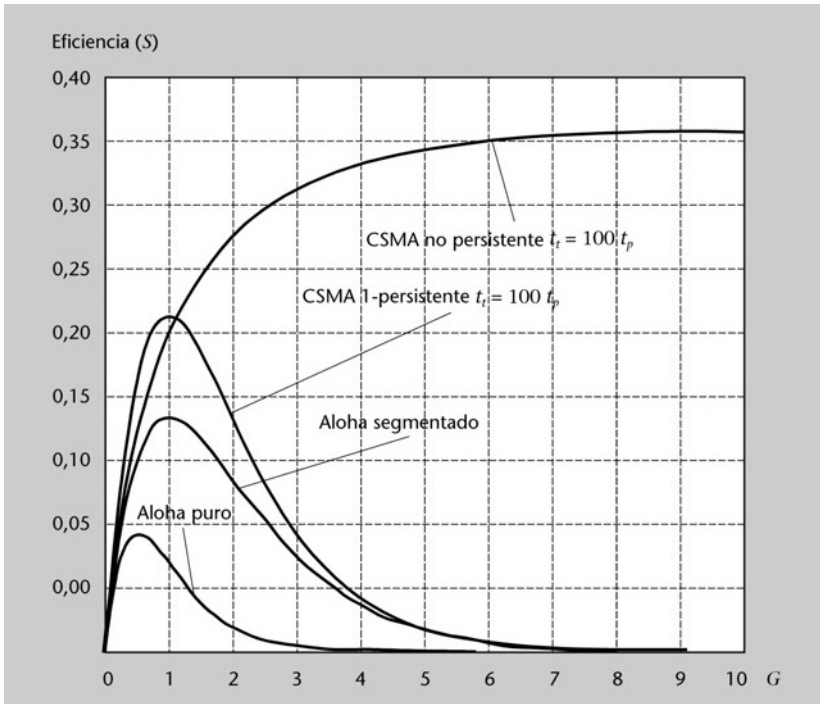
Esta conclusión es lógica si pensamos que cuando el tráfico ofrecido (*G*) aumenta, con un CSMA 1-persistente crece la posibilidad de que haya tramas que choquen en el medio; mientras que con un CSMA no persistente hay un tiempo aleatorio en el cual el medio está libre y, por lo tanto, es más difícil que las tramas choquen. En cambio, cuando el tráfico ofrecido es pequeño, resulta poco probable que haya más de una trama en el medio, de manera que con un CSMA 1-persistente aprovechamos mejor el canal que con un CSMA no persistente.

### *3.3.5. Comparación entre los algoritmos Aloha, Aloha puro y CSMA*

En la figura siguiente se comparan los algoritmos Aloha puro, Aloha segmentado, CSMA no persistente y CSMA 1-persistente (CSMA con $t_t \approx 100t_p$).

Podemos observar que el rendimiento de los algoritmos CSMA es superior al del algoritmo Aloha. La curva de rendimientos del algoritmo CSMA p-persistente depende de la probabilidad *p*: obtiene rendimientos mayores que 1-persistente a medida que *p* es menor. Incluso llega a superar el rendimiento de uno no persistente para $p = 0{,}01$.

La conclusión es que nos interesa un mecanismo que, además de reducir el tiempo de vulnerabilidad (el CSMA lo hace), reduzca también la duración de la colisión. Este mecanismo es el algoritmo CSMA/CD.




Comparación entre Aloha puro, Aloha segmentado, CSMA no persistente y CSMA 1-persistente.

### *3.3.6. Acceso múltiple por escucha de portadora con detección de colisiones*

El MAC CSMA/CD funciona exactamente como el CSMA, pero añade dos procedimientos adicionales:

- Cuando un terminal ya ha empezado a transmitir, sigue escuchando el medio con el fin de detectar otras posibles transmisiones. Esto permite implementar la detección de colisiones (CD).

  Si no hay ningún terminal transmitiendo, la detección de colisiones no señala ningún problema y, por lo tanto, la transmisión sigue su curso como en el CSMA. Si se detecta una transmisión, es decir, una colisión, el terminal transmisor deja de transmitir inmediatamente, lo cual reduce considerablemente la duración de la colisión.

  Esta situación puede desembocar en un problema. Las dos estaciones pueden haber detectado la colisión al mismo tiempo y, por lo tanto, las dos dejan de transmitir y a las dos les queda una trama pendiente. Si vuelven a trans-

mitir la trama inmediatamente, provocarán de nuevo una colisión. Este problema se soluciona con el otro procedimiento adicional.

- Después de detectar la colisión y una vez se ha detenido la transmisión, el terminal tiene que esperar un tiempo aleatorio para volver a transmitir. Este tiempo es el denominado *tiempo de back-off*, y al ser aleatorio, evitará una nueva colisión cuando las estaciones vuelvan a transmitir la trama.

> No hay que confundir el tiempo de *back-off* con la retransmisión *p*-persistente. El tiempo de *back-off* se aplica sólo después de haber transmitido una trama y haber detectado la colisión; en cambio, la transmisión *p*-persistente aplica un retardo aleatorio antes incluso de transmitir. Uno se aplica por CSMA (*p*-persistencia) y el otro, por CD *(back-off*).

Calculemos los tiempos de vulnerabilidad y el de duración de colisión. El tiempo de vulnerabilidad no varía respecto del tiempo del CSMA. El de colisión, obviamente, disminuye porque la transmisión se detiene al detectar otra transmisión en el medio. Si *tp* es el tiempo de propagación máximo entre dos terminales, la detección de la colisión y, por lo tanto, el paro de transmisión (y la duración del tiempo de colisión) será: $Tcol \leq 2tp$.

En la tabla siguiente resumimos los diferentes valores de *Tv* y *Tcol* para los mecanismos estudiados:

| Mecanismo de funcionamento | Tiempo de vulnerabilidad | Duración de las colisiones |
|---|---|---|
| Aloha puro | $T_v = 2t_t$ | $t_t \leq T_{col} \leq 2t_t$ |
| Aloha segmentado | $T_v = T_{CLK} \approx t_t$ | $T_{col} = T_{CLK} \approx tt$ |
| CSMA | $T_v = 2t_p$ | $t_t \leq T_{col} \leq t_t + 2t_p$ |
| CSMA/CD | $T_v = 2t_p$ | $T_{col} \leq 2t_p - t_t$ |

Resumen de la duración de las colisiones y el tiempo de vulnerabilidad para los mecanismos de acceso al medio aleatorios..

Finalmente, para este mecanismo el tiempo de vulnerabilidad define dos parámetros esenciales en una red:

- La longitud mínima de una trama ($L_{min}$). La trama tiene que ser lo bastante larga como para permitir detectar una colisión antes de que finalice la transmisión. Por lo tanto, la duración mínima de una trama ($t_{min} = L_{min}/v_t$) tiene que ser mayor que el tiempo de vulnerabilidad.

  Si la trama fuese menor que el tiempo de vulnerabilidad o su valor máximo, la estación transmisora no podría detectar a tiempo si se ha producido una colisión. Desde su punto de vista, ya habría acabado de transmitir la trama sin experimentar ninguna colisión, ya que el primer bit de la trama del otro terminal todavía no le habría llegado cuando ella ya había acabado de transmitirla correctamente.

- El diámetro máximo de la red. Las redes CSMA/CD tienen una limitación de alcance a causa de la ventana de colisión. El tiempo que una señal tarda en ir y volver desde una estación a otra tiene que ser menor que la ventana de colisión. Esto quiere decir que la suma de todos los retardos de la red multiplicada por dos debe ser menor que la ventana de colisiones o tiempo de vulnerabilidad:

$$2(\Sigma \text{ retardos red} + \Sigma \text{ retardos tarjeta} + \Sigma \text{ retardos cables}) \leq T_v$$

  Generalmente, se suele añadir un margen de salvaguarda a la desigualdad, de manera que ésta se convierte en:

$$2(\Sigma \text{ retardos red} + \Sigma \text{ retardos tarjeta} + \Sigma \text{ retardos cables}) + \text{margen} = T_v$$

### *3.3.7. CSMA/CD o anillos con paso de testigo*

En este subapartado intentaremos averiguar cuál de los dos estándares en los que nos hemos centrado (CSMA/CD y anillo con paso de testigo) es más eficiente.

Ésta es una cuestión difícil de responder. Desde el punto de vista técnico (de rendimientos), una red CSMA/CD es muy eficiente con cargas de tráfico pequeñas y bastante eficiente con cargas grandes. Es evidente que si la carga tiende a infinito el número de colisiones será tan grande que la eficiencia tenderá a cero.

Con una LAN en anillo con paso de testigo la eficiencia es pequeña si la carga es pequeña (si sólo hay una estación que transmite tramas, tiene que esperar a que el testigo recorra todo el anillo antes de poder transmitir otra trama). En cambio, si hay muchas estaciones y todas quieren transmitir, es una LAN muy efi-

ciente (el testigo recorre el anillo y todas las estaciones tienen la oportunidad de transmitir sus tramas). Además, el *token ring* no se muestra inestable en ninguna situación y se comporta equitativamente con todos los terminales, sea cual sea su carga.

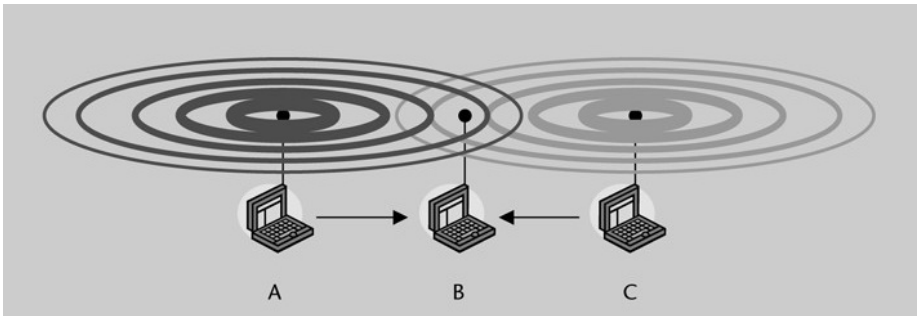
Una segunda cuestión que puede ser importante a la hora de elegir una LAN es el coste y los problemas de instalación que puede tener un ingeniero (que dependerán de la experiencia que tenga con este tipo de LAN). De todos modos, hay que evidenciar que en estos momentos las redes Ethernet, basadas en CSAM/CD, son el estándar *de facto* en el campo de las LAN, dominando absolutamente el mercado. Eso se debe al hecho de que las tecnologías actuales basadas en conmutadores (*switch*) incluso han eliminado que se puedan producir colisiones, cosa que ha permitido un enorme incremento en su rendimiento y alcanzar velocidades de transmisión del orden de Gigabits.

## 3.4. Acceso múltiple por escucha de portadora con eliminación de colisiones

El CSMA/CA es un paso adelante sobre el CSMA (no sobre el CSMA/CD) porque mejora el tiempo de colisión. El CSMA/CA se utiliza en redes en que no es posible la detección de colisiones, como las redes inalámbricas.

Los radiotransmisores *full duplex* sólo se pueden construir si la ida trabaja en una banda de frecuencia distinta de la de vuelta. Esto se debe al hecho de que la potencia recibida es muy inferior a la señal que se transmite, y sólo puede discriminarse con procedimientos discriminadores frecuenciales (filtros).
En el caso de una LAN no podemos pensar en trabajar con frecuencias distintas, ya que todos los terminales requerirían una y esto derrocharía el espectro. Estaríamos hablando del FDMA.

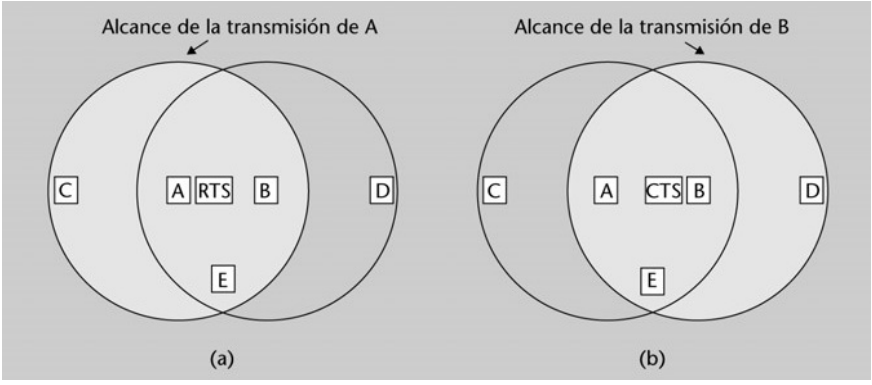
Los participantes de una LAN inalámbrica pueden escucharse parcialmente; es decir, cada terminal quizá recibe sólo una parte de los terminales, porque el resto los tiene demasiado lejos. Es lo que se conoce como *problema del "terminal oculto"*. Incluso si se pudiera implementar la detección de colisiones (como en el CSMA/CD), no se resolvería este problema: un bus CSMA/CD garantiza que todo el mundo escucha a todo el mundo, pero una LAN inalámbrica no puede ofrecer esta garantía.




El problema del "terminal oculto": el terminal A cree que el medio está libre aunque C transmita. El destinatario B sólo ve el resultado de una colisión.

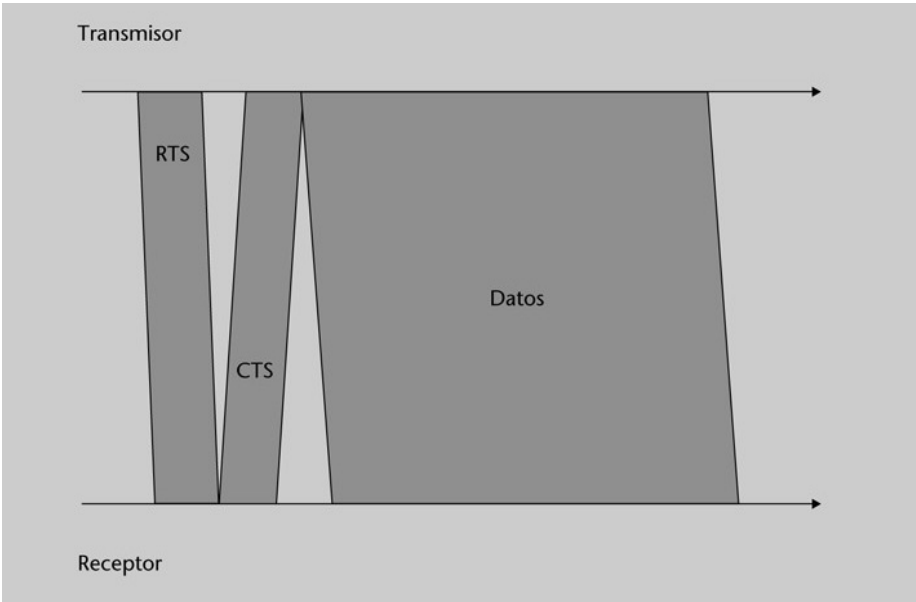
El CSMA/CA intenta evitar este problema con la ayuda del terminal receptor, que debe colaborar para detectar si el medio está ocupado: el CSMA/CA, antes de transmitir la trama, provoca que el terminal emisor envíe una pequeña trama denominada RTS (*Request To Send*, 'quiero enviar') dirigida al destinatario. El destinatario tiene que responder con el correspondiente CTS (*Clear To Send*, 'preparado para recibir').

Después de hacer estos pasos, los terminales al alcance del transmisor (terminal A en la figura) y los terminales al alcance del receptor (terminal B) saben que habrá una transmisión:




Para que estas tramas sean efectivas, es necesario que el RTS notifique la medida de la trama de datos que se enviará, de manera que todos los terminales de la LAN sepan el rato que la línea, en este caso el aire, estará ocupada.

El diagrama de tiempo resultante sería el siguiente:




Con esto se consigue que las colisiones se concentren en el periodo ocupado por las dos tramas RTS y CTS. El tiempo de vulnerabilidad será, como en el CSMA (y el CSMA/CD), si no se da el problema del "terminal oculto", $T_v \approx 2t_p$, ya que las condiciones son las mismas.

Si se da el problema del "terminal oculto", los cálculos pierden sentido, porque $t_p$ no sería un valor homogéneo en toda la red, sino que dependería del par de terminales que se considerasen.

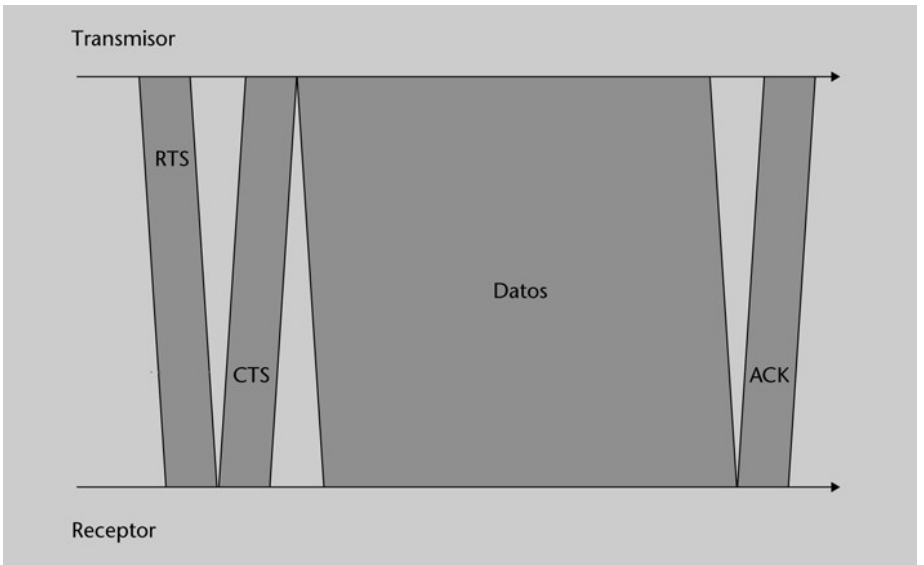
El tiempo de colisión mejora apreciablemente respecto del CSMA, porque su duración se circunscribe también a las tramas RTS y CTS, ya que durante la transmisión de la trama de datos no puede haber colisión (CA). Por lo tanto, $T_{col} \approx t_{RTS} + 2t_p$: mucho mejor que en el CSMA y similar a lo que consigue el CSMA/CD (si suponemos que $t_{RTS}$ es pequeño).

Con el CSMA/CA se consigue una duración de colisión y de vulnerabilidad similares a las conseguidas con el CSMA/CD (con el diferencial comentado de la longitud de las tramas RTS y CTS), pero a costa de que todas las tramas de informa-

ción vayan acompañadas de tramas de control adicionales que con el CSMA/CD no son necesarias; por lo tanto, la eficiencia de transmisión baja. En algunas situaciones, esto puede ser un problema.

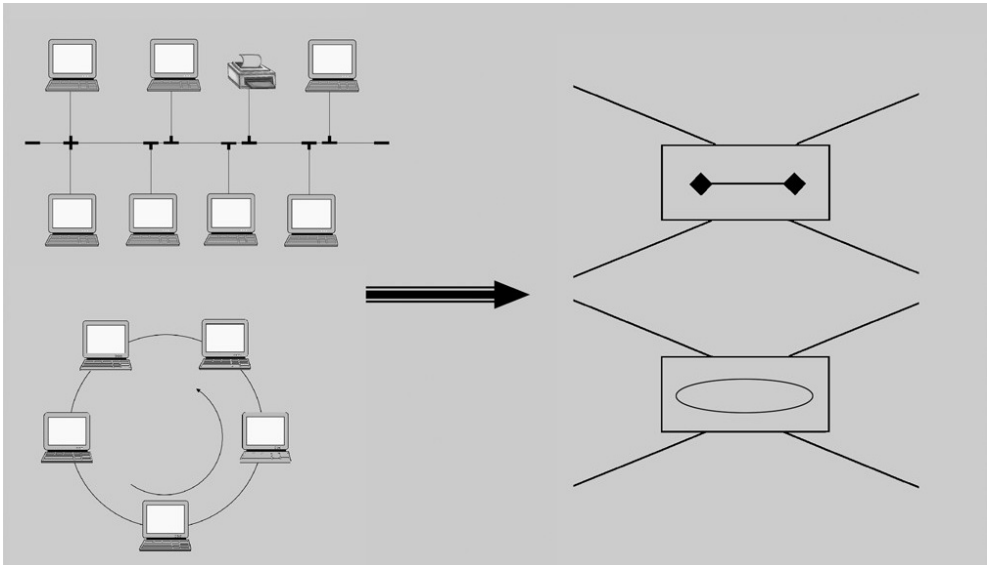
Cuando las tramas de información son pequeñas, algunas implementaciones degradan el MAC de CSMA/CA a CSMA, ya que la duración de la colisión será similar, pero en lugar de tener que transmitir tres tramas, se transmite sólo una.

A causa de la alta probabilidad de error de bit, las implementaciones reales CSMA/CA a menudo añaden una cuarta trama adicional con el fin de asegurar que el receptor ha recibido la trama. Se trata de la trama de confirmación o ACK:

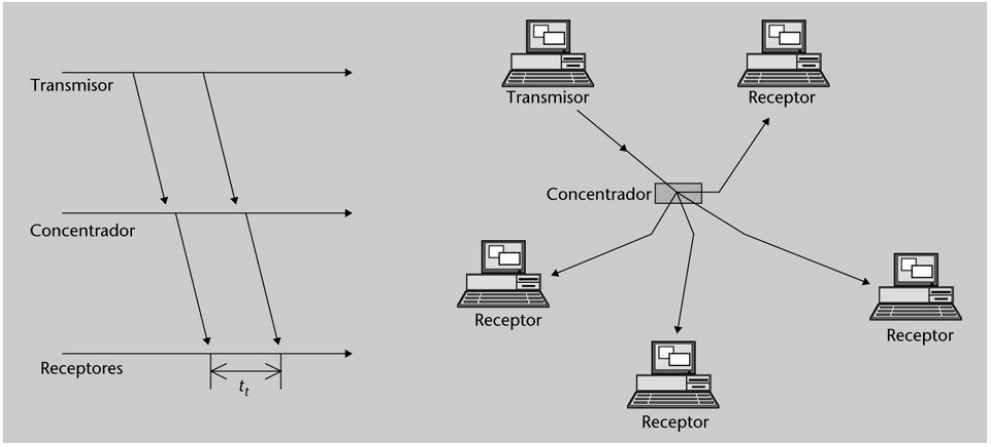


## 4. Commutación en las LAN: dominios de colisión

Recordemos la red que veíamos antes, donde teníamos una LAN con topología lógica en bus (o anillo) y topología física en estrella.



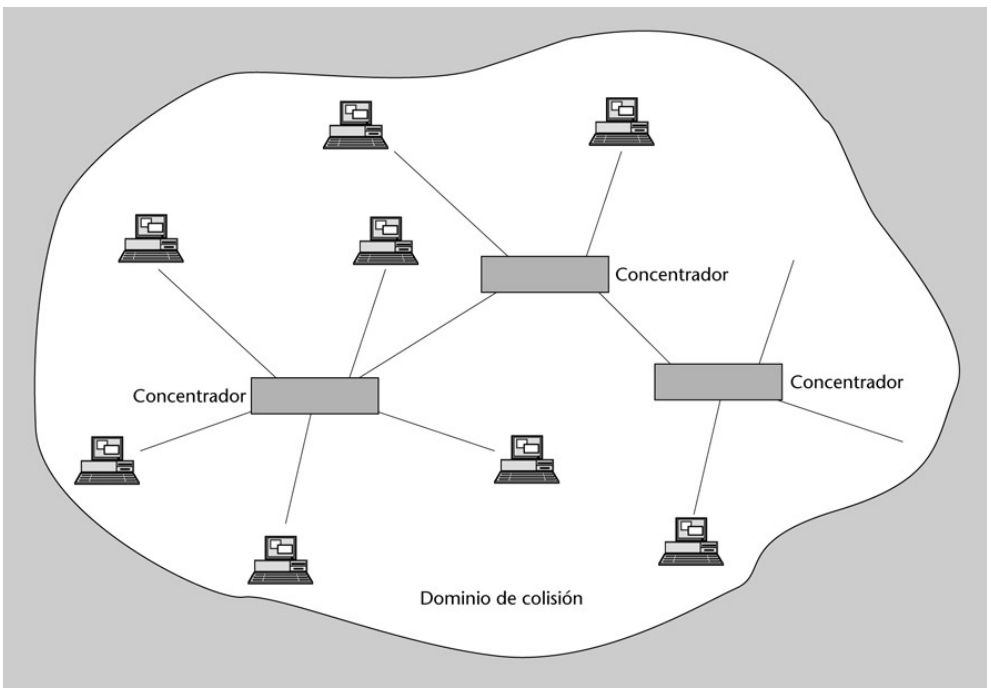
En el caso de la red en bus, el elemento que permite la interconexión es un concentrador (o *hub*), que se comporta como un simple repetidor: todo lo que le llega desde cualquier puerto lo retransmite a todo el resto.



Todos los terminales deben implementar el CSMA/CD. A efectos de estudio, la presencia del concentrador es anecdótica, y podríamos suponerlo invisible. La señal se difunde por el medio hasta llegar a todos los terminales.

En el diagrama de tiempo se ha reflejado un pequeño retardo de la señal a su paso por el concentrador. Este retardo se debe al hecho de que el concentrador es un repetidor y, por lo tanto, debe regenerar la temporización de los bits. Esto implica un cierto retardo de unos pocos bits.

Concentrador, terminales y todo el cableado están dentro de un mismo medio. Es lo que denominamos *dominio de colisión*. Un dominio de colisión es un ámbito dentro del cual un único MAC gobierna las transmisiones. Si dos terminales inician la comunicación simultáneamente en un mismo dominio de colisión, habrá colisión. En un dominio de colisión 10BaseT, ningún terminal ni ninguna combinación de terminales pueden llegar nunca a transmitir en total más de 10 Mbps.




Por lo tanto, si ampliamos una red aumentando el número de terminales, incluso si añadimos más concentradores, seguiremos teniendo un solo dominio

de colisión y, por lo tanto, cada vez la capacidad disponible para cada terminal individual será menor.

Sólo hay dos maneras de aumentar la capacidad de una red:

- Cambiar el estándar por otro a mayor velocidad. Por ejemplo, pasar de 10Base-T a 100Base-T[3] o 1000Base-T. Esta solución es bastante drástica. Pensad que no sólo se tienen que cambiar los concentradores (y quizá el cableado) sino también todas las tarjetas de los terminales. Pasar de 10 Mbps a 100 Mbps es caro.
- Segmentar el dominio de colisión de manera que el número de terminales por dominio de colisión sea inferior. Esta solución a menudo es más sencilla. Para dividir o interconectar dominios de colisión necesitamos un conmutador. Los conmutadores pueden tener ocho, dieciséis o veinticuatro puertos y, por lo tanto, desde este punto de vista son similares a los concentradores. Sin embargo, a diferencia de éstos, los puertos de un conmutador tratan la información individualmente: cada trama que reciben se almacena dentro de la memoria del puerto (o la memoria asignada al puerto), se analizan los posibles errores y se interpreta para decidir dónde se dirige.

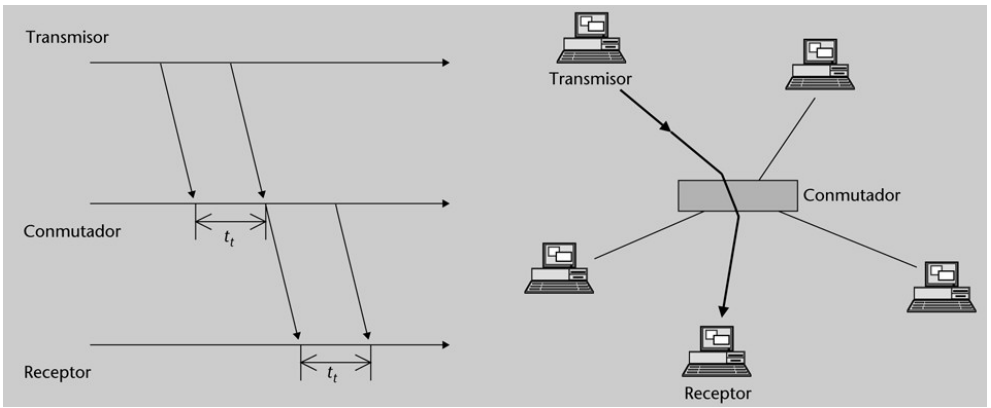
Al segmentar la red, dado que cada puerto recibe la trama, la almacena y se retransmite por el puerto de salida aplicando de nuevo el CSMA/CD, estamos hablando de dominios de colisión diferentes. Cada puerto de un conmutador conecta dominios de colisión distintos.

Los puentes (*bridges*) son una solución similar. Funcionalmente son idénticos a los conmutadores, pero técnicamente se basan en arquitecturas totalmente distintas y se han utilizado en otras épocas. En la actualidad se utilizan conmutadores.

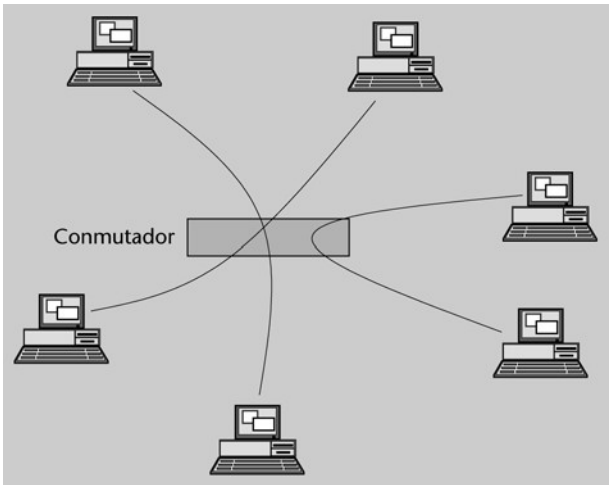
Un concentrador no lleva a cabo ninguna de estas operaciones, la trama como tal le es irrelevante: cuando recibe los primeros bits los copia inmediatamente ("conecta" sería una palabra más gráfica) al resto de los puertos. El concentrador no detecta errores ni interpreta ni nada; desde el punto de vista lógico, un concentrador es un sistema pasivo.

3. 100Base-Tx y 10Base-Tx utilizan sólo un par de ida y otro de vuelta.

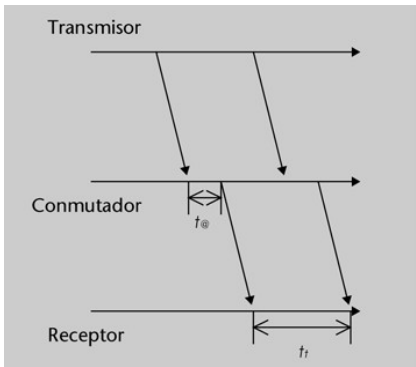
Sin embargo, hay que reconocer al menos una ventaja a los concentradores sobre los conmutadores. Para que un conmutador pueda decidir qué hace con las tramas entrantes, es necesario que almacene las tramas recibidas, aunque sólo sea momentáneamente, para poder procesar el contenido (básicamente la dirección de destino y el CRC). Este almacenamiento, como se refleja en el diagrama de tiempo siguiente, comporta un retardo de transmisión:




El retardo de transmisión es, aproximadamente, el doble de lo que comporta el uso de concentradores (si suponemos retardos de propagación negligibles). Como contrapartida, el conmutador deja los $n - 2$ terminales restantes con el puerto libre para poder transmitir: un conmutador puede mantener, teóricamente, $n/2$ transmisiones simultáneas de tramas entre todos los puertos:

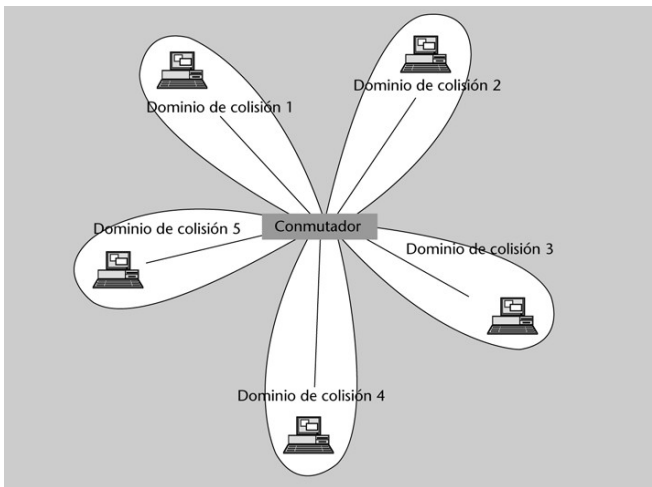


Hay conmutadores que encaminan la trama antes incluso de que haya sido recibida completamente. Son los conmutadores rápidos o *cut-through*, que en cuanto reciben el campo de dirección (en el byte 14) deciden el puerto de salida y empiezan a transmitir la trama. El retardo es muy inferior, pero en caso de que la trama sea errónea la transmitirán igualmente, ya que no lo pueden saber.

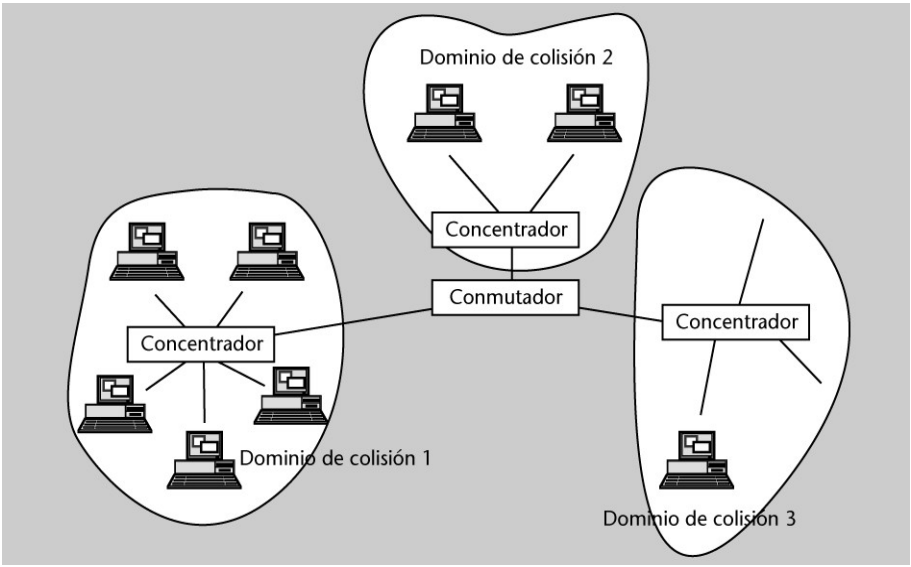



Los conmutadores rápidos presentan un problema adicional: si hay una colisión en el dominio origen se propaga al dominio destino. Para solucionar esta eventualidad, hay una variante de conmutadores (*Run-free cut-through*) que espera la ventana de colisión (64 bytes) para asegurarse de que no hay colisión en el dominio origen.

En la figura anterior, cada terminal tiene su propio dominio de colisión. De hecho, las colisiones todavía son posibles, ya que se puede dar el caso de que el puerto del conmutador y el terminal asociado decidan casualmente empezar a transmitir de manera simultánea.



Si en lugar de conectar directamente al conmutador los terminales conectamos concentradores, el sistema también mejora en cuanto a rendimiento respecto de uno conectado únicamente con concentradores:




En cualquier caso, cada dominio de colisión adicional supone un incremento de la capacidad agregada de la red si las tramas van mayoritariamente dirigidas a estaciones situadas en el mismo dominio de colisión que la estación origen. El conmutador filtra estas tramas de manera que no ocupen innecesariamente el resto de los dominios de colisión. Teóricamente, si segmentáramos el dominio original perfectamente, el conmutador no retransmitiría ninguna trama, con lo cual la capacidad de la red se habría multiplicado directamente por el número de nuevos dominios de colisión.

Los conmutadores conocen la posición de los terminales por un procedimiento de aprendizaje automático. Mientras este procedimiento no haya finalizado (por ejemplo, cuando el conmutador arranca), el conmutador reparte las tramas por todos los puertos: puesto que no conoce todavía la posición de las estaciones, actúa de manera similar a un concentrador. Cuando reciba las primeras tramas descubrirá, mediante la dirección origen, la posición de todas las estaciones.

En realidad, cuando añadamos un conmutador, la segmentación nunca será tan drástica; por lo tanto, el conmutador no filtra todo el tráfico. Consecuentemente,

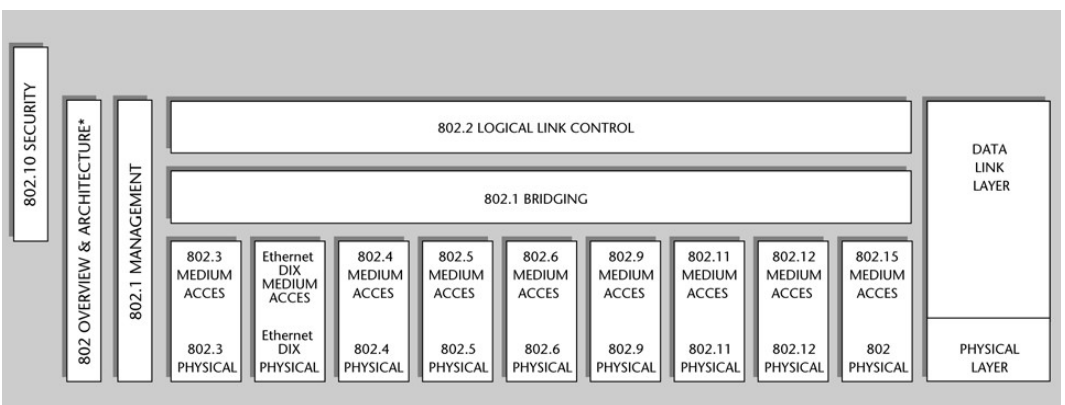
cada dominio soporta su tráfico interno, más un porcentaje del tráfico de los otros dominios. En general, si el tráfico no es interno, ocupará sólo dos dominios de colisión; por lo tanto, el resto de los $n - 2$ dominios estará libre para transmitir. En el peor de los casos, todo el tráfico ocupará dos dominios y, por lo tanto, la inserción de un conmutador aumenta la capacidad de la LAN agregada entre $n/2$ y $n$ veces, donde $n$ es el número de puertos. Y esto suponiendo que se cambia un concentrador por un conmutador. Si hay más conmutadores, entonces una sola trama puede llegar a ocupar tres o más dominios de colisión.

Los conmutadores se pueden interconectar como si fueran concentradores para hacer redes mayores. De hecho, puesto que cada dominio de colisión tiene un MAC de forma independiente, el alcance máximo de una red con conmutadores es teóricamente ilimitado.

## 5. Estándares para redes de área local

Casi todos los estándares de LAN han sido creados por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) de EE.UU., dentro de los grupos IEEE802.xx, y después los estándares han sido acogidos por la Organización Internacional de Estandarización (ISO), que ha denominado los grupos idénticamente pero utilizando el prefijo ISO8802.xx.

Todos los estándares IEEE802, a excepción del IEEE802.1 y el IEEE802.2, se refieren a protocolos MAC.



- IEEE802.1: define procedimientos para interconexión y gestión de conmutadores, protocolos para LAN virtuales y seguridad criptográfica de la transmisión.
- IEEE802.2: define el nivel de enlace lógico adecuado para LAN.
- Ethernet-DIX-II: estándar *de facto* de CSMA/CD en 10 Mbps. Fast Ethernet (100 Mbps) y Gigabit Ethernet (1 Gbps) son variantes del estándar ampliamente utilizadas. Para todas es común el cambio del MAC CSMA/CD por conmutadores.
- IEEE802.3: MAC CSMA/CD. Similar pero no interoperable con Ethernet-DIX-II.
- IEEE802.4: MAC *token bus*. Implementa un MAC equivalente a *token ring*, pero sobre topología física y lógica de bus. Actualmente se encuentra en desuso.
- IEEE802.5: MAC *token ring* derivado del protocolo Token Ring de IBM y compatible con el mismo.
- IEEE802.6: MAC DQDB o *Distributed Queue Dual Bus*. Está diseñado para redes de área metropolitana. Aunque se trata de un bus lógico, físicamente es un anillo, lo que le confiere tolerancia a cortes de la línea. Actualmente se encuentra en desuso.

Las redes de área metropolitana (MAN, Metropolitan Area Networks) eran un concepto relevante a principios de los noventa, cuando faltaban estándares para redes cuyo alcance estuviese entre el de una LAN y el de una WAN. En la actualidad, el concepto ha quedado marginado, porque en las LAN ya se utilizan tecnologías de WAN como la conmutación; por lo tanto, el espacio teórico de las MAN ha quedado pinzado.

- ANSI X3T12 o FDDI (*Fibre Distributed Data Interface*): MAC *token ring* doble con propiedades de tolerancia a fallos. Inicialmente estaba diseñado para redes MAN, pero tuvo un nicho de mercado en redes de servidor (*clusters*). Actualmente se encuentra en desuso, pero todavía pueden encontrarse redes con este estándar en funcionamiento.
- IEEE802.11 o WiFi (*Wireless Fidelity*): MAC CSMA/CA para redes inalámbricas. Actualmente es el estándar de LAN de mayor progresión.
- IEEE802.12 o 100VG-AnyLAN: estándar a 100 Mbps aparecido en los noventa como sustitución del Ethernet a 10 Mbps. Es un MAC *token ring* con prioridades y procedimientos para facilitar la interoperabilidad con

IEEE802.3. Actualmente, se encuentra en desuso a causa del éxito comercial de Fast Ethernet.

En los siguientes apartados vamos a detallar los estándares más relevantes:

## 5.1. Ethernet y IEEE802.3

A mediados de los setenta, Xerox lanzó Ethernet tal y como se conoce en la actualidad, fruto del trabajo de investigación de Norm Abramson, de la Universidad de Hawai. Abramson diseñó y construyó en 1971 una red vía satélite que utilizaba el protocolo Aloha, diseñado a este efecto. Era la red AlohaNet.

Para competir con la Token Ring de IBM, práctica dominadora del concepto LAN, Xerox unió esfuerzos con DEC (Digital Equipment Corporation) y con Intel para comercializar su Ethernet.

Problemas de encaje en el modelo LAN de IEEE802 (que apareció posteriormente) hacen que la versión estándar (*de iure*) IEEE802.3 no sea exactamente la misma que la de Ethernet. Puesto que, en general, utilizamos la palabra *Ethernet* para referirnos a las dos variantes, para distinguir claramente cuándo nos referimos a la Ethernet diferente de IEEE802.3, le llamaremos *Ethernet-DIX* (Digital, Intel, Xerox).

### *5.1.1. Nivel físico*

Ethernet utiliza diferentes tipos de medio físico y a diferentes velocidades. Para diferenciar los estándares, se les denomina según el patrón siguiente:

<Velocidad en Mbps><Tipo de codificación><Tipo de medio de transmisión>

- En cuanto a velocidad, ahora se puede encontrar en el mercado de hardware a 10, 100 y 1000 Mbps o incluso a 10 Gbps y ya con vistas a 100 Gb.
- Puesto que todas las codificaciones actuales son banda base, la parte central siempre es “Base”.
- La parte final acostumbra a ser “T” (de par trenzado) o “F” (de fibra óptica), más un sufijo adicional para distinguir las variantes.

Los estándares más utilizados son:

- 10Base-2 (IEEE802.3a): 10 Mbps sobre coaxial (el 2 viene del hecho de que el alcance aproximado sin repetidores es de 200 m). La codificación es Manchester. Topología coaxial. Se permiten cuatro repetidores separando cinco tramos de 200 m.
- 100Base-Tx (IEEE802.3u): 100 Mbps sobre par trenzado categoría 5. Codificación NRZ con precodificación (para garantizar el sincronismo de bit) 4B/5B (125 Mbaud). Topología en estrella. Se permite la interconexión de concentradores si están a menos de dos metros de distancia o bien si están apilados.
- 1000Base-T (IEEE802.3ab): 1 Gbps sobre par trenzado categoría 5. Codificación de cinco niveles que empaqueta dos bits por símbolo en paralelo sobre los cuatro pares en paralelo, y que resulta en 125 Mbaud en cada par, lo mismo que en 100Base-T. Topología en estrella. Los concentradores sólo se pueden interconectar apilándolos.

Todos los estándares en uso tienen topología en estrella. Además, todos funcionan sobre par trenzado. Hay también especificaciones para uso de fibra óptica, pero no las trataremos.

### *5.1.2. Control de acceso al medio*

El control de acceso al medio es CSMA/CD 1-persistente. La ventana de colisión, o tiempo máximo de vulnerabilidad, está definida en 512 bits. Por lo tanto, el alcance máximo teórico es 51,2 μs a 10 Mbps y 5,12 μs a 100 Mbps. A 1 Gbps se ha modificado este aspecto para permitir un alcance más razonable: 512 bytes, es decir, 4.096 bits; por lo tanto, la ventana de colisión es de 4,096 μs. Esta modificación comporta que el alcance máximo teórico sea de 5 km, 500 m y 400 m respectivamente. A causa de factores como la atenuación, interferencias, etc. Ninguno de los estándares anteriores llega al límite teórico.

Es necesario detallar la 1-persistencia un poco más. Cuando un terminal espera que el medio quede libre, tiene que dejar un rato de silencio para garantizar que todo el mundo percibe la finalización de la trama anterior. Es el espacio intertrama[4], que vale $t_{IFG}$ = 96 bits. El resto de los tiempos utilizados en Ethernet están definidos en términos de ventana de colisión.

4. En inglés, *Inter Frame Gap* (IFG).

La detección de colisiones también tiene un parámetro que hay que definir, la función de tiempo de *back-off*. Cuando hay una colisión, los terminales transmisores tienen que esperar un IFG y, después, decidir aleatoriamente si esperan1 o 0 ventanas de colisión:

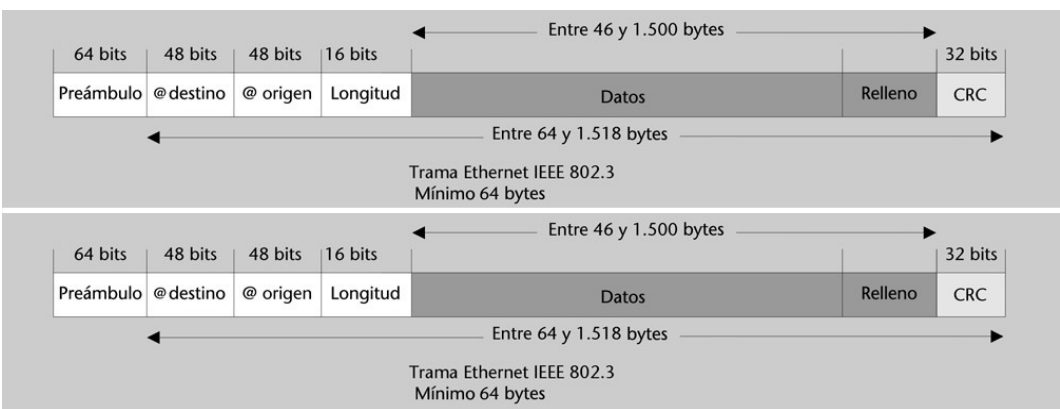
- Si un terminal elige 1 y el otro 0, hay uno que gana la contención y el otro tendrá que volver a esperar.
- Si los dos terminales (o más, si es el caso) eligen 1 o 0 (50% de probabilidad), vuelve a producirse una colisión.

En este segundo caso, el tiempo de *back-off* máximo se dobla, de manera que los terminales aleatoriamente esperan 0, 1, 2 o 3 ventanas de colisión. Si volviera a haber una colisión, el tiempo de *back-off* iría creciendo hasta valer$2^{10}$ –1 ventanas de colisión a la décima colisión. Este valor se mantiene hasta la decimosexta colisión, momento en el cual el *driver* de red tiene que notificar a la aplicación que hay problemas.

Este hecho sería altamente improbable, a no ser que hubiera algún problema en el cableado que provocase ecos, como un corte o una mala conexión: el eco se monitoriza como una colisión, que sistemáticamente se repite.

### *5.1.3. Formato de trama Ethernet*

Tal y como hemos dicho, hay dos estándares Ethernet, y precisamente la diferencia principal radica en el formato de trama. En ambos casos es minimalista; no hay prácticamente nada superfluo:



Veamos los campos comunes a los dos estándares:

- El preámbulo consta de 62 bits, alternativamente 1 y 0. Esto permite al receptor sincronizar el reloj de su receptor de manera precisa. Debemos tener en cuenta que antes de una trama siempre hay silencio (el IFG) y, por lo tanto, el receptor necesita "entrenar" el reloj de recepción (sincronismo de bit). Los dos últimos bits (para llegar a 8 bytes) son dos 1 seguidos. Esto permite al receptor perder algunos bits del preámbulo y mantener todavía el sincronismo de trama. Precisamente ésta es la función del preámbulo.
- La dirección de destino. Cada estación tiene su propia dirección, y cuando detecta una trama en el medio, simplemente tiene que vigilar si lleva su dirección. Si no, puede ignorar el resto de los campos.
- La dirección origen identifica la estación autora de la trama.
- El campo CRC contiene un CRC-32.

Ahora veremos los campos diferenciales:

El cuarto campo en Ethernet-DIX es el tipo. Este campo es la única concesión en el nivel de enlace en Ethernet-DIX: permite multiplexar conexiones. A IEEE 802.3 no le es necesario, ya que obliga a utilizar IEEE 802.2 por encima (un protocolo de enlace completo). En cambio, IEEE 802.3 utiliza el cuarto campo (también de 2 bytes) para definir la longitud del contenido de datos dentro de la trama. Hemos de tener en cuenta que la longitud mínima de la trama es de 512 bits (64 bytes); si restamos todos los campos que hemos visto, la trama debe contener: 64 B – 8 B – 6 B – 6 B – 2 B – 4 B = 38 B. Si la trama contiene menos de 38 B –el campo de longitud así lo indica–, añadimos los bytes de relleno necesarios (campo de *padding "relleno"*) para conseguir los 64 bytes totales mínimos.

Ethernet no resuelve este problema. Simplemente fuerza a la aplicación a generar tramas de 38 o más bytes de contenido. Si no es así, no las envía y devuelve un error.

Esto no es problema si Ethernet transporta TCP/IP. La cabecera IP son 20 bytes, y la TCP, 20 más. Cualquier paquete TCP/IP tiene, de este modo, 40 bytes o más.

Ahora se nos plantea una pregunta que ya hemos resuelto en parte: ¿puede coexistir tráfico Ethernet-DIX e IEEE802.3 dentro de una misma red?

Se puede encontrar una lista actualizada de tipos de Ethernet reconocidos y de fabricantes registrados en la dirección siguiente: http://www.iana.org/assignments/ethernetnumbers.

Desde el punto de vista físico ya hemos visto que pueden coexistir. Desde el punto de vista de la trama, hay un campo que es distinto en los dos estándares: tipo y longitud. Afortunadamente, la longitud de tramas, tanto Ethernet-DIX como IEEE802.3, está limitada a 1.500 bytes. Ya que los tipos Ethernet-DIX están estandarizados, simplemente hay que procurar no asignar tipos con un identificador inferior a 1.500. Esta corrección se hizo al estándar Ethernet-DIX cuando apareció el IEEE802.3. Es lo que se denomina ***Ethernet-DIX-II.*** Cuando hablamos de Ethernet-DIX nos referimos siempre a Ethernet-DIX-II.

### *5.1.4. Direcciones*

Las direcciones en Ethernet y en todos los estándares IEEE802 se definen igual. Son 48 bits que identifican inequívocamente al terminal cualquier estación Ethernet, IEEE802.3 o de cualquier otro tipo de LAN, que tiene una dirección única. Esto se consigue repartiendo un identificador de 22 bits a cada fabricante, que lo utiliza como prefijo para las direcciones de las tarjetas de red que fabrica. La dirección final tiene un formato como éste:

| 2 bits | 22 bits | 24 bits |
|---|---|---|
| "00" | Identificador de Organización | Identificador de tarjeta |

Por lo tanto, cada fabricante tiene un límite de $2^{24}$ direcciones: 16 millones de tarjetas. Los fabricantes que superan esta cifra reciben un nuevo identificador.

Los dos bits iniciales son siempre 0 para direcciones únicas. Sin embargo, en ciertos casos se utilizan tipos especiales. Si el primer bit es un 1, el formato cambia: los 47 bits restantes se utilizan para identificar grupos de terminales:

| 1 bits | 47 bits |
|---|---|
| "1" | Identificador de grupo |

Las direcciones de grupo o direcciones ***multicast*** se utilizan sólo como dirección de destino (nunca como dirección origen) para enviar la trama a todos los que pertenecen al grupo. El grupo con toda la dirección en "1" es el grupo ***broadcast***. Todas las estaciones pertenecen al grupo ***broadcast***.

Hay un tercer tipo de dirección, que son las direcciones localmente definidas y que empiezan por "01". Esta prestación no se utiliza demasiado, y muchas tarjetas no la soportan. Además, muchas tarjetas permiten cambiar los 46 bits de la dirección de estación.

## 5.2. Fast Ethernet y Gigabit Ethernet

Fast Ethernet es una tecnología de redes de área local de alta velocidad que ofrece un ancho de banda diez vece mayor que Ethernet, tanto al usuario final como a los servidores de una red de área local. Así pues, Fast Ethernet permite transmitir a velocidades de 100 Mbps.

El grupo IEEE Higher Speed Ethernet Study Group se formó para estudiar las posibilidades que había de transmitir a velocidades de 100 Mbps utilizando la tecnología Ethernet. Este grupo estableció diferentes objetivos para la nueva Ethernet de alta velocidad, pero surgieron divergencias respecto de la manera de acceder al medio físico. El debate se fundamentó en el hecho de que esta Ethernet de alta velocidad pudiera continuar soportando el CSMA/CD como protocolo de acceso al medio. Por este motivo, el grupo de estudio se dividió en dos grupos separados: la alianza Fast Ethernet y el Foro 100VG-AnyLAN. Cada grupo produjo una especificación para trabajar con Ethernet a altas velocidades:

- La alianza Fast Ethernet produjo 100BaseT, la especificación IEEE para la implementación de Ethernet a 100 Mbps sobre par trenzado no apantallado (UTP) y par trenzado apantallado (STP). La capa de control de acceso al medio es compatible con la especificación IEEE 802.3.Por lo tanto, se mantiene el protocolo CSMA/CD y únicamente se modifica la velocidad a 100 Mbps; aunque, evidentemente, este hecho implica dividir por diez la longitud máxima de red para poder conservar el mismo formato de trama.
- 100VG-AnyLAN es una especificación IEEE para las implementaciones de redes Ethernet y Token Ring sobre cuatro pares trenzados no apantallados.

La capa de acceso al medio no es compatible con la especificación IEEE 802.3. Lo que se hace es crear un nuevo protocolo de acceso al medio físico sin colisiones y más eficiente, pero manteniendo la misma estructura de trama.

Fast Ethernet se estandarizó para IEEE mediante la especificación IEEE802.3u. En cambio, 100VG-AnyLAN se estandarizó para IEEE mediante la especificación IEEE 802.12.

### *5.2.1. Redes Fast Ethernet 100BaseT*

Como hemos comentado anteriormente, 100BaseT utiliza la especificación IEEE802.3 sobre el CSMA/CD que ya utilizaban las redes Ethernet. De esta manera, 100BaseT mantiene el mismo formato y medida de trama, el mismo mecanismo de detección de errores y el mismo método de detección de colisiones.

Además, 100BaseT soporta velocidades duales de 10 y 100 Mbps y, por lo tanto, necesitará mecanismos de detección que seleccionen la mejor velocidad según las condiciones de transmisión.

Las redes 100BaseT presentan una característica opcional, denominada *autonegociación*, que permite a dos dispositivos intercambiar información sobre sus capacidades para crear un entorno de comunicaciones óptimo.

La autonegociación permite la comunicación de ciertos parámetros, como por ejemplo la velocidad de transmisión (10 Mbps o 100 Mbps), si el modo de operación es *full-duplex*, o bien la configuración de la señalización entre dispositivos.

Una de las diferencias más importantes entre 100BaseT y 10BaseT es el diámetro de la red: el diámetro máximo de una red 100BaseT es de doscientos cinco metros, aproximadamente diez veces menor que el de una red Ethernet de 10 Mbps.

El motivo de esta reducción del diámetro de la red es que 100BaseT utiliza el mismo mecanismo de detección de colisiones que 10BaseT. Si recordamos, en 10BaseT las limitaciones de distancia se definían para que una estación que trans-

mitía un paquete de tamaño mínimo a la estación más lejana de la red pudiera detectar una colisión. Este tamaño mínimo de la trama era de 64 bytes.

Si queremos mantener este tamaño mínimo de 64 bytes en Fast Ethernet, debemos recordar que ahora la velocidad de transmisión es diez veces mayor, pero la velocidad de propagación del medio no ha cambiado (continuamos hablando del mismo medio de transmisión). Consiguientemente, tendremos que acortar el dominio de colisión: una estación que transmite a una velocidad diez veces mayor deberá estar a una distancia máxima diez veces menor. Así conseguiremos que cualquier estación pueda detectar si se ha producido una colisión mientras transmite su trama.

100BaseT soporta tres medios de transmisión en la capa física:

- 100BaseTx
- 100BaseFx
- 100BaseT4

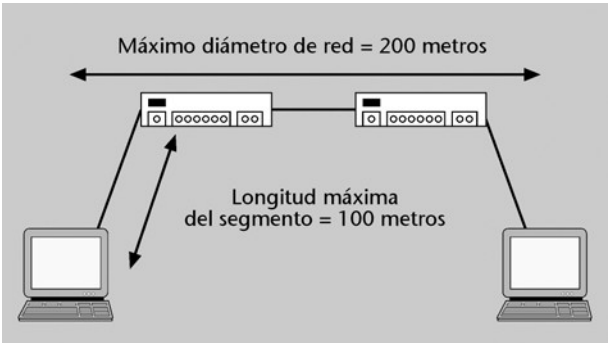
Las características de los medios físicos de 100BaseT se pueden resumir en la tabla siguiente:

| Características | 100BaseTx | 100BaseFx | 100BaseT4 |
|---|---|---|---|
| **Cable** | Categoría 5 UTP o STP | 62,5 / 125μ fibra multimodo | Categoría 3, 4 o 5 UTP |
| **Número de cables** | 2 pares | 2 fibras | 4 pares |
| **Longitud máxima del segmento** | 100 metros | 400 metros | 100 metros |
| **Máximo diámetro de la red** | 200 metros | 400 metros | 200 metros |

*100BaseTx*

100BaseTx soporta la transmisión de datos sobre cableado UTP y STP. En concreto, requiere que el cableado UTP sea de categoría 5, como mínimo, y funciona sobre dos pares.

La especificación IEEE 802.3u permite un máximo de dos repetidores y un diámetro máximo de red de unos doscientos metros. Además, un segmento, que se define como una conexión punto a punto entre dos dispositivos, será de cien metros como máximo.




Limitaciones de 100BaseTx y 100BaseT4.

*100BaseFx*

100BaseFx utiliza dos fibras ópticas multimodo como medio de transmisión. La especificación IEEE 802.3u define que la distancia máxima entre dos equipos de datos tiene que ser de unos cuatrocientos metros aproximadamente. En caso de que se necesite un repetidor, la longitud máxima será de unos trescientos metros.

*100BaseT4*

Si queremos disponer de la tecnología Fast Ethernet en una red ya cableada anteriormente y que utiliza un cableado de categoría 3, tendremos que utilizar la especificación 100BaseT4. 100BaseT4 necesita los cuatro pares de cobre para la transmisión de datos. Por ello, en este caso sólo podremos hablar de una red *half duplex*.

La especificación IEEE 802.3u permite un máximo de dos repetidores y un diámetro máximo de unos doscientos metros. La longitud máxima de un segmento será de cien metros, tal y como se muestra en la figura anterior.

### *5.2.2. Gigabit Ethernet*

La red Fast Ethernet se extendió muy rápidamente y superó las expectativas más optimistas. Como consecuencia, los precios bajaron y se popularizó su uso hasta el usuario final. Esto generaba requisitos de velocidades superiores en la red troncal (*backbone*) que no se podían atender con la tecnología del momento, si exceptuamos ATM a 622 Mbps, que tenía unos precios astronómicos. Esto provocó que en el año 1995 el subcomité 802.3 iniciara un nuevo grupo de trabajo, que creó la Gigabit Ethernet. El trabajo acabó con éxito el 29 de junio de 1998, con la aprobación del suplemento 802.3z.

Para aumentar la velocidad de 100 Mbps de Fast Ethernet hasta 1 Gbps, fue necesario hacer distintos cambios en la interfaz física. Por este motivo, Gigabit Ethernet es idéntica a la especificación Ethernet desde la capa de enlace en adelante: se ha mantenido el mismo formato de trama que en la 802.3. Las variaciones del nivel físico se han producido por la fusión de dos tecnologías: la especificación Ethernet 802.3 y ANSI X3T11 Fiber Channel.

La influencia de estas dos tecnologías significa que el estándar puede extraer beneficio de la tecnología de la interfaz a alta velocidad de Fiber Channel, a la vez que mantiene la compatibilidad con el formato de trama Ethernet 802.3 con las redes ya instaladas y el uso de modos de transmisión *full-duplex* o *half duplex*.

Las especificaciones elaboradas aprovechan los estándares ANSI de FDDI, de la misma manera que en Fast Ethernet, y aumenta diez veces la velocidad y utiliza como niveles físicos los siguientes:

- Emisores láser con fibra multimodo y monomodo
- Par trenzado apantallado
- Par trenzado no apantallado de categoría 5

La dificultad del reto tecnológico de la Gigabit Ethernet sobre UTP-cat.5 hizo que esta especificación se separara en un nuevo grupo que finalmente se especificó en marzo de 1999 como suplemento 802.3ab.

Gigabit Ethernet funciona a 1 Gbps y está orientada a conectar servidores, *bridges* o conmutadores; es decir, a funcionar como nivel físico de la red troncal de una red corporativa y, de forma ocasional, puede servir para conectar a un usuario final con su servidor.

Respecto de Gigabit Ethernet hay dos estándares, el correspondiente a IEEe 802.3z y el correspondiente a IEEE 802.3ab.

El estándar IEEE 802.3z funciona con dos fibras ópticas y tiene comportamiento *full-duplex*; por lo tanto, no utiliza CSMA/CD como modo de acceso al medio físico. Esta especificación está orientada a la conexión punto a punto de elementos de red troncal utilizando fibra óptica.

El estándar IEEE 802.3ab funciona con cuatro pares de cobre de categoría cinco, es similar al 100BaseT y sí que utiliza CSMA/CD como método de acceso al medio físico.

### *5.2.3. Funcionamiento de Gigabit Ethernet*

El estándar IEEE 802.3z especifica tres tipos de medios físicos:

- 1000BaseLX: láser de onda larga sobre fibra monomodo y multimodo
- 1000BaseSX: láser de onda corta sobre fibra multimodo
- 1000BaseCX: cable de cobre apantallado balanceado

El estándar IEEE 802.3ab especifica como medio de transmisión el cable UTP: 1000BaseT.

El nivel de enlace de Gigabit Ethernet es semejante al estándar Ethernet y Fast Ethernet explicados anteriormente. De este modo, las características de Ethernet –como la detección de colisiones, el diámetro de la red y el formato de la trama– serán las mismas para Gigabit Ethernet. Además, igual que Fast Ethernet, Gigabit Ethernet soporta transmisiones *full-duplex* y *half duplex*. Pero para poder hacer transmisiones *half duplex*, se añaden dos nuevas funcionalidades: extensión de portadora y *packet bursting*.

*Transmisión* half-duplex

Para las transmisiones *half duplex* se utiliza CSMA/CD con el objetivo de asegurar que las estaciones se pueden comunicar por un solo enlace y que pueden hacer una correcta detección de colisiones. La implementación de CSMA/CD para Gigabit Ethernet es la misma que para Ethernet o Fast Ethernet y permite la creación de una red compartida mediante concentradores (*hubs*) o conexiones punto a punto *half-duplex*.

En una transmisión *full-duplex* no se producen colisiones y, por consiguiente, no es necesario utilizar el protocolo de acceso al canal CSMA/CD.

CSMA/CD es un protocolo sensible al retraso y, por lo tanto, recordemos que es necesario definir el tamaño mínimo de trama o bien la máxima distancia entre dos estaciones que comparten el mismo segmento. De este modo, las estaciones podrán detectar posibles colisiones. A medida que la velocidad de la red aumente, el tiempo de transmisión mínimo de la trama disminuye, así como la máxima distancia entre estaciones.

El aumento de velocidades desde Ethernet hasta Gigabit Ethernet ha creado ciertos problemas en la implementación de CSMA/CD. Cuando IEEE definió la especificación 802.3u (100BaseT) mantuvo el mismo formato de trama que en Ethernet, y también la mínima longitud de trama. Evidentemente, si la velocidad es diez veces mayor que Ethernet, la trama es diez veces más corta (desde el punto de vista temporal). Puesto que la velocidad de propagación en el cable es constante, la distancia de colisión tiene que ser diez veces inferior, es decir, doscientos metros.

En la medida en que la 100BaseT confía en el *bridge* o conmutadores en lugar de confiar en los concentradores o repetidores, con esta distancia hay suficiente, pero cuando IEEE se plantea aumentar diez veces más la velocidad de transmisión con Gigabit Ethernet, aparece un grave problema: la distancia de colisión se reduce a veinte metros, lo cual es del todo impensable, ya que el parque instalado de par trenzado está limitado a cien metros.

Para resolver el problema de la distancia de colisión, IEEE suministra dos soluciones:

- Extensión de portadora
- *Packet bursting*

La técnica de extensión de portadora consiste en añadir bits a la trama Ethernet mínima hasta llegar a 512 bytes, y así se obtiene una trama de duración similar a una trama de 64 bytes a 100 Mbps. Dado que la extensión de portadora se hace desde el punto de vista físico, no hay que cambiar ningún tipo de software que haya escrito hasta el momento. Naturalmente, la extensión de portadora reduce el rendimiento de la Gigabit Ethernet a niveles alarmantes, ya que una trama de 64 bytes se envía a un ritmo de 120 Mbps, lo cual sólo aumenta en un 12% el rendimiento con respecto a la 100BaseT. Claro está que las estaciones no sólo envían tramas mínimas y que la longitud de trama depende de las capas de software que haya en

el servidor, pero el valor esperado de longitud de trama en la mayoría de las redes locales se sitúa, actualmente, entre los 200 y los 500 bytes.

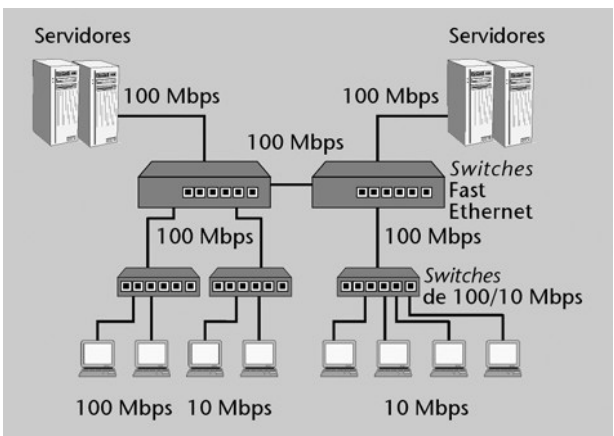
La técnica de *paquet bursting* consiste en introducir distintas tramas Ethernet clásicas en una supertrama mayor o igual que 512 bytes. De este modo, las otras estaciones verán la línea como ocupada y no transmitirán hasta que la estación origen no haya transmitido toda la supertrama.

### *5.2.4. Ejemplos de aplicación de Gigabit Ethernet*

A continuación veremos algunos escenarios en los que sería útil la aplicación de la tecnología Gigabit Ethernet.

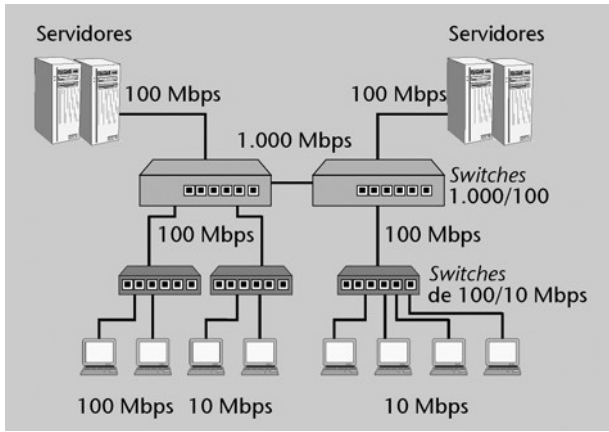
*Escenario inicial para una migración a Gigabit Ethernet*

Esquema de una corporación basada en una red troncal formada por dos conmutadores de 100 Mbps conectados entre ellos que derivan hacia repetidores de 100 Mbps y conmutadores de 100 a 10. Obviamente, tenemos un desequilibrio que se pondrá de manifiesto a medida que aumente el tráfico cruzado entre las dos ramas de la red troncal.



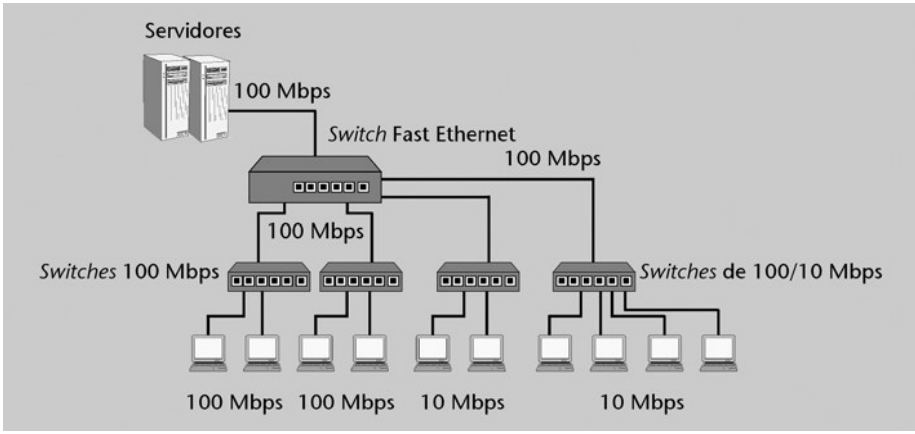

*Migración a Gigabit Ethernet*

Una primera migración que se puede hacer es cambiar los dos conmutadores 100/100 por un par de conmutadores 100/1.000:

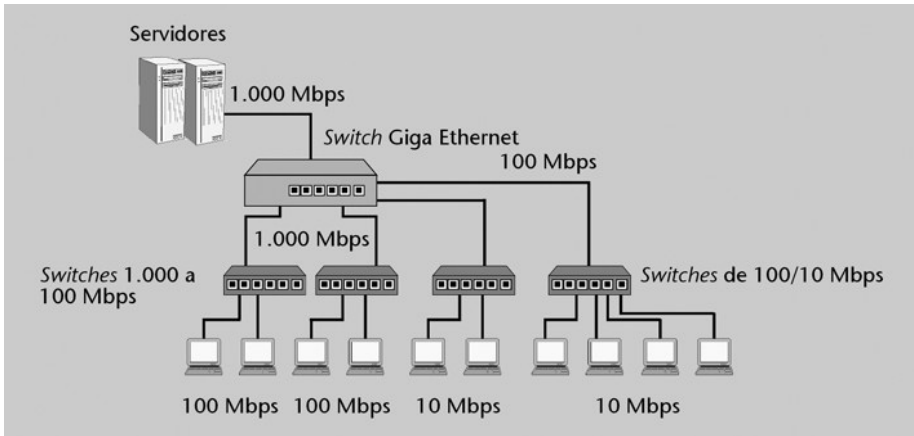



*Segundo escenario de migración a Gigabit Ethernet*

Otro ejemplo podría ser el de una corporación con una única red troncal de 100 Mbps, pero con una batería de servidores muy utilizados por la red y, además, con una necesidad de ancho de banda notable que proviene de las estaciones de trabajo de 100 Mbps.




El tráfico entre las estaciones de 100 MB y los servidores no está equilibrado. Una posible solución sería migrar la red troncal a Gigabit Ethernet y conectar los servidores y las estaciones que necesitan más ancho de banda a 1.000 Mbps.




Solución al segundo escenario de migración a Gigabit Ethernet.

## 5.3. Cableado EIA/TIA 568

Desde finales de los ochenta, los fabricantes empezaron a hacer presión para que se normalizasen las comunicaciones dentro de los edificios y se independizasen de los servicios, ya fueran de datos, de telefonía, etc. En 1991 se publicó el EIA/TIA 568 (la versión actual es el EIA/TIA 568B), referente al cableado de telecomunicaciones para edificios comerciales. Este estándar tenía los objetivos siguientes:

- Ser universal, tanto en servicios soportados como en fabricantes compatibles.
- Ser la base para el desarrollo de otros estándares de comunicaciones (voz, imagen, LAN, WAN, etc.), básicamente para comunicaciones en oficinas.
- Definir parámetros que permitiesen definir y establecer el cableado del edificio antes incluso de que la empresa se estableciese: se concibe el cableado como un servicio más del edificio (luz, agua… y datos).

El alcance del estándar cubre la especificación de la señal y la mecánica (tanto con respecto al uso de pares trenzados como de fibra óptica):

- Definición de señal. Prestaciones, ancho de banda, ruido e interferencias límite.
- Definición mecánica. Los conectores, disposición de contactos, armarios, geometría del cableado, alcance, etc.

Aunque el estándar define tanto el uso de fibras ópticas multimodo como monomodo, actualmente la alternativa más utilizada es el par trenzado. Tarde o temprano, la fibra sustituirá el cobre como soporte de las comunicaciones a causa de su mayor capacidad teórica, pero, por ahora, las sucesivas mejoras en la fabricación del cableado y, sobre todo, la mejora de los equipos terminales de circuito de datos, han hecho que la vida prevista del par trenzado se alargue.

### *5.3.1. Topología*

La topología básica de este estándar es la estrella, que se utiliza en el cableado de planta. El cableado de planta distribuye desde un punto central, o armario de planta (también denominado IDF, Intermediate Distribution Frame), a todos los puntos de la superficie o conectores. El estándar define el uso previsto de conectores.

Puesto que la definición del cableado es previa a la instalación de las oficinas, la localización de los conectores no puede basarse en la distribución final, ya que no se conoce.

Hay distintos mecanismos de distribución, pero el más utilizado es el de superficie, que establece la instalación de un conector doble cada 10 $m^2$ de espacio de trabajo. Éste es el espacio que se considera que ocupa un puesto de trabajo y se le prevé el uso de un terminal informático y un teléfono.

Los diferentes armarios de planta (usualmente uno por planta, si las dimensiones lo permiten) están conectados a la sala de equipamientos (o MDF, Main Distribution Frame). Esta sala concentra los bajantes de los armarios de planta y el cableado externo (telefonía, WAN, etc.). El cableado entre armarios de planta y la sala de equipamientos puede ser de hasta 90 m si es de cobre y hasta 3 km si es de fibra.

Tanto el armario de planta como la sala de equipamientos contienen *racks*. Dentro de los *racks* hay dos partes, que corresponden a dos funciones. Las entradas de planta (*patch panels*) identifican los conectores distribuidos por la planta. Mediante cables o puentes (*patch cords*) se conecta al servicio deseado: concentrador, conmutador, centralita telefónica, etc. El cableado horizontal puede medir como mucho 90 m (ya sea cobre o fibra), que sumados a la longitud de los puentes del armario y los puentes para conectar los terminales dan un total de 100 m como máximo.

Los *racks* de los armarios de planta tienen conexión con la sala de equipamientos. Es por este motivo por lo que normalmente se habla de topología en estrella jerárquica. La principal tiene el punto central en la sala de equipamientos y cada punta, los armarios de planta, son los nudos de las estrellas subordinadas.

### *5.3.2. Medio físico*

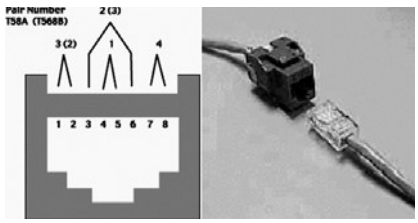
Como medio físico, hay diferentes tipos de par trenzado y de fibra óptica:

| | **Ancho de banda** | **Servicios posibles** |
|---|---|---|
| **UTP Categoría 1** | Sin definir | Telefonía |
| **UTP Categoría 3** | 16 MHz | LAN de baja velocidad: Ethernet 10BaseT |
| **UTP Categoría 4** | 20 MHz | LAN de baja velocidad: Token Ring 16 Mbps |
| **UTP Categoría 5e** | 100 MHz | LAN de alta velocidad: Fast Ethernet, Gigabit Ethernet, Fiber Channel, CDDI, etc. |
| **UTP Categoría 6** | 250 MHz | Sin servicios exclusivos actualmente, pero permite los servicios anteriores y servicios de mayor ancho de banda. |
| **F.O. multimodo de salto de índice** | 100 MHz | LAN de alta velocidad: Fast Ethernet, FDDI, etc. |
| **F.O. multimodo de gradiente de índice** | >1 GHz | LAN de alta velocidad: Gigabit Ethernet, Fiber Channel |
| **F.O. monomodo** | >1 GHz | Transmisión a larga distancia |

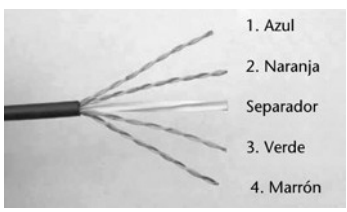
Los medios conductores están divididos en categorías que se diferencian básicamente por los sistemas de conexión (conectores, paneles), el cableado (densidad del trenzado de los pares) y los mecanismos de certificación. En la actualidad, el cableado más utilizado es el de categoría 5e.

De hecho, la categoría 5e es una variación en los parámetros de certificación de la antigua categoría 5 para garantizar el correcto funcionamiento de las LAN Gigabit Ethernet y Fiber Channel (entonces en proceso de definición). Teóricamente, un cableado categoría 5 es capaz de soportar tráfico Gigabit Ethernet (por ancho de banda, interferencias, etc.), pero se recomienda añadir los tests adicionales categoría 5e.

Cada uno de los terminales recibe cuatro pares trenzados que permiten soportar todo tipo de servicios. El servicio lo reciben por medio de un conector RJ45:

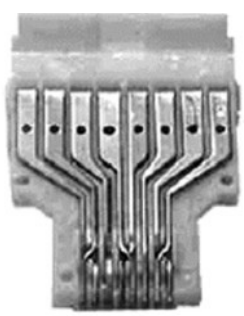



Cada conector recibe un cable con cuatro pares de hilos con colores distintos:




Es muy delicado trabajar con cableados categoría 5e y superior, ya que la geometría de la trenza está especialmente estudiada para evitar las interferencias entre los cuatro pares y el resto de los cables que pueda haber en los alrededores. Un error en el crimpado del conector que deshiciera sólo una bóveda de trenza podría convertir una instalación categoría 5 o 5e en categoría 3.

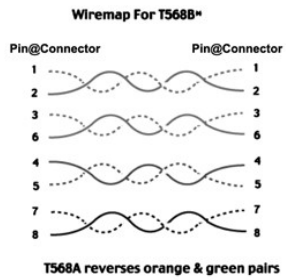
El trenzado debe continuarse, incluso, en los conectores:



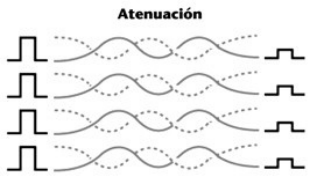
### *5.3.3. Certificación*

El proceso de certificación homologa una instalación (entera) como de una categoría o de otra. Habitualmente, se utilizan equipos que realizan todas las operaciones que a continuación comentaremos, de manera automática. Las mediciones básicas que hacen los equipos de tests son las siguientes:

**Continuidad**. Se mide que la conexión pin a pin sea correcta.



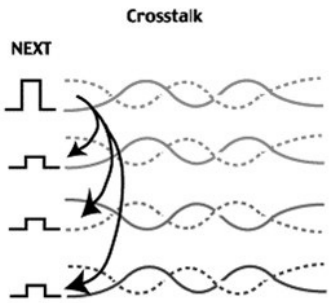

**Longitud**. Se lanzan pulsos de señal y se cronometra la señal del eco. De esta manera se comprueba que el cableado no supera la longitud máxima permitida. Con esta medición también se pueden identificar problemas de homogeneidad de la línea según los ecos adicionales que se puedan detectar.




**Atenuación**. Se envía una señal patrón desde un emisor y se mide la atenuación con un receptor a los cuatro pares en el otro extremo del cable. La señal de medida debe tener una frecuencia que ocupe sobre todo la banda alta de frecuencias, ya que las frecuencias altas son las que más atenuación sufren.

**NEXT** (*Near End Cross(X)-Talk*.) o autointerferencia. El NEXT se mide aplicando una señal a potencia nominal y frecuencia máxima a un par y midiendo la señal que pasa por radiación a los otros tres pares en el mismo extremo. Esta medida se repite para los cuatro pares.

Un valor más gráfico que el NEXT es el ACR (*Attenuation to Crosstalk Ratio*), que mide la diferencia entre el NEXT y la atenuación. Cuanto mayor sea el ACR, mejor:




**Otros tests**

Con la aparición de servicios sobre cableado estructurado, como la Gigabit Ethernet, que usan los cuatro pares en las dos direcciones simultáneamente, se han ido añadiendo otras medidas, como por ejemplo el cálculo del retardo diferencial en los cuatro pares, los ecos, etc.

## 5.4. Redes inalámbricas: norma IEEE802.11

IEEE 802.11 especifica un MAC CSMA/CA para redes inalámbricas. El estándar especifica diferentes bandas posibles, e incluso LAN por infrarrojos, pero las más utilizadas son las bandas ISM (*Industrial, Scientific and Medical)* de 2,4 GHz y 5 GHz.

**Banda ISM**

La banda ISM define las bandas de acceso libre regulado de 0,9 GHz, 2,4 GHz y 5,7 GHz:

| ISM | $f_{mím}$ | $f_{máx}$ |
|---|---|---|
| Banda 0,9 GHz | 902 MHz | 928 MHz |
| Banda 2,4 GHz | 2,4000 GHz | 2,4835 GHz |
| Banda 5 GHz | 5,150 GHz | 5,825 GHz |

La banda de 5 GHz está dividida en dos subbandas, separadas por un vacío de entre 5,35 GHz y 5,725 GHz.

IEEE 802.11 define dos modos de funcionamiento, uno basado en un punto central llamado punto de acceso (modo infraestructura) y el otro sin este punto (modo *ad-hoc*).

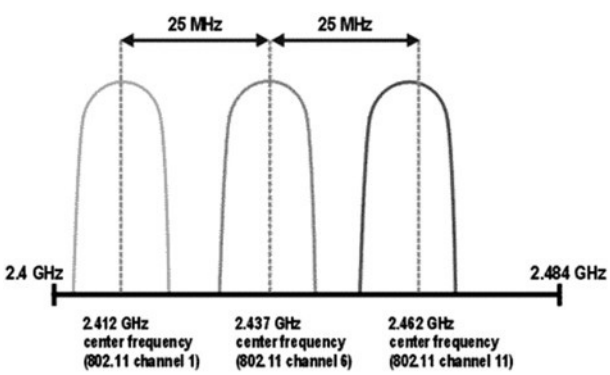
En el modo de funcionamiento basado en un punto central, aparte del control de acceso CSMA/CA, se puede reservar, bajo petición al punto de acceso, una fracción del tiempo para tráfico síncrono (para voz o imagen en tiempo real). Nosotros nos centraremos exclusivamente en el tráfico de datos, por lo tanto, controlado por CSMA/CA.

### *5.4.1. Nivel físico*

IEEE802.11 define distintos niveles físicos de entre 1 Mbps y 54 Mbps. Sin embargo, los más utilizados en la actualidad son los siguientes:

- IEEE802.11b (11 Mbps). Utiliza la banda de 2,4 GHz de la ISM. Dispone de catorce canales con un ancho de banda de 22 MHz cada uno. Si el SNR empeora, automáticamente la velocidad baja a 5,5 Mbps, 2 Mbps y 1Mbps. Es el más utilizado actualmente.

  Los catorce canales se reparten los 83,5 MHz disponibles. Por lo tanto, es obvio que no se pueden utilizar simultáneamente, ya que se interfieren. Como máximo se pueden utilizar tres canales dentro de un área determinada (por ejemplo el 1, el 6 y el 11).




  El número de canales disponibles varía en cada país, ya que la banda regulada puede variar.

Como se puede ver en la tabla siguiente, las legislaciones española y francesa son particularmente restrictivas, aunque se espera que esto cambie, tal y como ya ha pasado en Japón:

| Identificación de canal | Frecuencia (MHz) | Dominios reguladores | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | X'10'FCC | X'20'IC | X'30'ETSI | X'31'España | X'32'Francia | X'40'Japón | X'41'Japón |
| 1 | 2412 | X | X | X | - | - | - | X |
| 2 | 2417 | X | X | X | - | - | - | X |
| 3 | 2422 | X | X | X | - | - | - | X |
| 4 | 2427 | X | X | X | - | - | - | X |
| 5 | 2432 | X | X | X | - | - | - | X |
| 6 | 2437 | X | X | X | - | - | - | X |
| 7 | 2442 | X | X | X | - | - | - | X |
| 8 | 2447 | X | X | X | - | - | - | X |
| 9 | 2452 | X | X | X | - | - | - | X |
| 10 | 2457 | X | X | X | X | X | - | X |
| 11 | 2462 | X | X | X | X | X | - | X |
| 12 | 2467 | - | - | X | - | X | - | X |
| 13 | 2472 | - | - | X | - | X | - | X |
| 14 | 2484 | - | - | - | - | - | X | - |

Fuente: IEEE802.11b..

Los países marcados como ETSI son los europeos, excepto Francia y España. FCC es EE.UU., e IC, Canadá.

- IEEE802.11a (54 Mbps). Utiliza la banda de 5 GHz de la ISM. Tiene un alcance un poco inferior al de IEEE802.11b y ha sido superado por IEEE802.11g. Si el SNR empeora, la velocidad de transmisión se ajusta a 24 Mbps, 12 Mbps y 6 Mbps (aunque se pueden implementar opcionalmente otras velocidades intermedias).

- IEEE802.11g (54 Mbps). Es equivalente a IEEE 802.11a, pero utiliza la banda de 2,4 GHz de la ISM, que es menos vulnerable a los obstáculos (mayor alcance) y es interoperable con tarjetas IEEE802.11b.

### *5.4.2. Control de acceso al medio*

En este estándar no se puede utilizar el CSMA/CD como control de acceso al medio porque requiere que la electrónica de transmisión sea capaz de transmitir y recibir al mismo tiempo; y, en radiotransmisión, para conseguirlo habría que utilizar bandas de frecuencia distintas para transmitir y para recibir, lo que obligaría a dividir el ancho de banda en subbandas más pequeñas, con lo que se perderían prestaciones.

CSMA/CD no es aplicable para controlar el acceso al medio en este estándar porque la detección de colisiones (CD) exige tener un transmisor-receptor *full-duplex*.

No hay que confundir el hecho de que la electrónica de transmisión del CSMA/CD sea *full-duplex*, con el hecho de que el MAC CSMA/CD sea *full-duplex*. El CSMA/CD y todo el resto de los protocolos MAC son *half duplex* por definición: lo que un terminal transmite lo reciben los $n - 1$ restantes. Si transmitiesen dos, los $n - 2$ recibirían la suma de las dos señales y no entenderían nada.

Los ingenieros tuvieron que elegir un control de acceso al medio alternativo al CSMA/CD.

MAC dinámico estaba fuera de discusión, ya que una de las premisas del diseño era el soporte de sistemas móviles en movimiento (*roaming*): ir moviéndose por el espacio saltando de una LAN a otra implicaría una reconstrucción del anillo (lógico forzosamente) que fue descartada desde un principio.

Entre los sistemas aleatorios, el CSMA es el mejor de los que no requieren transmisores-receptores *full-duplex*. El problema es que se degrada rápidamente cuando se trabaja en redes donde la cobertura de señal no es completa.

Tal como se ha visto en el apartado anterior, la mejor opción es un CSMA/CA.

### *5.4.3. Topología*

Ya hemos visto que la única topología posible para redes inalámbricas es el bus. Cada dominio de colisión se llama BSS o *Basic Service Set*. Dentro de un mismo volumen sólo puede haber un BSS para cada canal.

Sin embargo, WiFi aporta dos modos de funcionamiento alternativos, cada uno con sus ventajas.

- Modo *ad-hoc*. Es el más puro de los dos, ya que el bus (el aire) es compartido sin intermediarios. Las estaciones seleccionan la banda de frecuencias que quieren utilizar y poca cosa más. Cada banda de frecuencias, el canal, es un dominio de colisión independiente.
- Modo con infraestructura. El BSS está comandado por un punto de acceso (*Access Point* o AP). Todas las transmisiones son entre el AP y los terminales, nunca entre terminales. Es aproximadamente equivalente a una topología en estrella, donde el AP es el repetidor o concentrador: todas las transmisiones de estación a estación deben pasar forzosamente a través del AP.

El modo *ad-hoc* tiene la ventaja de que no necesita hardware adicional a las tarjetas de las estaciones: dos terminales con las correspondientes tarjetas ya se pueden poner a trabajar.

El modo con infraestructura tiene la ventaja de que los terminales pueden conectarse aunque no tengan conexión ("visión") directa. Esto, que a primera vista parece trivial, no lo es tanto ya que los terminales normalmente están sobre mesas y se mueven; en cambio, los AP se suelen poner levantados del suelo en posiciones fijas y estudiadas. Es mucho más fácil garantizar conexión de todos contra el AP, que de todos contra todos.

En definitiva, el modo *ad-hoc* es más útil para redes casuales (por ejemplo, una reunión) que para trabajo diario.

# 6. Nuevas prestaciones de las redes Ethernet

## 6.1. Ethernet full-duplex

Las redes Ethernet se diseñaron como una tecnología *half-duplex*, es decir, que la transmisión de datos entre una estación origen y una estación destino sólo se puede producir en un único sentido a la vez. Así pues, cualquier estación Ethernet antes de transmitir información comprueba que el medio no esté ocupado. Si está libre, la estación empieza a transmitir y lo ocupa. Pero tal como hemos visto, eso puede provocar colisiones si dos o más estaciones empiezan a transmitir simultáneamente, y por lo tanto, se necesitan mecanismos para evitar esta situación, o cuando menos, recuperarse si se produce.

Añadir las técnicas de conmutación a las LAN con el uso de los conmutadores ha permitido incorporar la posibilidad de transmisiones *full-duplex* en Ethernet, y así aumentar el rendimiento de la red. Por qué los conmutadores permiten transmisiones *full-duplex* está claro: los enlaces entre el conmutador y las estaciones son independientes entre ellos. Dicho de otra manera, cada estación tiene su propio dominio de colisión. Por lo tanto, el conmutador puede intercambiar tramas con todas las estaciones simultáneamente, y en *full-duplex*.

Obviamente no se pueden construir concentradores *full-duplex*, ya que en cada dominio de colisión sólo puede haber una transmisión activa a cada instante. De hecho, el concepto dominio de colisión es de poca o nula aplicación en puertos *full-duplex*.

De hecho, eso ya no es una red local CSMA/CD. Lo que tenemos son enlaces punto a punto entre las estaciones y el conmutador que anecdóticamente utiliza como formato de trama el Ethernet y como codificación utiliza 10baseT, 100baseT, etc.

Ethernet en modo *full-duplex* permite la transmisión y la recepción de datos entre dos estaciones de manera simultánea.

Para que una red Ethernet pueda trabajar en modo *full-duplex* se tienen que cumplir varias condiciones:

- El medio físico tiene que permitir la transmisión *full-duplex*.
- Sólo puede haber dos dispositivos conectados entre ellos.

- Las tarjetas de red de los dos equipos tienen que soportar la transmisión *full-duplex*.

No todos los medios físicos pueden soportar transmisiones full-duplex. Por ejemplo, 10Base5, 10Base2 o 100Base-T4 no lo permiten. En cambio, 10BaseT, 100BaseT o 100BaseFl sí que lo permiten.

Hasta ahora, la sustitución de un concentrador por un conmutador no exigía ningún cambio más. Pero con Ethernet *full-duplex* hay que cambiar las tarjetas de las estaciones, ya que las normales aplican el CSMA/CD de manera contumaz, abortando cualquier transmisión simultánea en las dos direcciones.

Otra característica importante de una red Ethernet *full-duplex* es la supresión de la restricción sobre la distancia máxima. Recordemos que esta restricción la imponía el protocolo de acceso al medio con el fin de poder detectar las posibles colisiones. Sin embargo, si en el modo *full-duplex* tenemos un dominio libre de colisiones, podemos ver que esta restricción ya no es necesaria. Ahora será la atenuación de la señal que provoque el medio físico utilizado la que imponga la limitación de distancia.

Así, pues, el uso de Ethernet *full-duplex* aumenta el rendimiento de la red, permite distancias mayores y simplifica el funcionamiento, porque ya no se utiliza el protocolo de control de acceso al medio. Además, la transmisión *full-duplex* en las redes Gigabit Ethernet permite suprimir las técnicas de extensión de portadora y de ráfagas de tramas, ya que en este entorno no son necesarias.

## 6.2 Control de flujo

Inicialmente, diferentes fabricantes implementaron el modo de transmisión *full-duplex* como una extensión no estándar. Sin embargo, en 1997 se estandarizó mediante el subcomité 802.3x y se incluyó la funcionalidad de control de flujo.

El control de flujo se utiliza entre estaciones que se transmiten datos en modo *full-duplex*. Cuando la estación receptora detecta que no puede aceptar más tramas entrantes, envía un comando llamado *pause*. Este comando indica al emisor que pare de enviar información durante un periodo de tiempo especificado en la misma trama. Durante este tiempo, el receptor puede enviar nuevos comandos *pause* que alarguen, reduzcan o eliminen la pausa que ha anunciado previamente.

El control de flujo es una técnica que quiere evitar el desbordamiento de las memorias intermedias (*buffers*) de almacenamiento del receptor, junto con la consiguiente eliminación de tramas que este hecho comportaría.

## 6.3. Sistemas multiestándar: autonegociación

Los sistemas Ethernet muchas veces soportan los diferentes estándares vistos en anteriores apartados. Pueden trabajar con diferentes medios físicos (100BaseTx, 100BaseT4, 10BaseT…), con diferentes velocidades de transmisión, y con *half-duplex* o *full-duplex*. Además, los conmutadores permiten que diferentes puertos trabajen de diferente manera.

La técnica de autonegociación permite a dos dispositivos el intercambio de información sobre sus capacidades, para crear un entorno a comunicaciones eficiente.

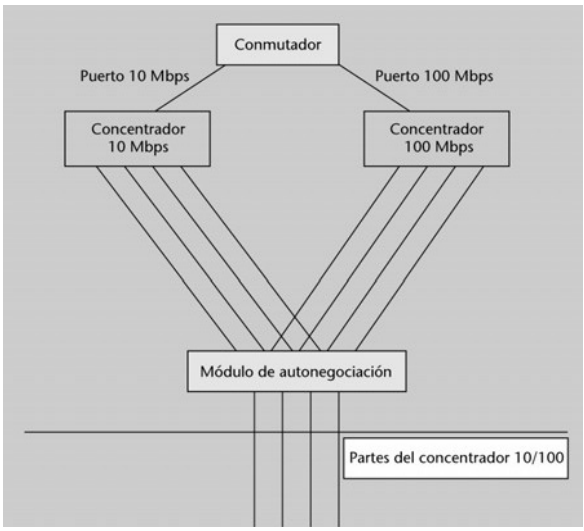
**Estándar**

Este procedimiento ha sido estandarizado en la misma revisión que la Fast Ethernet: IEEE802.3u. Así, cada vez que uno de los puertos de un conmutador se reinicia, pone en marcha un procedimiento de autonegociación, que consiste a acordar con el oponente el estándar de más prestaciones que tengan en común.

En el caso de dispositivos con RJ-45, en primer lugar se negocia el medio físico y la velocidad de transmisión, que empieza por Gigabit Ethernet y acaba con el Ethernet clásico. A continuación se negocia el modo de transmisión *half-duplex* o *full-duplex* y, finalmente, el control de flujo y el establecimiento simétrico o asimétrico de éste. Con la fibra óptica el medio físico no es negociable y, por lo tanto, sólo se intercambian información sobre el modo de funcionamiento y el control de flujo.

En cambio, un concentrador no puede actuar de la misma manera: no puede tener puertos *full-duplex*, y tampoco puertos de diferentes velocidades. Un concentrador es simplemente un repetidor de la señal que recibe; por lo tanto, no es posible que una señal recibida a una velocidad se retransmita a una velocidad diferente.

Hay que tener en cuenta que hablamos de concentradores reales. El mercado está lleno de concentradores multivelocidad a 10/100 Mbps y a 10/100/1000 Gbps (*half duplex* todos ellos, eso sí) y con capacidad de autonegociación. El intríngulis se resuelve cuando estudiamos su arquitectura, y es que no se trata de un concentrador, sino de dos (o tres) interconectados mediante un conmutador de dos o tres puertos. La arquitectura para un concentrador 10/100 es la siguiente:




Un concentrador 10/100 crea dos dominios de colisión (un concentrador 10/100/1000 crea tres) interconectados por un conmutador. Por lo tanto, la capacidad del concentrador será 10 Mbps + 100 Mbps = 110 Mbps (1,11 Gbps en un concentrador 10/100/1000).

## 7. Redes de área local virtuales: VLAN

La configuración de una red de área local típica depende, en gran manera, de la infraestructura física de sus conexiones. Así pues, la agrupación de usuarios se hace según su situación física, su conexión a un determinado concentrador o conmutador y su cableado correspondiente. Si el dispositivo que interconecta los diferentes concentradores es un encaminador (*router*), éste proporcionará segmentación y evitará que el tráfico de difusión (*broadcast*) pase de una red a otra.

En cambio, si el dispositivo que interconecta los diferentes concentradores es un conmutador, éste no podrá evitar que el tráfico de difusión pase de un segmento a otro. Además, este tipo de segmentación no permite agrupar a los usuarios según el tipo de tarea que hacen, el departamento al cual pertenecen o el ancho de banda que necesitan.

Una LAN virtual, o VLAN, es una agrupación lógica de dispositivos o de usuarios. Esta agrupación no está restringida por el segmento físico donde están conectados, sino que los dispositivos o usuarios se pueden agrupar por la función que lleven a cabo, el departamento del cual forman parte o la aplicación que utilizan, por ejemplo.

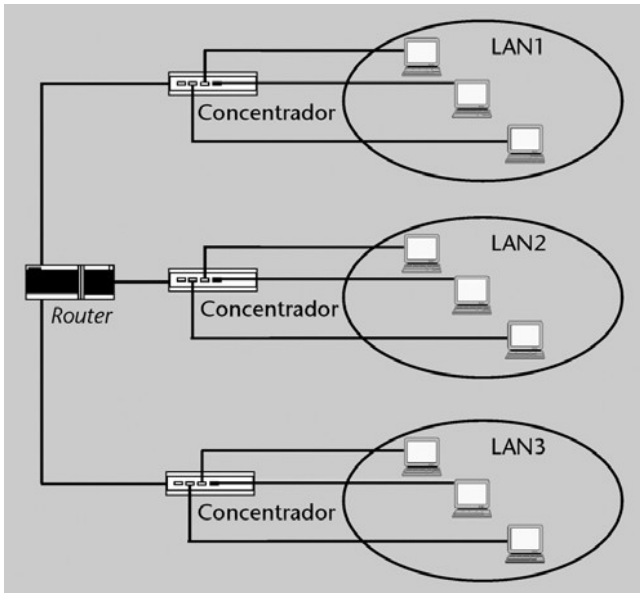
Una VLAN crea un dominio de difusión único que no está restringido a un segmento físico. Así pues, el tráfico de difusión no se propaga entre las LAN virtuales. Por lo tanto, para que dos usuarios de VLAN diferentes se puedan comunicar entre ellos, será necesario que un encaminador interconecte las dos VLAN. Dos usuarios que estén conectados al mismo conmutador y que pertenezcan a VLAN diferentes no se podrán comunicar entre ellos mientras no haya un encaminador entre las dos VLAN.

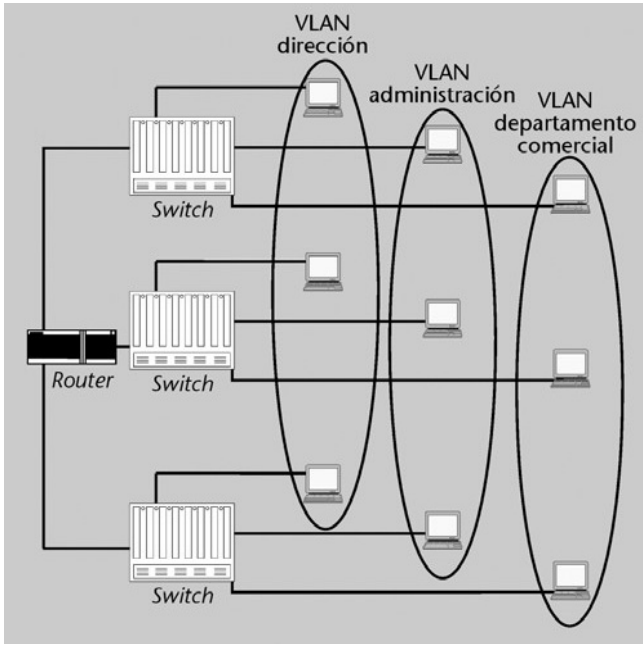
Es importante destacar que la implementación de VLAN se hace mediante software instalado en los conmutadores. Consiguientemente, cada fabricante implementa su versión para la creación y gestión de VLAN.

## 7.1. Características de las VLAN

Las VLAN segmentan de manera lógica la infraestructura física de las redes de área local. Este hecho permite agrupar a usuarios que están separados geográficamente en una misma VLAN. Además, con esta agrupación lógica se consigue que el tráfico de difusión que genera una estación de una VLAN se transmita sólo a los puertos que pertenecen a la misma VLAN.

Las figuras siguientes muestran la diferencia entre la segmentación LAN y VLAN:




En una red que ya utiliza elementos de conmutación y en la cual, además, los conmutadores soportan la programación de VLAN, la migración hacia un entorno con LAN virtuales es muy sencillo. En este caso, sólo hay que decidir cuál

será el criterio de agrupación de los dispositivos y configurar los conmutadores de manera adecuada. En caso de que la red no sea conmutada y utilice concentradores, el primer paso consistirá en sustituirlos por conmutadores que soporten VLAN. Sin embargo, si todos los dispositivos conectados a un determinado concentrador tienen que pertenecer a la misma VLAN, sólo habrá que conectar uno de los puertos del concentrador a un conmutador.

Las características más importantes de las VLAN son las siguientes:

- Las VLAN trabajan en los niveles 2 y 3 del modelo de referencia OSI.
- El tráfico de difusión sólo se propaga entre los puertos de una misma VLAN, pero no se transmite entre VLAN.
- La comunicación entre VLAN necesita un dispositivo de encaminamiento de nivel de red.
- Las VLAN se configuran mediante software y es el administrador de red el que asigna a los diferentes usuarios a cada VLAN.
- La configuración de las VLAN depende de cada conmutador: cada fabricante dispone de una implementación propietaria de gestión de VLAN en sus conmutadores.
- La definición de VLAN mejora la seguridad de una red, pues permite definir qué comunicaciones entre VLAN están permitidas y cuáles no.
- La agrupación de dispositivos se puede hacer por departamento o por el tipo de aplicaciones que utilizan los usuarios, por ejemplo.

En una arquitectura VLAN es importante poder transportar información entre los diferentes conmutadores y encaminadores que forman parte de toda la red. Recordemos que usuarios de una misma VLAN pueden estar conectados a diferentes conmutadores y, consiguientemente, éstos tienen que ser capaces de poder intercambiarse información. Así pues, la troncal de la red estará formada por enlaces entre estos conmutadores y los encaminadores. Además, deberá soportar grandes volúmenes de información entre los dispositivos, y también información de las VLAN y de los dispositivos que las forman.

El encaminador, dispositivo de nivel de red, proporciona la comunicación entre los grupos de trabajo definidos lógicamente, las diferentes VLAN. Además, proporciona acceso a los recursos compartidos como, por ejemplo, los servidores. También permite la conexión con otras zonas de la red que no sigan la arquitectura de VLAN.

El conmutador es el dispositivo fundamental de la arquitectura VLAN. Este dispositivo tiene que ser capaz de filtrar las tramas según los criterios elegidos por el administrador de red a la hora de crear las VLAN.

## 7.2. Tipos de VLAN

Mediante VLAN podemos disponer de una red conmutada que está segmentada lógicamente según las aplicaciones, los protocolos y las funciones de los usuarios, sin que importe dónde están situados físicamente estos usuarios. Así pues, podríamos definir que cada puerto de un conmutador forma parte de una VLAN distinta, si es necesario.

Las VLAN se pueden clasificar según la forma de asignación de los puertos de un conmutador a una VLAN. Sin embargo, otra manera de clasificar las VLAN dependerá del tipo de información que utilice el conmutador para agrupar los dispositivos de una manera lógica.

A continuación analizaremos las diferentes VLAN, que se pueden clasificar en los tipos siguientes:

- VLAN estáticas o dinámicas
- VLAN basadas en el puerto o en el protocolo

### *7.2.1. VLAN estáticas y VLAN dinámicas*

En una VLAN estática el administrador de red define de manera permanente la relación entre los puertos de un conmutador y la VLAN a la cual pertenecen.

En una VLAN estática los puertos de un conmutador se asignan estáticamente a una VLAN determinada. Así pues, la configuración sólo cambiará si el administrador de red cambia un puerto de una VLAN a otra. En caso contrario, esta configuración se mantendrá fija.

Este tipo de VLAN comporta un control muy grande del administrador, pero es fácil de configurar, monitorizar y gestionar. Esto se debe a que es el administrador el que tiene el control total.

En una VLAN dinámica, los puertos de un conmutador determinan de manera automática su asignación a una VLAN.

La asignación automática del puerto a las VLAN dinámicas puede depender de las direcciones de nivel de enlace, de las direcciones lógicas o bien del tipo de protocolo de los paquetes.

**Procedimiento de asignación automática del puerto a las VLAN dinámicas**

Imaginemos que el criterio de selección es la dirección de nivel de enlace. Cuando una estación está conectada a un puerto que no pertenece a ninguna VLAN y transmite una trama, el conmutador detecta la dirección de nivel de enlace en la trama, busca a qué VLAN debe pertenecer en su base de datos de VLAN y, automáticamente, configura el puerto con las características de la VLAN correspondiente.

Este tipo de VLAN requiere menos administración en caso de que un usuario cambie de situación geográfica, ya que no es necesario definir para cada puerto su asignación de VLAN. Sin embargo, hay que disponer de una base de datos actualizada con todos los usuarios y el criterio de asignación de VLAN elegido.

### *7.2.2. VLAN basadas en el puerto o en el protocolo*

En las VLAN basadas en el puerto, el conmutador utiliza los números de puerto para hacer la agrupación lógica de usuarios.

Es decir, a cada VLAN le corresponderá toda una serie de puertos determinados. En el fondo, tendremos que entender que se trata de una VLAN estática, como la estudiada anteriormente.

Cuando hablamos de VLAN basada en el protocolo, el conmutador clasificará el tráfico recibido según el tipo de protocolo del paquete de nivel de red.

En el caso de VLAN basadas en el protocolo la información discriminante será el tipo de protocolo de nivel 3. Así pues, podremos tener VLAN IP, VLAN IPX, etc.

### 7.3. IEEE 802.1Q

El conmutador es el dispositivo encargado de filtrar y permitir el paso de las tramas según los criterios de VLAN y, además, tiene que compartir esta información con otros conmutadores y encaminadores de la red.

Si tenemos a usuarios de una misma VLAN conectados a diferentes conmutadores, será necesario que éstos se pasen información entre ellos y que conozcan a qué VLAN pertenecen las tramas que conmutan. Se puede dar el caso de que un usuario conectado a un conmutador envíe tramas a otro usuario de la misma VLAN que cuelga de otro conmutador. Además, recordemos que las tramas de difusión llegan a todos los miembros de una VLAN. Por lo tanto, un conmutador deberá reconocer y asociar las tramas que recibe con la VLAN correspondiente.

La técnica de identificación de tramas asigna un identificador determinado para cada trama. Esta técnica introduce un identificador en la cabecera de la trama de nivel de enlace. Este identificador se transmite para la red troncal y es examinado por los diferentes conmutadores y encaminadores que atraviesa. Cuando la trama está a punto de salir de la red troncal, el conmutador elimina este identificador, con el objetivo de que no llegue hasta la estación de destino.

IEEE 802.1q define que la técnica de identificación de tramas es la solución utilizada en la implementación de VLAN.

Así pues, un conmutador puede tener dos tipos de puertos, que se clasifican según su funcionalidad. Estos puertos se denominan:

- *Access port*. Puerto que pertenece a una determinada VLAN.
- *Trunk port*. Puerto que pertenece a diferentes VLAN.

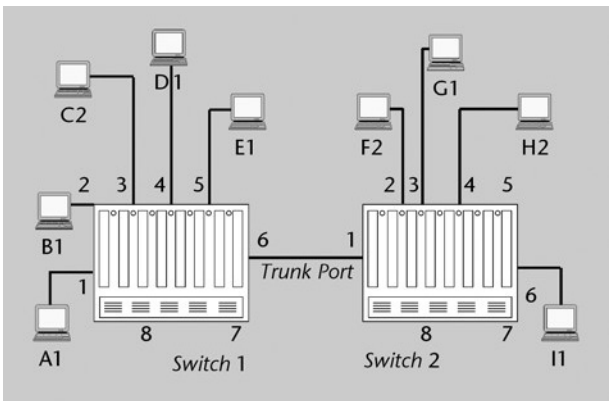
Cualquier puerto, por defecto, siempre es *access port* y, por lo tanto, únicamente puede pertenecer a una sola VLAN. En cambio, un *trunk port* permitirá la interconexión de conmutadores y, consiguientemente, tendrá que pertenecer a todas las VLAN de las cuales transmita tramas. En un *trunk port*, la información viajará acompañada con la etiqueta de la VLAN correspondiente. En cambio, las tramas que salen de un puerto de acceso siempre estarán sin marcar (*untagged*) y esto significa que no llevarán información especial con respecto a qué VLAN en particular pertenecen.

Los *trunk ports* se utilizan usualmente para interconectar diferentes conmutadores, sin tener en cuenta a qué VLAN pertenecen las tramas intercambiadas.

Por ejemplo, en una situación como la de la figura de la página siguiente, las tramas que salen de un conmutador hacia el otro conmutador tienen que mantener la información con respecto a qué VLAN pertenecen para que puedan llegar a su destino. En cambio, cualquier trama que salga de un puerto de acceso es *untagged*. Esto quiere decir que no podemos utilizar un puerto de acceso para unir los dos conmutadores.

Por los *trunk ports* circulan tramas que pueden pertenecer a más de un tipo de VLAN. Si estos dos conmutadores están interconectados, el puerto de unión entre los dos conmutadores debe trabajar con tramas que pueden pertenecer a las dos VLAN, y por ello deberá ser un *trunk port*, ya que si una trama llega al conmutador 1 se tendrá que enviar al conmutador 2, tanto si pertenece a la VLAN1 como si pertenece a la VLAN2. El *trunk port* deberá ser capaz de distinguir a qué VLAN pertenece la trama para dirigirla allí donde sea necesario.

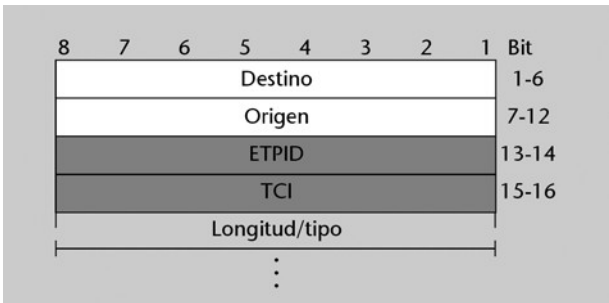
**Uso de los trunk ports**



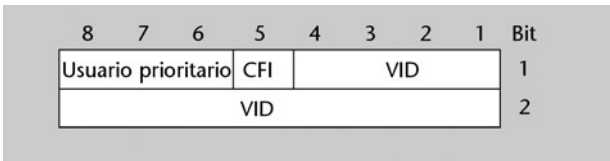

En esta red, si inferimos que el número de cada estación indica la VLAN a la cual pertenece, tendríamos que configurar los conmutadores de la manera siguiente:
Conmutador 1:
Puerto 6: trunk port
Puertos 1, 2, 4, 5 i 6: VLAN 1
Puertos 3, 6: VLAN 2
Conmutador 2:
Puerto 1: trunk port
Puertos 1, 3, 6: VLAN 1
Puertos 1, 2, 4: VLAN 2

Como hemos comentado anteriormente, los *trunk ports* siempre transmiten y reciben tramas *tagged*, es decir, tramas marcadas. El estándar IEEE 802.1Q especifica el formato de esta marca. La única excepción son las tramas del protocolo Spanning Tree, que se transmiten *untagged*.

El estándar 802.1Q incluye la definición del formato de trama de una VLAN que lleva la identificación de la VLAN y la información de la prioridad del usuario. Esta información adicional se incluye en un nuevo campo conocido con el nombre de ***tag header*** y que está situado inmediatamente después de los campos de las direcciones MAC destino y origen existentes en la trama original.




Formato del campo *Tag header* definido por el IEEE 802.1q.




Formato del campo TCI.

Los campos que forman la trama de identificación de VLAN, definida por el IEEE 802.1q, son los siguientes:

- ETPID (*Ethernet-coded tag protocol identifier*): su valor es 81-00.
- TCI (*tag control information*), que se descompone en los campos siguientes:
  - *User priority*: 3 bits que representan ocho niveles de prioridad.
  - CFI: *canonical format indicator*.
  - VID (VLAN *identifier*): identifica a qué VLAN pertenece la trama.
- 0: indica que no hay información VLAN, es decir, que el *tag header* sólo contiene información de prioridad.

- 1: *default value*.
- FFF: reservado.
- Los otros valores quedan disponibles para el uso de identificadores de VLAN.

Capítulo V

# Redes de gran alcance: Internet

*José M. Barceló Ordinas*
*Jordi Íñigo Griera*
*Enric Peig Olivé*

## Introducción

Lo que se conoce como red Internet es un conjunto heterogéneo de redes interconectadas. Precisamente, es la capacidad de homogeneizar lo que de hecho es heterogéneo, lo que ha catapultado la red Internet a su estatus actual.

Los protocolos que distinguen la red Internet como una unidad son el IP y el TCP. De hecho, estos últimos son sólo dos de entre la media docena de protocolos que se necesitan para hacer funcionar la red Internet; sin embargo, son los más importantes. Por este motivo, a todos en conjunto se les llama normalmente *pila* (*stack*) *TCP/IP* (*Tansmission Protocol Suite/Internet Protocol*).

La red ARPANET de Estados Unidos es la precursora de la red Internet actual. Desde 1969, ARPANET financió la interconexión de nodos de conmutación de paquetes por medio de líneas punto a punto dedicadas. Durante la primera década de funcionamiento permitió a un ingente conjunto de investigadores desarrollar y perfeccionar las técnicas para la gestión y el uso de las redes de conmutación de paquetes. Hacia el año 1979, la pila TCP/IP se empieza a perfilar como el conjunto de protocolos de futuro de la red y, al final de 1982 todos los nodos de ARPANET ya habían adoptado el TCP/IP.

Rápidamente, ARPANET deviene el nudo de interconexión principal de otras redes de datos, sobre todo de los académicos. Los ordenadores militares, que también habían formado parte de la red inicial y de su desarrollo, se dividen y forman una red propia (la MILNET). A mediados de los años ochenta, el éxito del TCP/IP queda demostrado por el hecho de que la nueva red de conmutación de paquetes para la investigación del NSF de Estados Unidos financia la NSFNET. Con el tiempo, todas las redes académicas, en primer lugar las de Estados Unidos, des-

pués las de Europa y más tarde las del resto del mundo se acabarán conectando a esta red. La red de redes, Internet, empezó su crecimiento vertiginoso –casi explosivo– hacia el año 1986.

A principios de los años noventa, las principales universidades ya formaban parte de Internet y, cuando el *boom* de Internet se preveía que se frenaría porque la mayoría de las universidades ya se habían conectado a la misma, las empresas empezaron a ver el potencial de la red, en primer lugar como medio de interconexión y, un poco más tarde, como herramienta de *marketing* (en 1993 aparece el HTTP, el protocolo del WWW).

De hecho, Internet presupone que las máquinas que se conectan ya están de alguna manera en una red local. Internet, como su nombre indica (es un apócope de Internetworking), se preocupa de eso, de interconectar redes, no equipos aislados. Por eso va bien tener nociones de redes locales antes de entrar en el estudio de IP.

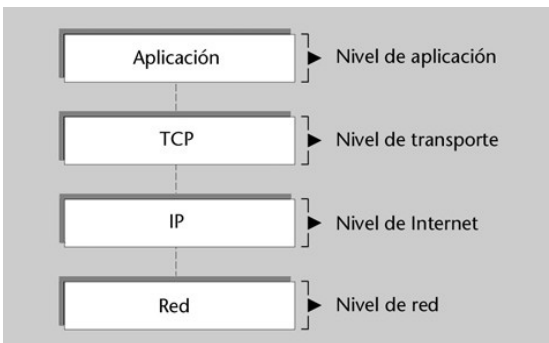
En este capítulo presentamos el mapa de protocolos completo y nos centramos en el protocolo de red, IP.

## 1. Estructura de protocolos en Internet

El modelo Internet gira en torno a los protocolos TCP/IP. De hecho, podríamos considerar que el modelo de la red Internet consta sólo de cuatro partes o niveles; es decir, todo lo que hay por debajo del IP, el IP, el TCP y todo lo que hay por encima del TCP:

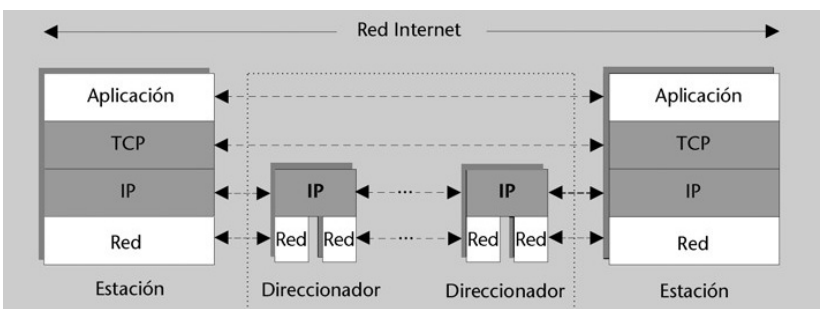
1) Nivel de red: por norma general, está formado por una red LAN, o WAN (de conexión punto a punto) homogénea. Todos los equipos conectados a Internet implementan dicho nivel. Todo lo que se encuentra por debajo del IP constituye el nivel de red física o, simplemente, nivel de red.
2) Nivel IP o nivel Internet (nivel de *Internetworking*): confiere unidad a todos los miembros de la red y, por consiguiente, es el que permite que todos se puedan interconectar, con independencia de si se conectan a la misma por medio de línea telefónica, ISDN o una LAN Ethernet. El encaminamiento y la asignación de direcciones constituyen sus principales funciones. Todos los equipos conectados a Internet implementan este nivel.

3) Nivel TCP o nivel de transporte: confiere fiabilidad a la red. El control de flujo y de errores se lleva a cabo principalmente dentro de este nivel, que sólo es implementado por los equipos usuarios de la red Internet o por los terminales de Internet. Los equipos de conmutación (encaminadores o *routers*) no lo necesitan.
4) Nivel de aplicación: engloba todo lo que hay por encima del TCP; es el nivel que corresponde a las aplicaciones que utilizan Internet: clientes y servidores de WWW, correo electrónico, FTP, etc. se encuentran dentro de este nivel, que sólo es implementado por los equipos usuarios de la red Internet o los terminales de Internet. Los equipos de conmutación no lo utilizan.




Modelo Internet.

Sólo los equipos terminales implementan todos los niveles; los equipos intermedios únicamente implementan el nivel de red y el nivel IP:

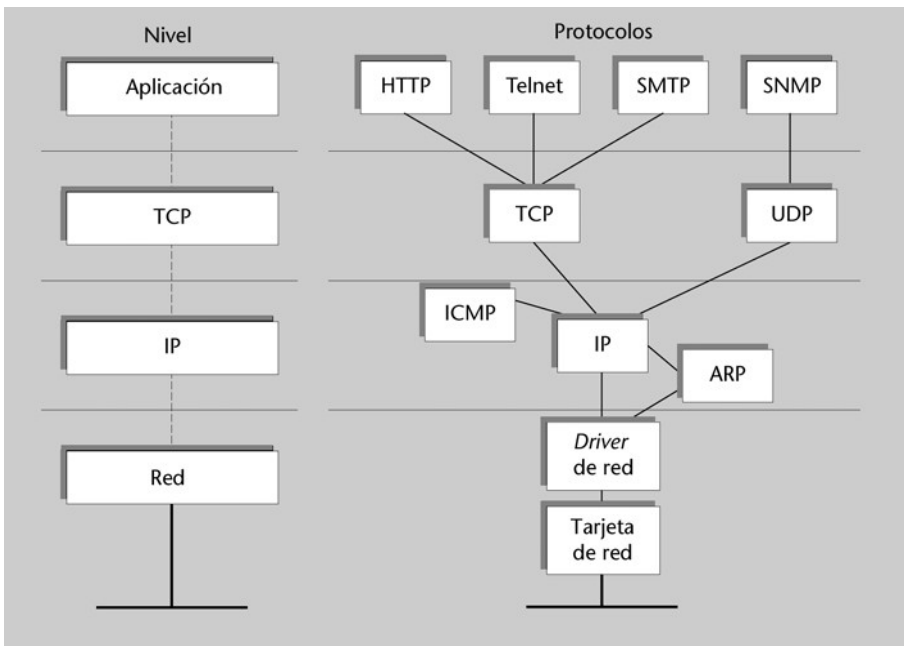



Situación de los niveles de la Red Internet.

En este capítulo veremos el nivel IP y cómo se "conecta" a los niveles bajos.

### 1.1. Protocolos de Internet

En cada uno de los niveles de la red expuestos encontramos protocolos diferentes. La situación relativa de cada protocolo en los diferentes niveles se muestra en la figura siguiente:




Hay que tener en cuenta que el concepto *nivel* no existe en Internet. Este concepto se utiliza en otros modelos de red, como la OSI. No obstante, como es un concepto útil, en este capítulo lo utilizaremos para plantear el estudio de los diferentes protocolos.

### 1.2. Encapsulamiento

Será preciso saber, por ejemplo, qué implicaciones tiene el hecho de que un protocolo como el TCP, como se ve en la figura del subapartado anterior, vaya sobre otro protocolo, el IP, y que este último vaya, a su vez, sobre Ethernet.

Cada uno de estos protocolos funciona con unas estructuras fundamentales

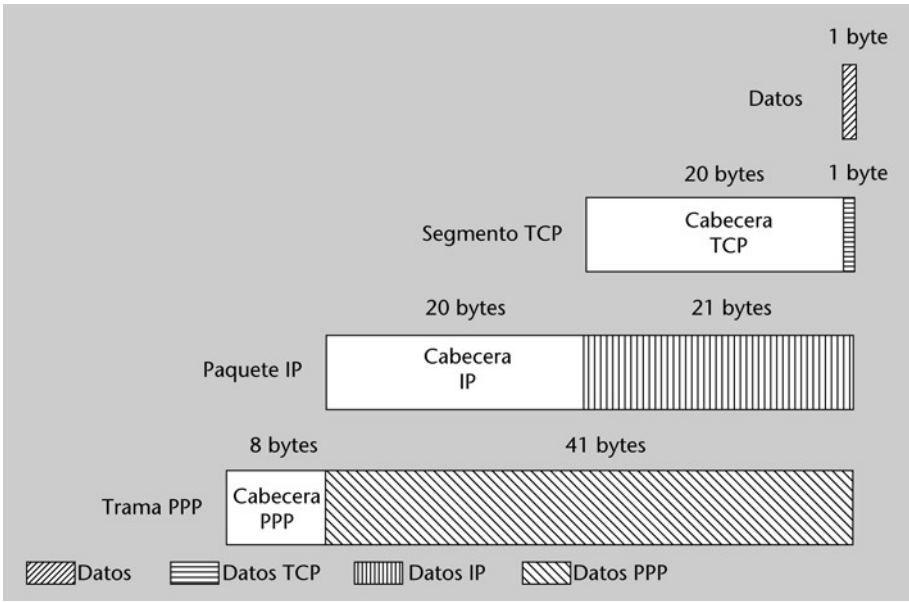
que genéricamente se conocen como PDU (*Protocol Data Units*). Sin embargo, en cada nivel se utilizan nombres diferentes para denominar lo que, de hecho, tiene funciones equivalentes:

- Las PDU Ethernet o PPP se denominan tramas.
- Las PDU del nivel de interconexión (IP o ARP) se suelen denominar *paquetes*. Sin embargo, por lo general las PDU ICMP se denominan *mensajes*, seguramente porque viajan en paquetes IP.
- En el nivel de transporte, se habla de *segmentos* en TCP, y de *datagramas* en UDP.
- En niveles superiores que utilizan UDP, por norma general se utiliza la palabra PDU (SNMP-PDU, por ejemplo). En el caso del TCP, el servicio que proporciona a las aplicaciones es el flujo de bytes sin estructura (*byte stream*). Por tanto, el concepto *PDU* deja de tener sentido en el nivel superior a TCP.

Todavía no hemos respondido, sin embargo, a la pregunta inicial. El resultado de los diferentes encapsulamientos en cada nivel es que, cuando el nivel superior decide transmitir cierta información, se provoca una cascada de PDU que va descendiendo hasta el nivel inferior, que finalmente es el que transmite físicamente los bits que resultan del mismo.

**Longitud de la trama necesaria para transmitir un *byte* de información**

Veamos, a modo de ejemplo, la emulación de terminal que efectúa el programa Telnet y el protocolo que lleva el mismo nombre. Imaginemos que pulsamos una tecla sobre la ventana del programa telnet. Esta acción llega en forma de un byte único al nivel TCP. Este nivel encapsula dicho byte en un segmento TCP, que tendrá una cabecera de 20 bytes y un contenido que será el byte correspondiente a la tecla pulsada. Estos 21 bytes se transportarán dentro de un paquete IP que, generalmente, formará un paquete con 20 bytes de cabecera más el contenido ya mencionado de 21 bytes. Este paquete de 41 bytes irá, a su vez, dentro de una trama que lo transportará por su soporte físico de transmisión. Si el soporte de transmisión es una línea telefónica con un módem y utilizamos PPP, el resultado puede ser que le añadamos 8 bytes más. En el caso de estar en una LAN iEEE8023, serían 26 bytes más, en lugar de 8.
Si observamos el caso de una trama PPP, de todo ello resulta una trama de 49 bytes (8 + 20 + 20 + 1) de longitud para transmitir uno solo. La figura siguiente muestra esta estructura en:




Pirámide PDU.

## 2. El IP (*Internet protocol*)

IP y TCP son un par de protocolos muy compenetrados. El IP es un protocolo de interconexión de red orientado a datagrama. Por lo tanto, no dispone del concepto de circuito virtual, de manera que no es capaz de recuperar tramas perdidas, ni de garantizar que las tramas se entregarán en el orden correcto –ya que los paquetes pueden seguir caminos diferentes y, por lo tanto, sufrir retrasos diferentes–, ni que el ritmo de recepción sea el adecuado para que el receptor procese convenientemente los datos.

El IP es del tipo *best effort* (podríamos traducirlo como 'con la mejor intención', o 'quien hace lo que puede no está obligado a más'). Evidentemente, cuando utilizamos una red no siempre nos podemos conformar con conseguir que la información llegue cuando la red puede. Es aquí donde interviene el TCP, que es responsable de hacer que la información llegue en las condiciones de fiabilidad deseadas.

Hemos visto protocolos con una filosofía similar a la del *best-effort*: los protocolos de LAN, como es el caso de Ethernet. Hay detractores acérrimos y también defensores incondicionales de esta filosofía, y seguramente los dos grupos tienen una parte de razón. La red X.25 es un ejemplo casi opuesto a esta filosofía que da a sus usuarios unos niveles de fiabilidad y flexibilidad razonablemente elevados, gracias a la utilización de la conmutación de paquetes con circuitos virtuales.
Ciertamente, la historia da la razón a los defensores de la filosofía *best-effort*, ya que el X.25 actualmente se considera casi una tecnología obsoleta, mientras que el IP está de moda. Las razones se tienen que buscar, como siempre, en el coste. La filosofía IP permite implementar redes con un coste mínimo: pensad que TCP sólo se tiene que implementar en los terminales de la red, y no entre los nodos de conmutación. Por lo tanto, los nodos de conmutación IP son mucho más simples que los de X.25.

El trabajo principal de IP es el encaminamiento de paquetes. O sea, decidir a cada momento y en cada lugar qué camino tienen que tomar los paquetes, asumiendo que en la red hay multiplicidad de caminos entre los dispositivos de encaminamiento. En este apartado nos centraremos en el punto de vista de las estaciones, y veremos la asignación de direcciones y el formato del paquete IP. Más adelante analizaremos los algoritmos de encaminamiento que implementan estos dispositivos.

## 2.1. Direcciones IP

Las direcciones IP son únicas para cada máquina. Para ser precisos, cada dirección es única para cada una de las interfaces de red IP de cada máquina. Si una máquina tiene más de una interfaz de red (como por ejemplo, un encaminador), necesitará una dirección IP para cada uno.

Las direcciones IP tienen una longitud de 32 bits (4 bytes).

Para representar una dirección, se suele escribir los 4 bytes en decimal y separados por puntos. Por ejemplo, la dirección 10010011010100111001100101100100 se escribe:

147.83.153.100

La numeración en IP sigue una filosofía jerárquica. Cada dirección está formada por dos partes. Una corresponde a la red donde está la estación, llamada *identificador de red*, y la otra, a la propia estación, denominada *identificador de estación*.

Para conseguir que no haya ninguna dirección igual, Internet dispone de una organización llamada *Internet Network Information Center* o *InterNIC* que se dedica a esta tarea. Actualmente, esta entidad delega la responsabilidad de la asignación de direcciones a entidades regionales. Las direcciones se asignan por grupos o redes, no individualmente.

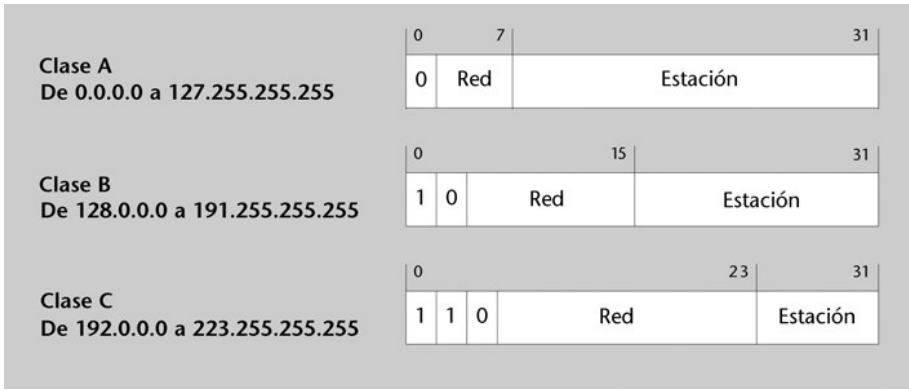
Los tipos de redes que tienen cabida en Internet se distinguen por la cantidad de estaciones que pueden soportar, y son las siguientes:

1) Las redes de clase A reservan el primer byte como identificador de red y los tres restantes como identificadores de estación. El primer bit del primer byte vale 0; por lo tanto, en Internet sólo puede haber 128 redes de clase A (con $2^{24}$ estaciones cada una como máximo). Hace mucho tiempo que ya no queda ninguna para asignar.
2) Las redes de clase B tienen 16 bits para cada campo. Los dos primeros bits del identificador de red valen 1 0; por lo tanto, hay 16.384 ($2^{14}$) redes de, como mucho, 65.536 estaciones. De clase B, no queda ninguna para asignar.
3) Las redes de clase C reservan 24 bits para el identificador de red (con los tres primeros bits valiendo 1 1 0) mientras que los 8 restantes son para el identificador de estación.
4) Las redes de clase D están destinadas al tráfico *multicast*. Los cuatro primeros bits valen 1 1 1 0, y el resto se destinan al identificador del grupo *multicast*. No se asignan a máquinas concretas.
5) Finalmente, las redes de clase E, que actualmente no tienen ningún uso concreto. Los cuatro primeros bits valen 1 1 1 1.

Una vez conocida una dirección, es fácil saber si corresponde a una red de clase A, B o C, como se puede ver en la figura siguiente.

La clase A está pensada para grandes empresas o corporaciones, con muchos terminales por identificar. La clase B es para corporaciones medias, y la clase C para entornos mucho más pequeños.

La dirección comentada anteriormente (147.83.153.100) es de clase B.




## 2.2. Subredes

Cuando un administrador de sistemas recibe el encargo de gestionar un conjunto de direcciones, se puede encontrar que tenga una sola red de muchas estaciones (una clase A, por ejemplo), y que en su organización haya muchas LAN diferentes. Si se pudiera subdividir la red IP original en diversas subredes, eso le proporcionaría mucha flexibilidad y una mejor gestión de la red. El mecanismo que lo permite es la máscara de red y se habla de *subnetting*.

Mediante una máscara de 32 bits, definiremos los bits que identifican la red (bits en 1) y los que identifican la estación (bits en 0). Normalmente los bits 1 y los 0 son consecutivos, pero no necesariamente tiene que ser así.

A continuación definimos de nuevo el concepto *identificador de red*, adaptándolo a la máscara:

> El identificador de red es la porción de dirección IP que encaja con los bits a 1 de la máscara.

Tal como se define la máscara, las clases A, B y C tienen una máscara asociada:

A: 11111111 00000000000000000000000
B: 1111111111111111 0000000000000000
C: 11111111111111111111111 00000000

Pero si ampliamos el número de unos, lo que hacemos es convertir bits de estación en bits de red. Por ejemplo, si tenemos la red clase A 11.0.0.0:

00001011.00000000.00000000.00000000

y definimos la máscara

11111111.11100000.00000000.00000000

quiere decir que los 3 bits primeros del segundo byte se dedican a subred y no a estación. Por lo tanto, puedo crear 8 subredes diferentes, y en cada una tener $2^{21}$-2 estaciones, porque 21 es el número de bits a 0 de la máscara:

Primera subred: 00001011.000 00000.00000000.00000000
Segunda subred: 00001011.001 00000.00000000.00000000
Tercera subred: 00001011.010 00000.00000000.00000000
etc.

Así, la dirección 11.32.0.1 es la estación 1 de la subred 2 de la red clase A 11. Pero también puede ser que no haya *subnetting* y sea la estación 32.0.1 de la red 11. Cuando hay *subnetting* se tiene que proporcionar la máscara de red.

Una notación alternativa de la máscara es dando el número de bits 1 de la máscara. Así pues, la máscara 255.255.255.0 es una máscara de 24 bits y la 255.255.255.128 es una máscara de 25 bits. A veces podemos ver una dirección con el añadido de la máscara, por ejemplo:

147.83.153.100/24

Esta notación está bastante extendida, pero se ha tener en cuenta que sólo es útil para máscaras con 1 consecutivos.

En el ejemplo anterior, distinguiríamos los dos casos descritos porque el primero sería 11.32.0.1/11 y el segundo 11.32.0.1/8.

Si no se da la máscara, es que no hay *subnetting*, y, por lo tanto hay que considerar las clases básicas A, B y C.

La máscara permite a una estación decidir si el destino al que tiene que transmitir un paquete se encuentra dentro de la misma red de área local que ella o si, por contra, se encuentra en una LAN remota y, por lo tanto, tiene que delegar la transmisión a algún equipo de su misma LAN (el encaminador) para que se encargue de hacer llegar el paquete a su destino.

Todas las estaciones de una misma red de área local tienen que utilizar el mismo identificador de red y todas las estaciones tienen que tener la misma máscara.

Si tenemos dos estaciones con las direcciones 147.83.153.100 y 147.83.153.200, podemos deducir que están interconectadas directamente (por una LAN) si la máscara de su red es 255.255.255.0. En cambio, deduciríamos que no están conectadas con la misma LAN si la máscara fuera, por ejemplo, 255.255.255.128.

## 2.3. Direcciones de propósito especial

Existen diferentes direcciones especiales, que no se utilizan para estaciones concretas. Son éstas:

1) Dirección de red: las direcciones de red se expresan con la dirección que tendría cualquier estación suya y con todos los bits del identificador de estación a 0. Por ejemplo, la red en que se encuentra la estación 147.83.153.100/24 es la 147.83.153.0/24 y la red en que se encuentra la estación 147.83.153.200/25 es la 147.83.153.128/25.
2) Dirección 0.0.0.0: señala el mismo ordenador que lo envía.Tiene dos funciones básicas:
   - Aparecer como dirección origen en paquetes IP generados por estaciones sin dirección IP asignada. Normalmente sólo sale mientras la estación trata de averiguar su dirección mediante protocolos como el RARP (*Reverse Address Resolution Protocol*), el BOOTP (*Bootstrap Protocol*) o el DHCP (*Dynamic Host Configuration Protocol*).
   - Servir al software de gestión de encaminamiento para indicar la ruta por defecto.
3) Dirección 127.0.0.1 (*loopback*): dirección no válida para los paquetes IP. El software de red la utiliza para transmitir paquetes a la máquina local (de hecho, los paquetes no son transmitidos, sino que son entregados en el des-

tino por el mismo sistema operativo). En realidad, los tres bytes del identificador de estación son irrelevantes. Esta dirección sólo tiene interés a la hora de programar aplicaciones; los sistemas de red no verán nunca que ningún paquete viaje por la red con esta dirección como origen o destino.

4) Dirección 255.255.255.255 (*broadcast*): sólo es válida como dirección de destino de un paquete. Se utiliza para transmitir paquetes a todas las estaciones localizadas dentro de la misma LAN que la máquina de origen. Hay una versión equivalente, que es el *broadcast* dirigido. En este segundo caso, el paquete es recibido por todas las máquinas de una LAN especificada por el identificador de red. El identificador de estación tiene que ser todo 1. Por lo tanto, para enviar un *broadcast* a la red 147.83.153.0 con la máscara 255.255.255.0 (o 147.83.153.0/24) podemos utilizar la dirección 255.255.255.255 si estamos dentro de la red 147.83.153.0, o bien la 147.83.153.255 si estamos en una estación remota. El primer caso lo llamaremos *broadcast* local, y el segundo, *broadcast* remoto.
5) Todas las direcciones incluidas en estos tres rangos:
   - 10.0.0.0/8
   - 172.16.0.0/16 a 172.31.0.0/16
   - 192.168.0.0/24 a 192.168.255.0/24

Estas direcciones, que corresponden respectivamente a redes de clase A, B y C, no son asignadas por Internet, ni nunca lo serán. Se consideran direcciones privadas. Se utilizan en redes que trabajan con los protocolos TCP/IP pero que no está previsto que se conecten a Internet, o en caso de que sí lo hagan, estén parcialmente ocultas detrás de *proxies* o de cortafuegos, los cuales se encargarán de traducir su dirección a otra que esté en los rangos de direcciones públicas.

Por lo tanto, en las redes de clase A, B y C no tenemos $2^{24}$, $2^{16}$ o $2^8$ estaciones posibles, respectivamente, sino $2^{24}$-2, $2^{16}$-2 y $2^8$-2. Las direcciones que tienen todos los bits correspondientes a la estación a cero y las que los tienen todos a uno no son direcciones válidas para estaciones, ya que corresponden a la dirección de red y a la dirección de *broadcast* remoto, respectivamente.

## 2.4. CIDR y *supernetting*

El mecanismo de *subnetting* es secundario a la clase. O sea, a la vista de los primeros bits sabemos si una dirección es de clase A o B o C, y después podemos mirar si la máscara –si la hay– es la que toca o es mayor, y por lo tanto hay *subnetting*.

Más adelante, sin embargo, se propuso utilizar el mismo mecanismo pero no para partir redes sino para juntar. Por ejemplo las cuatro redes clase C siguientes:

200.10.8.0
200.10.9.0
200.10.10.0
200.10.11.0

Tienen en común los primeros 20 bits de los 24 que forman el identificador de red. Por lo tanto, se puede utilizar esta notación más compacta:

200.10.8.0/20

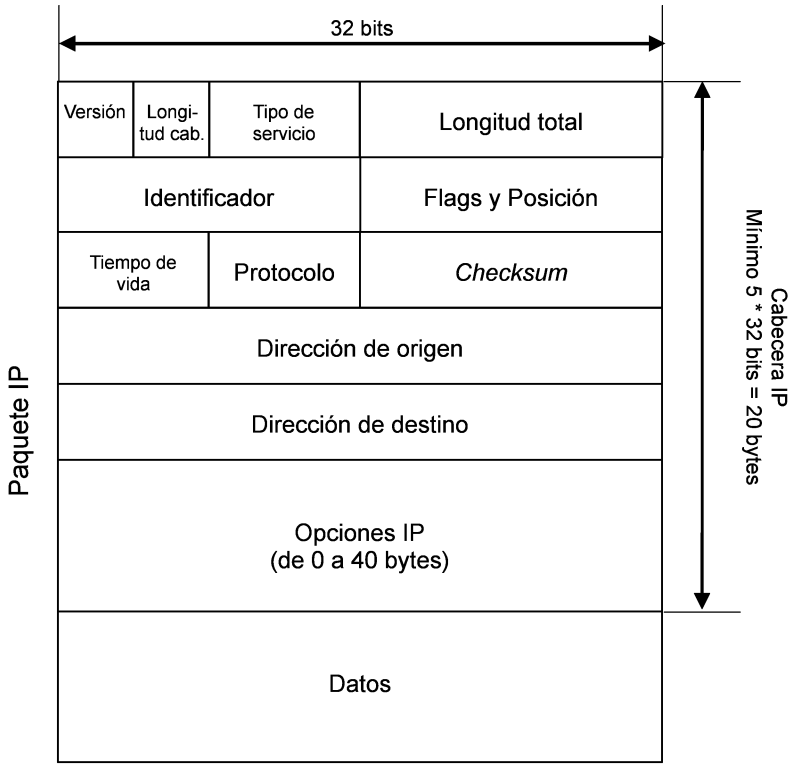
O sea, esta dirección no corresponde a una subred sino a un grupo de cuatro redes diferentes. Hemos extrapolado el concepto de máscara, y además, ahora ya no tenemos que saber de qué clase es. Aplicamos la máscara directamente y podemos separar el identificador de red del identificador de estación.

Utilizar la máscara de red para juntar redes se denomina *supernetting*.

Una de las ventajas del *supernetting* es que simplifica el encaminamiento. Si una organización necesita 1.000 direcciones IP pero no le pueden dar ninguna red clase B porque no quedan libres, le dan cuatro redes de clase C (cada clase C permite 254 estaciones), pero con un solo identificador de red. Sin poder hacer *supernetting*, en cada encaminador (como veremos más adelante) tendría que haber cuatro entradas, una para cada red. Con el *supernetting*, tiene bastante con un solo identificador de red.

Como hemos dicho, ya no nos hace falta la clase. Para tomar decisiones de encaminamiento, simplemente se aplica la máscara para separar la red (ya sea superred, red estándar o subred) de la estación. Por eso se habla de CIDR (*classless interDomain routing*).

## 2.5. El formato del paquete IP




A continuación describiremos los campos que componen el paquete IP:

- **Versión**: siempre vale cuatro (0100), para los paquetes de la versión actual (IPv4).
- **Longitud de la cabecera**: da la longitud de la cabecera en palabras de 32 bits (4 bytes). Por tanto, el número de bytes de la cabecera tiene que ser múltiplo de 4. Asimismo, limita la longitud de la cabecera a 60 bytes ($15 \cdot 4$), puesto que 15 es el máximo que se puede expresar con cuatro dígitos binarios. Si no hay campo de opciones, la cabecera tiene 20 bytes; por tanto, el campo Longitud de la cabecer en este caso valdría 5 ($5 \cdot 4$ bytes = 20 bytes).
- **Tipo de servicio**: este campo se encuentra dividido en varios subcampos. Permite pedir un trato especial para el paquete y raramente se implementa.
- **Longitud total del paquete**: da la longitud total del paquete (cabecera incluida) en bytes. Como este campo es de 16 bits, un paquete IP no puede tener más de 65.535 bytes ($2^{16}$-1).

- **Identificación del paquete**: contiene un identificador que es diferente para cada paquete que genera la estación.
- **Indicadores (*flags*) y posición de este fragmento**: estos campos permiten gestionar la fragmentación de paquetes.
- **Tiempo de vida o TTL (*Time To Live*)**: indica el número máximo de encaminadores que puede cruzar el paquete. Se utiliza para evitar que un paquete pueda quedar dando vueltas indefinidamente dentro de la red en caso de que haya algún problema al entregarlo. Cada encaminador disminuye el campo en uno; cuando uno de ellos detecta un paquete con TTL = 1, lo elimina y envía a quien lo ha emitido un mensaje de error por medio de un mensaje ICMP.
- **Protocolo**: identifica el tipo de protocolo que transporta el paquete. Los valores más comunes de los diferentes tipos de protocolos son los siguientes:

| | |
|---|---|
| TCP | 6 |
| UDP | 17 |
| ICMP | 1 |

- ***Checksum***: realiza el control de errores en la cabecera. El campo de datos no queda protegido por ningún *checksum*; es responsabilidad de los usuarios del IP (TCP, UDP, etc.) el control de los posibles errores en su contenido.

**Algoritmo de cálculo del *checksum***

El *checksum* (o comprobación por adición) es básicamente la suma aritmética de los bytes de la cabecera agrupados de dos en dos (si el resultado necesita más de 16 bits, se suman los bits sobrantes al mismo resultado).

Este algoritmo, si bien es fácil y rápido de calcular, no se caracteriza por poseer unas grandes cualidades para la detección de errores. Ello, sumado al hecho de que el contenido del IP no tiene *checksum* y a otros factores (tales como que muchos sistemas no calculan el *checksum* de los paquetes UDP), demuestra que en el mundo de Internet no existe un interés especial por la detección de errores. Con frecuencia, éste ha sido un argumento en el que se han basado los detractores del protocolo IP para atacarlo.

La detección de errores dentro del campo de información es responsabilidad de quien entra la información. Los usuarios más habituales del IP son los protocolos TCP, UDP e ICMP, y todos protegen la información con un campo adicional de *checksum*.

- **Dirección de origen IP**: identifica la máquina que ha generado el paquete.
- **Dirección de destino IP**: identifica la máquina a la que va destinado el paquete.
- **Opciones**: existen diferentes servicios asignados a este campo, pero por lo general no se utilizan. Comentaremos algunos de los mismos cuando hablemos del ICMP Sea como sea, la longitud total está limitada a 40 bytes ($15 \cdot 4 - 20$). 15 es el número máximo que se puede poner en el campo Longitud de la cabecera, y 20 es lo que ocupan los campos fijos de la cabecera.

### *2.5.1. Fragmentación*

Para gestionar la fragmentación, IP destina 16 bits de la cabecera: 13 para el campo denominado *posición de este fragmento* y 3 para tres indicadores de un bit de los cuales se usan sólo dos:

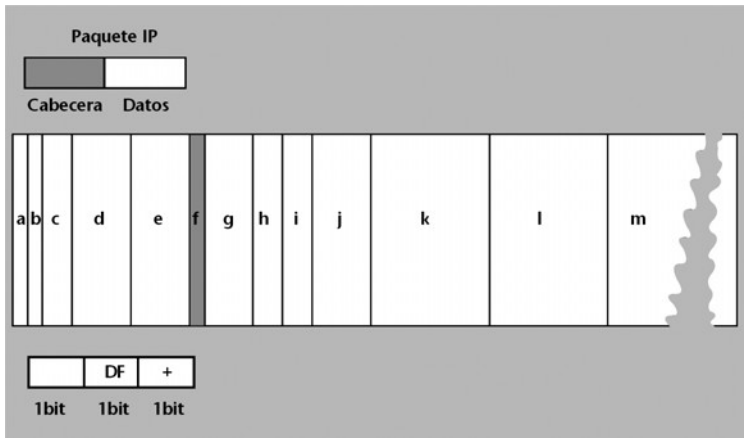
1) MF (*More Fragments*)
2) DF (*Don't Fragment*)

Los paquetes IP van siempre insertados en tramas de nivel de enlace o MAC. Una estación, cuando transmite un paquete IP, conoce la MTU (*Maximum Transmission Unit*) de su interfaz de red, y está restringida a transmitir paquetes IP que se puedan transmitir por su interfaz a la red, con una longitud nunca superior a esta MTU.

Si el paquete pasa por un encaminador conectado a una red con una MTU inferior al tamaño del paquete, es preciso fragmentarlo. Los fragmentos resultantes de la operación son paquetes como cualquier otro, pero que poseen las características siguientes:

1) Todos los fragmentos que provienen de un paquete fragmentado tienen el mismo identificador de paquete en las respectivas cabeceras IP.
2) Todos los fragmentos indican con el campo Posición de este fragmento a qué byte corresponde el primer byte de datos del fragmento dentro del paquete original. El primer fragmento tiene un cero en este campo.
3) El bit MF del campo Indicadores es 1 en todos los fragmentos a excepción del último, que, por ello se suele denominar *hay más*.

El resto de los campos de la cabecera se copia íntegramente del paquete original a los fragmentos que resultan de la fragmentación, excepto el indicador de longitud del paquete y el *checksum*, que se deben calcular de nuevo.




En principio, podríamos suponer que, cuando los fragmentos vuelven a una red con una MTU suficiente, el encaminador que los recibe los reunifica. Sin embargo, no se hace así, en primer lugar porque la ley no escrita de Internet, según la cual el encaminador debe llevar a cabo el mínimo trabajo necesario, no se cumpliría. En segundo lugar (y más importante), porque nadie garantiza que todos los fragmentos sigan el mismo camino. La única estación capaz de recomponer los fragmentos es la de destino.

La estación de destino, cuando recibe un fragmento, reserva suficiente memoria para alojar el mayor paquete IP posible (65.535 bytes), puesto que no tiene manera de saber cuál es el tamaño del paquete original. A partir de aquí, pone en marcha un temporizador[1] y empieza a recolectar los fragmentos según su identificador. Cuando se han recibido todos los fragmentos, se entrega el paquete a la aplicación (protocolo) correspondiente. En caso de que el temporizador salte, se descartan todos los fragmentos llegados hasta aquel momento. El nivel superior (TCP) es el responsable de pedir una retransmisión cuando convenga, puesto que el IP no dispone de ningún mecanismo para pedirla.

1. Este temporizador permite detectar la pérdida de un fragmento por el camino y, por tanto, la pérdida de la memoria reservada.

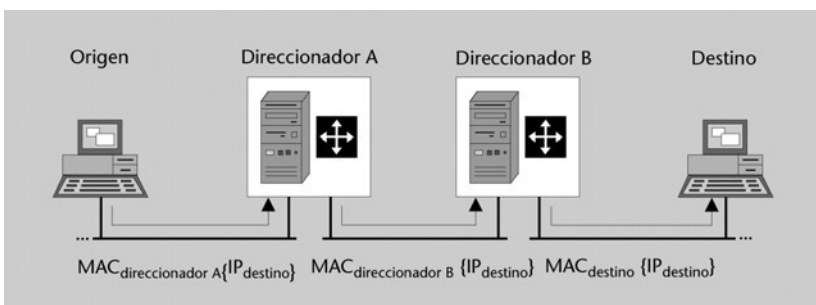
**Paquetes no fragmentables**

El bit DF (*Don't Fragment*), especifica que el paquete no debe fragmentarse. Si fuera preciso fragmentarlo, el paquete sería descartado y se enviaría un mensaje ICMP indicando el error a la estación originadora del paquete. Usualmente, se activa el bit DF sólo cuando la fragmentación no es deseable como, por ejemplo, cuando enviamos paquetes a estaciones con la pila TCP/IP implementada en ROM, puesto que entonces se implementa sólo una mínima parte de la funcionalidad de TCP/IP.

## 2.6. Encaminamiento y encaminadores

Las atribuciones principales de los encaminadores son interconectar dos o más subredes IP y encargarse de direccionar el tráfico destinado a estaciones remotas.

Cada encaminador de la red Internet debe ser capaz de decidir (en una fracción de segundo) a dónde debe dirigir un paquete basándose exclusivamente en la dirección de destino, y en el conocimiento que tiene de la red y de su posición (la del encaminador) dentro de Internet. De hecho, cada encaminador no decide la ruta entera del paquete, sino sólo el trozo de ruta en que participa: el salto (*hop*) siguiente. Entre el origen y el destino tenemos un número variable de saltos, y en cada uno de ellos participan un par de encaminadores (el emisor y el receptor del paquete) a excepción del primero y el último, en que participan las estaciones emisora y transmisora, respectivamente. Las conexiones entre encaminadores, o bien entre encaminador y estación terminal, las componen LAN o enlaces punto a punto.




Transporte de un paquete IP a través de tres subredes.

En la figura anterior, podemos ver un paquete IP que cruza tres redes de área local En cada una el paquete va encapsulado dentro de una trama de un tipo

adecuado a la subred que cruza (en el dibujo podrían ser tres LAN Ethernet). Las direcciones de origen y de destino del nivel LAN en cada tramo son diferentes. En cambio, la dirección de origen y la de destino de cada paquete IP no varían. La situación plantea dos problemas y, para resolverlos, es preciso efectuar los pasos siguientes:

- Averiguar cuál es la correspondencia entre direcciones IP y MAC (sólo en los casos en que el enlace entre los encaminadores sea una LAN). Ello nos permitirá enviar los paquetes IP a un encaminador a pesar de que el paquete IP no tenga su dirección. El mapeado IP-MAC se efectúa de manera automática por medio del protocolo ARP.
- Decidir en cada momento cuál es el encaminador siguiente (el salto siguiente).

### *2.6.1. La tabla de encaminamiento*

El encaminamiento se lleva a cabo a partir de tablas de encaminamiento que disponen de información limitada, pero suficiente para permitir la interconexión de todas las subredes que componen Internet. Cada equipo conectado a una red IP necesita una tabla de encaminamiento.

**Tablas de encaminamiento**

Estas tablas se introducen dentro del encaminador por medio de un terminal que se conecta al mismo directamente (consola). Asimismo, existen protocolos que permiten que las tablas se actualicen automáticamente cuando se detectan cambios de topología. Los más utilizados son el RIP (*routing information protocol*) y el OSPF (*open shortest path first*).

El encaminador decide la ruta de los paquetes consultando su tabla de encaminamiento. Las estaciones terminales también necesitan una tabla de encaminamiento, aunque sólo sea para poder descubrir si se están comunicando con una estación local de la LAN (y, por consiguiente, se pueden comunicar directamente con la misma) o si el paquete IP debe ir a una estación remota (conectada a otra LAN) y, por tanto, deben dejarlo en manos del encaminador.

*Tabla de encaminamiento de una estación con una única interfaz*[2]

Empezamos estudiando la tabla de encaminamiento de una estación conectada a una LAN (dirección 147.83.153.103/24) con una única tarjeta Ethernet. Por norma general, una tabla de encaminamiento dispone también de información sobre las "cualidades" de cada una de las rutas descritas, que no aparecen en el ejemplo:

| | Dirección | Máscara | Encaminador | Interfaz |
|---|---|---|---|---|
| **1** | 147.83.153.103 | 255.255.255.255 | 127.0.0.1 | *Loopback* |
| **2** | 127.0.0.0 | 255.0.0.0 | 127.0.0.1 | *Loopback* |
| **3** | 147.83.153.0 | 255.255.255.0 | 147.83.153.103 | ether0 |
| **4** | 255.255.255.255 | 255.255.255.255 | 147.83.153.103 | ether0 |
| **5** | 0.0.0.0 | 0.0.0.0 | 147.83.153.5 | ether0 |

Tabla de encaminamiento de la estación.

Ésta es una tabla típica que podemos encontrar en una estación conectada a Internet. Las filas de la tabla o reglas se consultan no en el orden en que el comando route nos las presenta, sino en orden de máscara decreciente (en primer lugar, las entradas con 32 bits, etc.). De las entradas que forman la tabla es preciso señalar lo siguiente:

- La primera entrada y la segunda permiten transmitir paquetes IP a las direcciones 147.83.153.103 y a todas las direcciones que empiecen por 127 (fijaos en la máscara de las dos entradas). En ambos casos los paquetes se envían a la interfaz virtual *loopback* (la interfaz con que se conoce el ordenador local desde el mismo ordenador local). Ninguno de los paquetes que se direccione con alguna de estas dos reglas saldrá a la red: será cortocircuitado directamente por el sistema operativo y se entregará al proceso de destino sin que nunca llegue a ninguna tarjeta de comunicaciones.

2. Una interfaz es una tarjeta de co- nexión física a la red. Ejemplos de interfaz serían una tarjeta Ethernet (por ejemplo, la Ether0) o un puerto serie. En el ejemplo vemos una interfaz loopback, que es un caso espe- cial, como veremos en el texto. Para saber qué interfaces hay disponibles en un sistema, en UNIX o en Windows NT sólo es necesario invocar el comando $ ifconfig, y en Windows 2000 o superior C:> ipconfig.

- La tercera entrada será adoptada por todos los paquetes destinados a la red local. El campo Encaminador es el mismo que la dirección local, lo que significa que no se delegará su transmisión a ningún encaminador.
- La cuarta entrada tiene una importancia relativa. Nos indica que los *broadcasts* IP se restringirán a la red local.
- La quinta entrada (0.0.0.0/0) permite a la estación comunicarse con estaciones remotas. Notad que la máscara no tiene ningún bit a 1. Ésta es la ruta por defecto. El encaminador establecido para acceder a estaciones remotas queda identificado en la tabla con la dirección 147.83.153.5.

Toda esta información se calcula a partir de tres datos que se introducen en la configuración de red:

1) La dirección local (en el ejemplo: 147.83.153.103).
2) La máscara de la LAN local (en el ejemplo: 255.255.255.0).
3) La dirección del encaminador (en el ejemplo: 147.83.153.5).

En casos más complejos (más de una interfaz de acceso a Internet, más de un encaminador en la red local, etc.), la información deberá modificarse con el comando route. Asimismo, existen otros mecanismos que no requieren la intervención humana y que permiten descubrir encaminadores que no se han configurado previamente por medio del protocolo ICMP.

En un encaminador, las tablas de encaminamiento tienen más entradas que las de una estación; sin embargo, el funcionamiento es el mismo que hemos visto en el caso anterior. Observad una que conecta la red 147.83.153.0/24 con el exterior por medio de la red 147.83.30.0/24.

| | **Dirección** | **Máscara** | **Encaminador** | **Interfaz** |
|---|---|---|---|---|
| **1** | 147.83.153.5 | 255.255.255.255 | 127.0.0.1 | *Loopback* |
| **2** | 147.83.30.2 | 255.255.255.255 | 127.0.0.1 | *Loopback* |
| **3** | 127.0.0.0 | 255.0.0.0 | 127.0.0.1 | *Loopback* |
| **4** | 147.83.153.0 | 255.255.255.0 | 147.83.153.5 | ether1 |
| **5** | 147.83.30.0 | 255.255.255.0 | 147.83.30.2 | ether0 |
| **6** | 255.255.255.255 | 255.255.255.255 | 147.83.153.5 | ether1 |
| **7** | 0.0.0.0 | 0.0.0.0 | 147.83.30.1 | ether0 |

Tabla de encaminamiento de un encaminador.

Prácticamente no observamos nada nuevo. Simplemente se han duplicado las entradas específicas de interfaz. El encaminador también tiene una entrada por defecto. En este caso, todo el tráfico no destinado a las dos redes a que está directamente conectado se direcciona hacia otro encaminador situado en la red 147.83.30.0 (el 147.83.30.1).

## 3. El ARP (*address resolution protocol*)

Hemos visto, cuando describíamos el funcionamiento de los encaminadores, que éstos necesitan saber cuál es la dirección MAC correspondiente a una dirección IP. De este mapeo de direcciones se encarga el ARP.

El ARP se basa en sólo dos paquetes:

- El paquete de petición ARP
- El paquete de respuesta ARP

El funcionamiento básico del ARP es el siguiente: Suponemos que nos queremos conectar vía telnet a una máquina que está en nuestra misma LAN. Por ejemplo, desde 147.83.153.100 nos queremos conectar a 147.83.153.103. Aplicando la máscara de red a la dirección solicitada, la estación origen deduce que la de destino se encuentra en su misma red (o subred) y, por lo tanto, no hay que delegar en ningún encaminador. Ahora, para poder enviarle las tramas donde tienen que ir los paquetes IP que genera la aplicación telnet, necesita conocer su dirección MAC. Para tenerla, emite una petición ARP. Se trata de un paquete encapsulado directamente sobre una trama Ethernet (con tipo = 0x0806). Como dirección de destino lleva la dirección *broadcast* (FF:FF:FF:FF:FF:FF) para que llegue a todas las estaciones de la LAN, y como contenido, lleva la dirección IP para la cual se quiere saber su dirección MAC correspondiente. La estación que reconoce su IP en la petición ARP, responde con una respuesta ARP dirigida al origen de la petición. De hecho, el contenido de esta respuesta es irrelevante. Lo único importante es la dirección MAC de origen de la trama.

Lo mismo sucede cuando la estación deduce que la IP solicitada por alguna aplicación corresponde a una estación remota, o sea, de fuera de su LAN. En este caso, las tramas se tienen que enviar al encaminador para que se encargue de sacar

los paquetes de la LAN, hacia su destino. Para hacerlo, la estación origen tiene que averiguar la dirección MAC de encaminador, y lo hace con el mismo mecanismo descrito anteriormente. O sea, emite una petición ARP que contiene la dirección IP del encaminador, el cual le responde con su dirección MAC. A continuación la estación origen genera una trama Ethernet destinada al encaminador que contiene el paquete IP que se tiene que enviar fuera.

Es fácil deducir que en el funcionamiento normal de una estación, las peticiones ARP pueden ser muy habituales. Y en muchos casos, siempre pidiendo por las mismas direcciones (como por ejemplo, la del encaminador). Para evitar este alud de peticiones repetidas, cada estación dispone de una tabla con las parejas ya obtenidas, de modo que si una dirección IP ya está, no hace falta volver a pedir su MAC correspondiente.

Esta tabla se conoce como caché ARP, y tiene el aspecto siguiente:

| | Dirección IP | Dirección MAC | Interfaz |
|---|---|---|---|
| **1** | 147.83.153.103 | 08:00:00:10:97:00 | ether() |
| **2** | 147.83.153.5 | 00:0c:aa:00:0f:e0 | ether() |

Tabla caché ARP.

Cada fila corresponde a un mapeo IP-MAC y la interfaz por donde se ha solicitado.

Esta tabla se va llenando automáticamente a medida que se realizan nuevas peticiones, pero también se puede manipular con el comando arp. Con ella se pueden consultar, añadir o borrar entradas.

La tabla caché ARP se llama *caché* porque, de hecho, actúa como una memoria auxiliar que evita la consulta de la información a la LAN mientras se tenga una copia local. Puede parecer que gracias a esta tabla, las consultas ARP tendrían que ser cada vez menos, a medida que se van recogiendo direcciones MAC de todas las estaciones de la LAN. Pero eso no es así. Las entradas de la tabla caducan al cabo de un periodo de tiempo relativamente breve (entre uno y unos cuantos minutos, según el sistema).

*Usos alternativos del ARP*

El ARP, además del uso principal que acabamos de describir, tiene algunos usos alternativos interesantes, como por ejemplo:

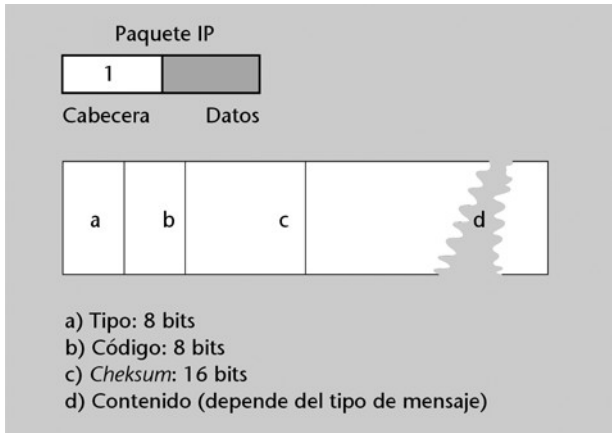
- ARP innecesario (*gratuitous-ARP*): se utiliza cuando una estación arranca para saber si hay alguna otra estación que está utilizando su misma dirección IP. Mediante una petición ARP, puede preguntar quién tiene su IP (un conflicto de este tipo podría dejar las dos estaciones fuera de combate).
- ARP subsidiario (*proxy-ARP*): se utiliza en situaciones en las que un encaminador divide una subred sin que las estaciones ni el encaminador que los conecta a Internet modifiquen su máscara. Esta técnica, a pesar de no ser muy ortodoxa, se aplica frecuentemente cuando el encaminador de una red privada se conecta a Internet mediante una red ajena (un proveedor de Internet, por ejemplo).

## 4. El ICMP (*Internet control message protocol*)

La cuestión de si el ICMP es un protocolo, o si más bien es una herramienta que utiliza el protocolo IP para notificar errores genera mucha polémica. Lo cierto es que el ICMP constituye el mecanismo básico para la gestión de las diferentes incidencias que pueden ocurrir en una red IP.

### 4.1. Mensajes ICMP

Los mensajes ICMP viajan dentro de paquetes IP (al contrario de lo que sucedía con los paquetes ARP), en el campo de datos con el campo Protocolo igual a 1. El formato del mensaje presentado en la figura siguiente nos facilitará el estudio de los diferentes usos del paquete ICMP:




Existen trece *tipos* de mensajes ICMP en uso en la actualidad, y alrededor de una treintena de subtipos identificados con el campo Código.

| Tipo | Código | Descripción | Clase |
|---|---|---|---|
| 0 | 0 | Respuesta de eco (*echo reply*) | Petición |
| 8 | 0 | Petición de eco (*echo request*) | Peticiión |
| 3 | 0-15 | Destino inalcanzable (*unreachable destination*) | Error |
| 4 | 0 | Los diferentes códigos permiten definir si lo que no se puede alcanzar es la subred (0, 6, 9, 11), la estación (1, 7, 10, 12), el protocolo (2), o el puerto (3) y los motivos. | Error |
| 5 | 0-3 | Petición de control de flujo (*source quench*) | Error |
| 9 | 0 | Redireccionamiento | Petición |
| 10 | 0 | Publicación de rutas | Petición |
| 11 | 0-1 | El tiempo de vida ha expirado (*time exceeded*) | Error |
| 12 | 0-1 | Cabecera IP incorrecta | Error |
| 13 | 0 | Petición de hora | Petición |
| 14 | 0 | Respuesta de hora (en milisegundos desde la medianoche) | Petición |
| 17 | 0 | Petición de la máscara de la subred | Petición |
| 18 | 0 | Respuesta de la máscara de la subred | Petición |

La última columna nos permite distinguir mensajes ICMP de notificación de error de mensajes que son parte de una petición (la petición o la respuesta). Esta distinción es importante, puesto que los mensajes de error ICMP no pueden generar otros mensajes de error ICMP. En particular, no se generan mensajes de error en respuesta a los paquetes o mensajes siguientes:

- Los mensajes de error ICMP.
- Un paquete IP destinado a una dirección *broadcast* (sea un *broadcast* IP o un *broadcast* MAC).
- Un fragmento que no sea el primero.
- Una dirección de origen que no identifique una única estación. Por ejemplo, la dirección de origen válida 0.0.0.0 o la dirección de origen no válida 255.255.255.255.

Fijaos en que en cualquiera de estas situaciones podrían provocarse respuestas en cascada que podrían afectar gravemente a la red.

**Parches**

Desgraciadamente, no todas las implementaciones TCP/IP han tenido en cuenta las excepciones existentes en la generación de mensajes de error, y algunos usuarios desaprensivos han explotado estos problemas para bloquear sistemas y redes remotas. Asimismo, de vez en cuando se descubren nuevas excepciones que pueden afectar a nuestros sistemas.

Los fabricantes de sistemas operativos publican regularmente parches (*patch*) que permiten solucionar los problemas (las vulnerabilidades) que se han descubierto desde la fecha de publicación de la última versión.

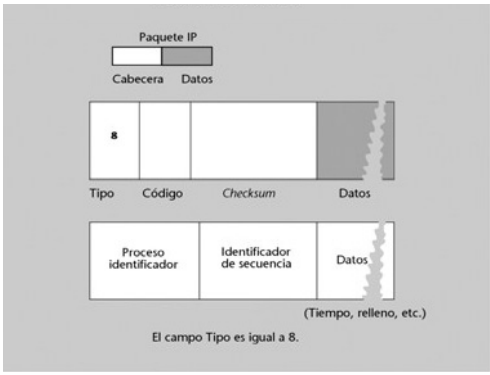
## 4.2. El programa ping

El programa ping permite descubrir si una estación se encuentra activa o no, simplemente efectuando lo siguiente:

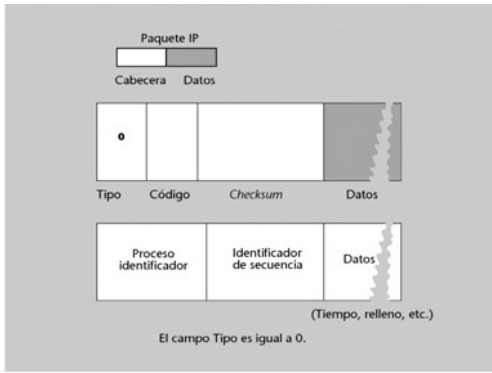
```
$ ping <direccion_IP_destino>
```

La instrucción ping envía un mensaje ICMP del tipo 8 (petición de eco) con el destino indicado. El receptor de la petición debe responder con una respuesta

de eco (ICMP tipo 0), y, cuando el ping la recibe, indica en pantalla que la estación remota está activa.




Mensaje ICMP de petición de eco.



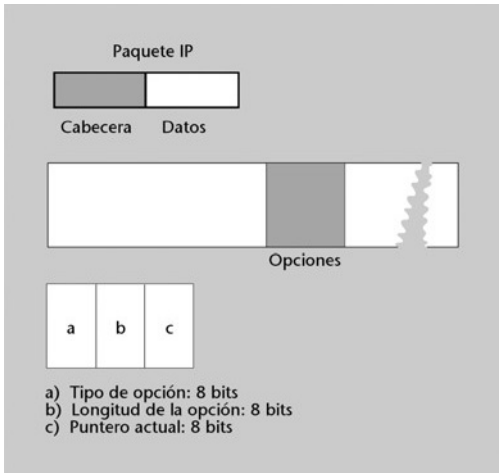

Mensaje ICMP de respuesta de eco.

En sistemas multitarea puede haber más de una petición en curso. Para saber a cuál de las instrucciones ping debe entregarse la respuesta, cada ping asigna el identificador del proceso ping a los mensajes generados. El campo Identificador de secuencia permite al ping enviar diferentes paquetes e identificar sus respuestas (ping –t en MSWindows y ping en UNIX[3]). El campo de datos permite añadir una cadena arbitraria de bytes a la petición de eco (ping –s en UNIX y ping –t en MSWindows). La respuesta debe llevar una copia de dichos bytes.

Con el ping tenemos otras opciones disponibles. En particular, la opción de memorización de rutas (*record route* o RR)[4]. Esta opción no se refleja en ninguno de los campos del mensaje ICMP, sino que se encuentra dentro de la misma cabecera IP, en el campo de opciones.

Las diferentes opciones que se encuentran disponibles en la cabecera se identifican por medio del primer byte del campo de opciones de la cabecera IP:

3. En diferentes versiones de UNIX los parámetros pueden cambiar. En caso de duda, hacemos **man ping**.
4. La sintaxis de la opción RR en UNIX es **ping –R** y en MSWindows, **ping –r**.




Si el originador quiere activar la opción RR, el primer byte debe ser 7. En todas las opciones el primer byte indica el tipo, y el segundo, la longitud en bytes de la opción. En este caso siempre se pide el máximo posible, que es 39 bytes, puesto que el campo siguiente tiene un byte (puntero), al que sigue una cadena de campos de 4 bytes (que son las direcciones IP encontradas). El campo Puntero se inicializa a 4, y dentro de los 4 bytes siguientes se guarda la dirección de la estación en que ejecutamos el ping.

Cada encaminador debe mirar dentro de los paquetes IP (ICMP o no) para ver si tienen la opción RR activada. Si un encaminador encuentra un paquete con esta opción activada, modifica la posición apuntada por el puntero con su dirección (por norma general, la dirección de salida del encaminador) e incrementa el valor del puntero en cuatro unidades. Cuando vuelve el mensaje de respuesta de eco, se ha añadido una lista con todos los saltos que ha tenido que realizar el paquete (tanto de ida, como de vuelta).

El campo de opciones tiene limitaciones de tamaño: sólo dispone de 36 bytes (39 3) para guardar direcciones IP. Como cada dirección ocupa 4 bytes, sólo hay espacio para nueve direcciones IP. Si a ello le añadimos que no todos los encaminadores comprueban si hay opciones dentro de los paquetes IP, o no actualizan la opción RR, hace poco útil este tipo de ping en el mundo real.

### 4.3. El programa traceroute

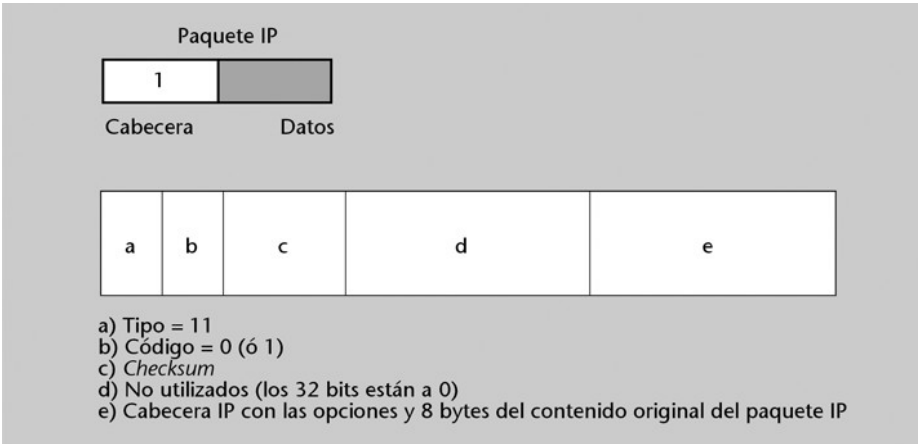
El programa traceroute permite encontrar las rutas entre un origen y un destino sin ninguna de las limitaciones del ping -R (ping -r en MSWindows). Utiliza un mecanismo bastante ingenioso basado en mensajes genéricos ICMP (y no los específicos petición de eco y respuesta de eco del ping).

El funcionamiento se basa en la explotación de dos mensajes ICMP:

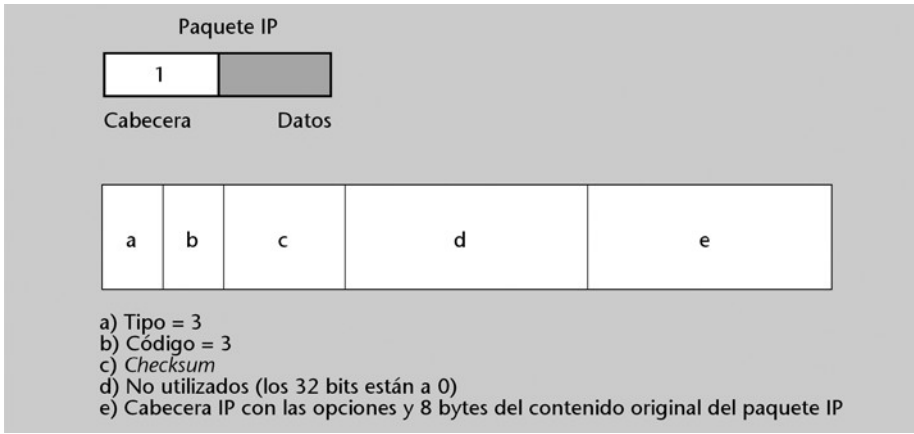
1) Tiempo de vida agotado (*time-exceeded*): cuando un encaminador recibe un paquete, aparte de las tareas primordiales de encaminamiento, debe reducir en una unidad el valor del campo TTL de la cabecera IP.

   En caso de que el valor (después de la reducción) sea cero, el paquete debe eliminarse. Sin embargo, esta eliminación no es silenciosa, sino que el encaminador responsable envía una notificación de la misma al originador del paquete por medio de un mensaje ICMP tipo 11 y código 0 (tiempo de vida agotado).

   Este paquete ICMP contiene la cabecera del paquete IP que se ha eliminado y los primeros 8 bytes de su contenido (seguramente la cabecera UDP o los primeros bytes de la cabecera TCP).




2) Puerto inalcanzable (*unreachable-port*): cuando una estación recibe un datagrama UDP o un segmento TCP destinado a un puerto que la máquina no escucha, responde con un mensaje de error de puerto inalcanzable (tipo 3 con código 3).




El programa traceroute simplemente debe enviar paquetes al destino con TTL secuencialmente ascendentes: el paquete (con independencia del tipo que sea) que tenga el TTL = 1 será rechazado por el primer encaminador, el que tenga TTL = 2 lo será por el segundo, y así consecutivamente. Cada uno de los encaminadores devolverá un mensaje ICMP "tiempo de vida agotado", una pista del todo suficiente para que el originador averigüe el camino que han seguido todos los paquetes.

Las implementaciones de UNIX probadas utilizan un datagrama UDP destinado a un puerto elevado (para minimizar la probabilidad de que exista un servicio usando el puerto). El tracert de Windows utiliza precisamente mensajes ICMP *echo-request*.

Cuando el mensaje llega al destino, debe devolver algún mensaje para saber que la secuencia ha finalizado. Por norma general, el mensaje será "puerto inalcanzable" si el mensaje enviado era un datagrama UDP a un puerto no usado, o bien una respuesta de eco si lo que se ha enviado son paquetes ICMP de petición de eco.

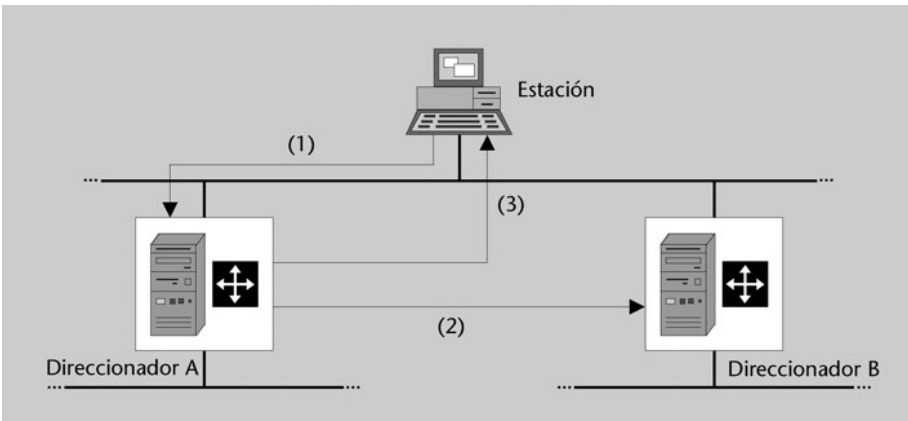
## 4.4. Mensaje de reencaminamiento

Es normal que los sistemas se conecten a Internet sólo configurando su dirección IP, la máscara de la LAN y el encaminador que gestiona la comunicación remota. Cuando hay más de un encaminador en la LAN local, el que hemos configu-

rado en la estación detecta que recibe paquetes que, en realidad, deberían ir a otro de los encaminadores de la misma LAN.

A este efecto, los encaminadores disponen del mensaje ICMP de redireccionamiento (*redirect*), que actúa de la manera siguiente:

1) La estación envía un paquete al encaminador que tiene configurado (A). El encaminador A descubre que la mejor ruta pasa por utilizar el encaminador B.
2) El encaminador A direcciona el paquete hacia el encaminador B.
3) Notifica a la estación que modifique su tabla de encaminamiento.




Redireccionamiento de paquetes.

Notad que, normalmente, el encaminador continúa direccionando los paquetes (paso 2). Aunque la estación no hiciera caso del mensaje ICMP de redireccionamiento (paso 3), continuaría teniendo conectividad con el exterior; sin embargo, si le hace caso, mejorará el rendimiento del sistema.

Cuando una estación obedece a un ICMP de redireccionamiento, su tabla de encaminamiento queda convenientemente actualizada. El comando route puede proporcionarnos alguna pista de si este fenómeno tiene lugar.

Como reseña final, hay que tener en cuenta que algunas implementaciones actuales ignoran este tipo de mensajes ICMP. Esto se debe al hecho de que este mecanismo abre la puerta a ser explotado de manera trivial por usuarios maliciosos que deseen suplantar el encaminador, de manera que engañen al resto de estaciones de la LAN y redirijan todo el tráfico saliente hacia él (normalmente,

con no muy buenas intenciones: poder capturar o analizar el tráfico, denegación de servicio, etc.). En cualquier caso, es importante ser consciente de esta vulnerabilidad.

## 5. Protocolos de encaminamiento

Como hemos visto anteriormente, hay que llenar las tablas de encaminamiento que los encaminadores utilizan para hacer circular los paquetes por la red. Eso se puede hacer de forma manual, pero también existen protocolos que permiten a los encaminadores intercambiar información sobre la red y cambiar las propias tablas para que en cada momento los caminos que se utilicen sean los mejores.

La estructura de la red Internet ha evolucionado mucho a lo largo del tiempo, desde el original NSFNet, hasta la red ubicua que es hoy en día. También lo han hecho los protocolos de encaminamiento. Al principio se podía plantear un sistema de control centralizado de la red. Hoy en día eso es claramente imposible.

En estos momentos se utiliza el concepto de sistema autónomo (*autonomus system*, AS) para separar claramente lo que pasa dentro del AS de lo que pasa fuera.

Un AS es una colección de redes IP y direcccionadores bajo el control de una organización, que utiliza una política de encaminamiento común.

Cada AS tiene un número (ASN), y es asignado por la IANA. Actualmente son números de 16 bits, pero próximamente pasarán a 32.

Dentro de un AS se utiliza un IGP (*interior gateway protocol*) para que los encaminadores se intercambien la información, mientras que entre AS se utiliza un EGP (*exterior gateway protocol*).

Los entornos donde actúan los IGP y los EGP son muy diferentes, y por eso su funcionamiento difiere bastante. Un IGP actúa dentro de un AS, y el nivel de información y detalle del AS que se tiene es mucho máselevado que el que pueda tener un EGP respecto del resto de AS.

Con respecto a EGP, el más usado (y con diferencia) es el BGP (*border gateway protocol*). En cambio existen diversos protocolos que se utilizan como IGP, incluidos en dos categorías diferentes:

1) Protocolos de encaminamiento basados en vectores de distancias. En estos protocolos los encaminadores no tienen conocimiento de la topología de la red entera. Van manteniendo la información de sus tablas intercambiando información con sus vecinos. Utilizan el algoritmo de Bellman-Ford para calcular los costes de los caminos. Los protocolos de este tipo más comunes son el RIP (*routing information protocol*) y el IGRP (*interior gateway routing protocol*).
2) Protocolos de encaminamiento basados en estado de enlaces. En estos protocolos los encaminadores sí que tienen información de toda la red, y a partir de esta información pueden calcular cuál es el mejor salto siguiente para cada paquete que tienen que direccionar. Los protocolos de este tipo más comunes son el OSPF (*open short path first*) y el IS-IS (*intermediate system to intermediate system*).

A continuacióndescribimos de forma sintética las principales características de los cuatro protocolos interiores.

### 5.1. RIP

Es uno de los primeros protocolos que se desarrollaron. De hecho, ya estaba incluido en el UNIX de la Universidad de Berkeley. Actualmente hay tres versiones de este protocolo: la primera, que prácticamente ya no se utiliza porque no soporta ni subredes ni CIDR, la segunda, que es el que se utiliza en IP versión 4, y la tercera, que se ha hecho para soportar IPv6. RIP es abierto, es simple y es fácil de configurar.

RIP utiliza mensajes de petición, con los cuales los encaminadores piden información a sus vecinos, y mensajes de respuesta, que contienen esta información. Los mensajes viajan en paquetes UDP, puerto 920.

Los encaminadores calculan la métrica de un destino (o sea, la distancia hasta él) como la suma de la métrica comunicada por un vecino más la distancia hasta este vecino. En caso de que haya más de una métrica para el mismo destino, se escoge la menor.

Las limitaciones de RIP vienen por dos características que tiene: por un lado sólo tiene en cuenta el número de saltos para decidir la mejor ruta, y no tiene en

cuenta otras características de la red como la capacidad o la velocidad de los enlaces. Eso puede provocar ineficiencias porque puede preferir una ruta con pocos saltos pero muy lentos por delante de una ruta con algún salto más pero mucho más rápidos. Por otro lado, no es capaz de detectar rutas circulares. Por eso se limitan las rutas a 15 saltos.

### 5.2. IGRP

IGRP es propietario. Fue desarrollado por la empresa Cisco para superar las limitaciones de RIP. Tiene en cuenta múltiples factores para calcular las distancias (anchura de lado, carga real de los enlaces, retraso, coste económico…) y permite caminos de hasta 255 saltos.

### 5.3. OSPF

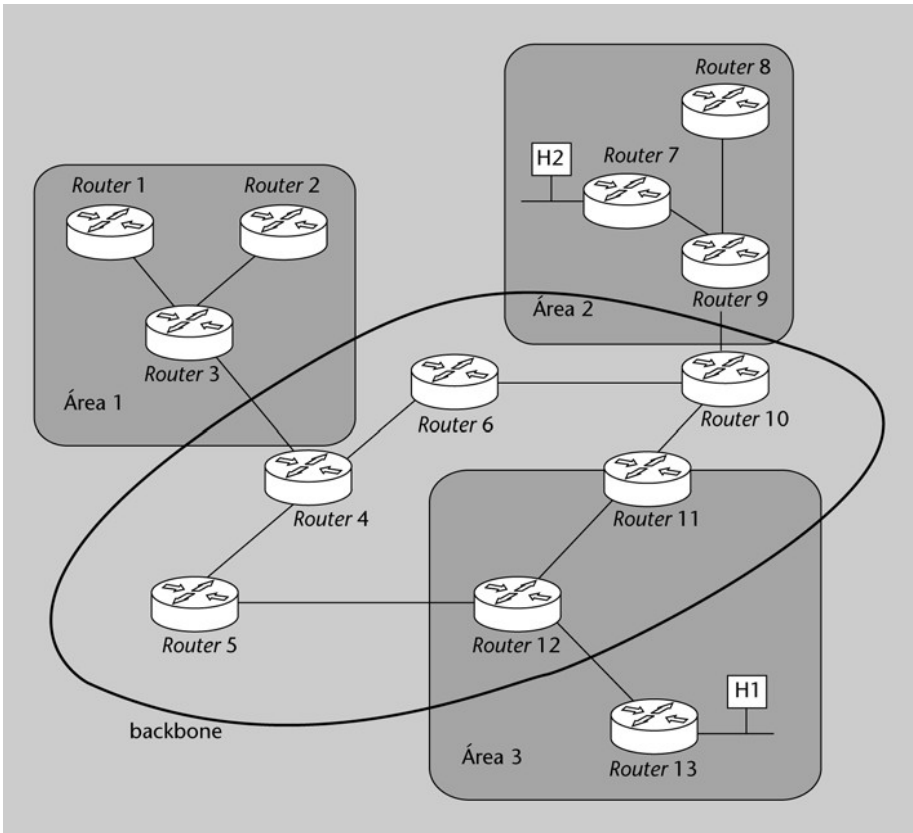
Es el protocolo de red interior más usado en redes grandes. Utiliza el algoritmo de Dijkstra para calcular las rutas más cortas posibles, y se encarga de que todos los encaminadores de la red conozcan la topología del AS completo. Esta información se mantiene como un grafo dirigido. El coste de los enlaces se calcula a partir de una función configurable por el usuario, que puede tener en cuenta todos los factores relacionados con la red: retraso, velocidad, coste económico, etc.

Una red OSPF se puede descomponer en redes más pequeñas o áreas, que son grupos lógicos de encaminadores. Hay un área especial, llamada *backbone*, que es el núcleo de una red OSPF. Todas las otras áreas están conectadas a un encaminador del *backbone*. Así, las rutas entre diferentes áreas atraviesan siempre el *backbone*.

Una entidad OSPF puede estar en siete estados diferentes, y va cambiando de estado a medida que avanza el diálogo que establececon otros vecinos.

Los estados on:

- Desactivado
- Inicialización
- Dos-vías




- Pre-intercambio
- Intercambio
- Carga
- Adyacencia

El diálogo entre vecinos se realiza con paquetes IP, del tipo 89.

## 5.4. IS-IS

De hecho, IS-IS nació como un protocolo de encaminamiento para el nivel red de la torre OSI, y es coetáneo de OSPF. La adaptación que se hizo para Internet se llama *IS-IS integrado*. IS-IS y OSPF son conceptualmente bastante parecidos.

Los dos se basan en el estado de los enlaces y utilizan el algoritmo de Dijkstra para calcular las métricas. La diferencia más notable radica en la manera como se definen las áreas. EN IS-IS los encaminadores pueden ser de nivel 1(intraárea), de nivel 2 (interárea) o de nivel1-2 (ambos), y no existe el concepto de *backbone* por donde tienen que pasar todas las rutas.

## 6. Mecanismos de acceso a Internet

Los mecanismos de acceso a Internet más habituales son la red telefónica (por módem) que se utiliza, sobre todo, en el ámbito doméstico, el ADSL que, aunque utiliza la infraestructura de acceso de la red telefónica, no se puede decir que vaya sobre la línea telefónica, y la Ethernet.

1) En accesos por medio de la red telefónica y, en general, en accesos por medio de redes conmutadas (incluyendo el acceso por la RDSI), se suele utilizar el protocolo PPP.

   Si bien durante mucho tiempo se han utilizado el SLIP y el CSLIP, en la actualidad se han dejado prácticamente de lado en favor del PPP, que tiene más flexibilidad (permite gestionar automáticamente ciertos parámetros IP y multiplexar, dentro de la misma conexión, diferentes protocolos de interconexión, aparte del IP) y es más fiable (dispone de CRC en cada trama).

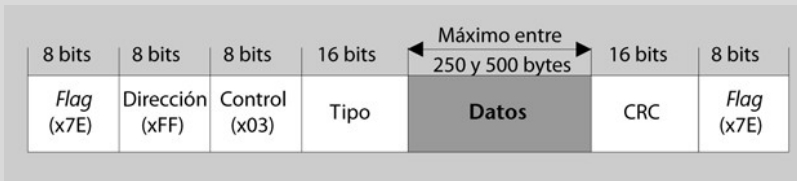
2) En LAN, el protocolo que se utiliza en más del 90% de los casos es la Ethernet. Nosotros nos centraremos sólo en los detalles de las direcciones de dicho protocolo, puesto que es lo único que afecta a la manera de funcionar del IP.

Casi todos los protocolos de LAN que componen el 10% restante (IEEE802.3 CSMA/CD, IEEE802.5 Token Ring, etc.) utilizan una estructura de direcciones idéntica a la de Ethernet. De hecho, podríamos hablar de compatibilidad, puesto que la asignación de direcciones se hace globalmente para todas las LAN mencionadas y la gestiona el IEEE.

## 6.1. Acceso telefónico: el PPP

Un enlace PPP es un enlace punto a punto entre dos ordenadores; el cliente que es quien se quiere conectar a Internet y un servidor PPP que forma parte de una LAN con conexión a Internet. Este servidor se encarga de hacer llegar al ordenador cliente todos los paquetes que sean para él, y enviar hacia la marcha todos los que salgan del cliente.

El PPP es fundamentalmente un protocolo derivado del HDLC para conexiones balanceadas (HDLC-ABM). El formato de la trama PPP se representa en la figura siguiente:




Los campos Indicador (*flag*), Dirección y Control están fijados en los valores de la figura anterior. El campo Dirección tiene el valor 11111111, que es el de difusión o *broadcast* en la mayoría de los protocolos (por ejemplo, en todos los HDLC). Ello significa que este campo (como el de control) no se utiliza. Su utilización en el PPP sólo se puede justificar por el posible uso de tarjetas HDLC genéricas para conexiones PPP. Como cualquier protocolo HDLC, debe aplicar el mecanismo de transparencia de inserción de ceros (*bit stuffing*).

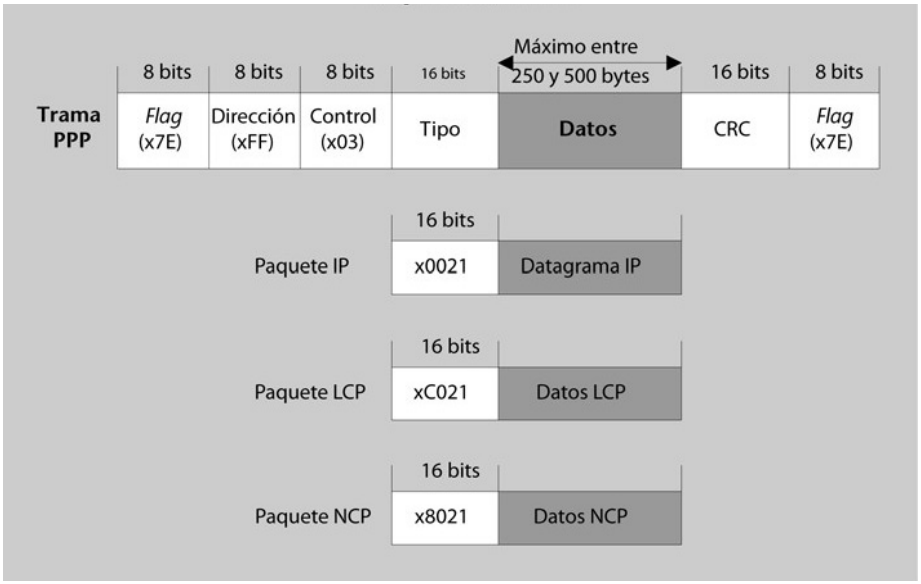
Como el campo de control está fijado (siempre vale 3), el PPP no distingue los diferentes tipos de tramas. Todas son de información, no hay confirmaciones ni tramas de establecimiento de enlace (es un protocolo no orientado a conexión).

El PPP especifica una variante orientada a carácter (los protocolos de la familia HDLC están orientados a bit), que es la que más se utiliza en enlaces por medio de módem (un contraejemplo serían las conexiones mediante la RDSI).

Lo importante es lo que transportan las tramas PPP. El mismo estándar define la multiplexación de diferentes protocolos, que se distinguirán por medio del campo Tipo. Los que nos interesan son los siguientes:

- LCP (*Link Control Protocol)*: es el protocolo encargado de realizar el test del enlace, el control de la conexión y la gestión del enlace.

- Protocolos de red: son tramas que encapsulan paquetes de nivel superior, como puede ser IP. Pero también pueden transportar NETBEUI, AppleTalk, Decnet, etc.
- NCP (*Network Control Protocol)*: es el protocolo que se utiliza para tareas de gestión de los protocolos de red que transporta el enlace. En el caso del IP, permite que uno de los terminales asigne la dirección de Internet al otro y configure los diferentes parámetros de la red (encaminador, máscara, etc.)




Encapsulamiento PPP.

### *6.1.1. Compresión de las cabeceras*

Como ya hemos visto, la pirámide de PDU provoca ineficiencias en la transmisión, sobre todo en protocolos como Telnet, que con frecuencia envían bloques muy cortos de información. Asimismo, el PPP se ha utilizado en enlaces telefónicos que funcionan a velocidades máximas teóricas de entre 9.600 bps (módems norma V.32) y 33.600 bps (módems norma V.34). Ello hace que dicha ineficiencia sea todavía más grave.

**Velocidades de transmisión de datos con Telnet**

Si se utiliza Telnet, para enviar un carácter (de 1 byte) debemos enviar 8 + 20 + 20 + 1 = 49 bytes. Con un módem de 9.600 bps, si utilizáramos mensajes de 8 bits de longitud más 1 bit de arranque y 1 bit de parada, podríamos transmitir:

$$9.600 / (1 + 8 + 1) = 960 \text{ caracteres/segundo.}$$

Con las limitaciones impuestas por las cabeceras, sólo podremos transmitir:

$$9.600/(1 + 8 + 1) \cdot (1 / 49) = 19{,}5 \text{ caracteres/segundo.}$$

De hecho, nadie es capaz de mecanografiar tan rápido. Sin embargo, no debemos perder de vista que el enlace puede ser compartido por otras conexiones (transferencia de ficheros, consulta de la web, etc.).

La compresión de cabeceras Van Jacobson mejora considerablemente este problema. Este tipo de compresión se basa en el hecho de que en un acceso a Internet con PPP, en general no habrá demasiadas conexiones TCP simultáneas. La compresión Van Jacobson permite mantener una tabla de hasta dieciséis conexiones simultáneas TCP/IP, a las que asignará dieciséis identificadores diferentes. Como la mayoría de los campos de las dos cabeceras TCP/IP no varían durante el transcurso de una misma conexión, basta con tener entre 3 y 5 bytes de cabecera para cada paquete (combinando PPP y TCP/IP) para mantener un funcionamiento correcto de la conexión. La ganancia en eficiencia de la transmisión es suficientemente importante.

Para saber con exactitud qué ganancia se consigue, precisamos saber qué longitud pueden tener las tramas PPP.

Por medio del LCP también pueden obtenerse otras mejoras como, por ejemplo, la eliminación de los campos de control y de dirección.

### *6.1.2. MTU*

Muchos protocolos poseen una longitud máxima de transmisión. Esta última, llamada MTU, está determinada por diferentes factores según el caso. En el PPP (que es lo que ahora nos interesa), encontramos un protocolo derivado del HDLC. Este último no impone ningún límite en la MTU de las tramas; en teoría, podríamos hacerlas tan largas como la probabilidad de error nos permita (la probabilidad de error en uno de los bits de la trama no debería ser demasiado alta).

Sin embargo, existe un factor que limitará la MTU más que los límites impuestos por los protocolos: el confort de la conexión. Se ha demostrado que, en conexiones en tiempo real (como una conexión Telnet), el usuario debe recibir una reacción a sus acciones en una décima de segundo como máximo. Retrasos superiores provocan cansancio y dan la sensación de que "la máquina va lenta", que todos hemos experimentado alguna vez.

Si pensamos que dentro de la misma conexión podemos tener multiplexadas otras de transferencia (como FTP o web), nos encontraremos que los paquetes de las aplicaciones en tiempo real, que suelen ser cortos, deben esperarse detrás de los paquetes de longitud máxima que se utilizan en las aplicaciones que tienen una tasa de transmisión elevada (supondremos que los paquetes de la aplicación en tiempo real tienen preferencia sobre otros que previamente estuvieran en la cola de salida).

La única manera de hacer que una aplicación estándar de transferencia utilice paquetes más pequeños es reducir la MTU de la conexión. En el caso del PPP, si tomamos como referencia un módem de 33.600 bps, tenemos que un paquete de una décima de segundo de duración tendría los bytes siguientes:

$$33.600 \text{ bps} \cdot (0{,}1 \text{ s}) \cdot (1 \text{ byte} / 8 \text{ bits}) = 420 \text{ bytes}.$$

En conexiones PPP tendremos, pues, la MTU entre 250 y 500 bytes. Con módems que dispongan de compresión, los valores que obtendremos pueden aumentar.

**Módems para línea telefónica**

Las velocidades de transmisión de los módems estándar son 9.600 bps (V.32), 14.400 bps (V.32bis) y 33.600 bps (V.34).

En general, los módems disponen de dispositivos electrónicos para comprimir y descomprimir datos. Los estándares V.42bis y MNP9 realizan compresiones de hasta 1:4, con lo que pueden lograr velocidades de transmisión efectiva de hasta 134.400 bps (V.34 + V.42bis) en situaciones favorables.

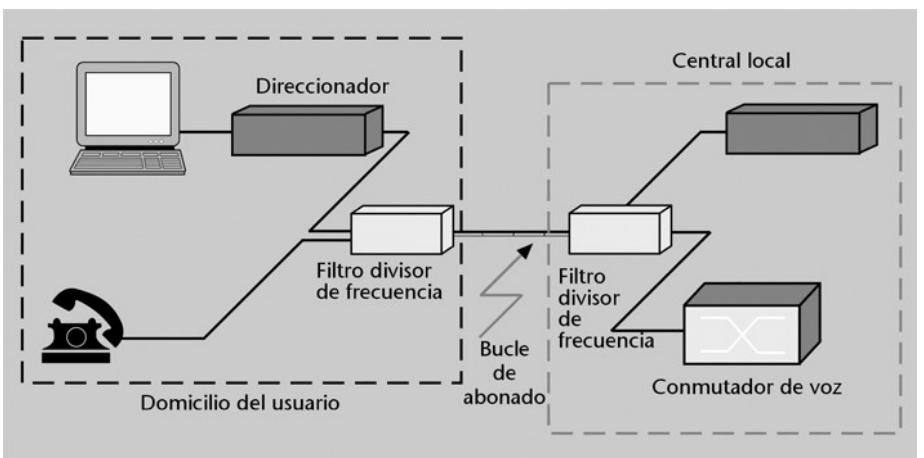
Hay módems (los que siguen el estándar V.90) que pueden lograr velocidades de hasta 56 kbps. En realidad, no son módems en el sentido estricto, puesto que necesitan que en el otro extremo de la línea haya un codec conectado a una línea digital.

## 6.2. Acceso ADSL

Si bien el acceso a Internet por módem por medio de PPP ha sido la manera habitual de conexión por parte de los usuarios domésticos y las pequeñas empresas durante la primera "década Internet", esto ha cambiado y en la segunda "década Internet", se han adoptado sistemas que facilitan la conexión permanente.

La conexión por módem ocupa la línea de teléfono del abonado y, lo que todavía es peor, establece una llamada durante un tiempo indefinido.

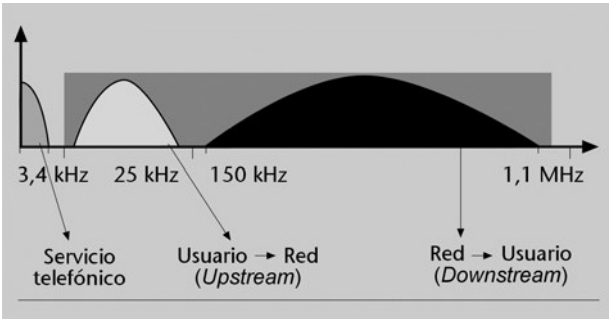
El ADSL representa una solución a este problema, puesto que, por un lado (aunque utiliza el cableado de la línea telefónica, el bucle de abonado), la línea telefónica queda libre para llamadas mientras se está conectado a Internet y, por otro, la conexión no consume recursos de la red telefónica, puesto que, cuando la señal llega a la centralita, se extrae del bucle de abonado y pasa a la red IP de la operadora.




En general, el encaminador que hay en el domicilio del usuario ofrece una conexión Ethernet equivalente a un concentrador, lo que permite que el abonado conecte más de un ordenador por medio de la línea ADSL.

Para permitir que la línea telefónica conviva con la conexión a Internet permanente, se realiza una división del espectro: la parte baja continúa ocupada por el canal de voz (hasta 4 kHz) y, a partir de aquí, se sitúa el espectro de la codifi-

cación ADSL (con la limitación propia del par de hilos). El sistema es bidireccional: reserva el espectro bajo para la salida a la red, y el resto, para la entrada:




Espectro de frecuencia ADSL.

Algunas operadoras, para aprovechar mejor líneas malas, utilizan una variante del estándar que permite la transmisión bidireccional con los dos canales en la misma banda.

Esta solución, aunque necesita DCE más caros, permite aprovechar líneas con peores condiciones y/u obtener velocidades más altas.

Generalmente, no se puede llegar a los 1,1 MHz que indica la figura, puesto que la calidad del par de hilos es muy variable (clima, longitud, edad, etc.), por lo que se utiliza una codificación adaptativa: los dos sentidos del canal se dividen en subbandas de 4 kHz independientes (DMT, *discrete multitone*) que se codifican (en realidad, se modulan) con procedimientos casi idénticos a los utilizados por los módems tradicionales. Con ello, se consiguen treinta y dos canales de salida y hasta doscientos cincuenta y seis de entrada, con una capacidad de 60 kbps cada uno de los mismos –modulados en QAM (*quadrature amplitude modulation*), como los módems telefónicos– que, acumulados, proporcionan un máximo de 1,92 Mbps de salida y 15,36 Mbps de entrada.

No obstante, las mejores conexiones no consiguen aprovechar correctamente los canales superiores y se considera que el límite máximo de entrada es de 8 Mbps. Asimismo, las operadoras no suelen ofrecer conexiones tan rápidas, lo que hace que no se pueda lograr el límite teórico, aunque las velocidades ofrecidas van subiendo.

Los protocolos utilizados dentro del ADSL dependen de la operadora y no están definidos por el estándar. Entre los posibles protocolos para el ADSL, tenemos el ATM y el mismo PPP.

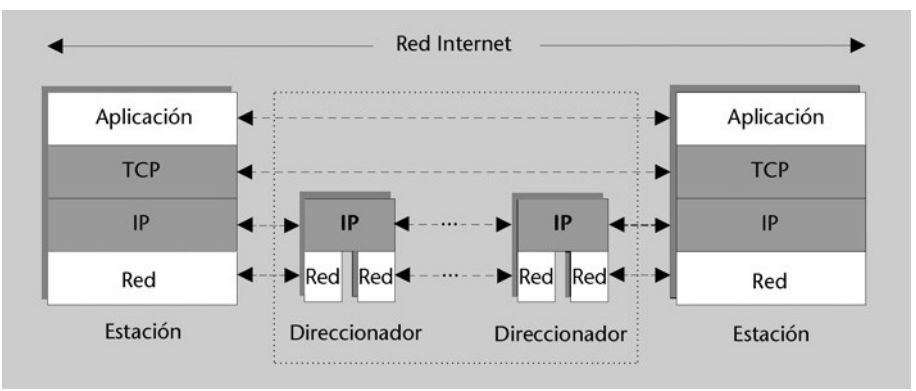
## 6.3. Acceso LAN: el protocolo Ethernet

Seguramente, la simplicidad de este protocolo de red, y no sus prestaciones teóricas, ha hecho que sea el más utilizado en redes de área local prácticamente desde que DEC, Intel y Xerox establecieron un estándar (*de facto*) para LAN con control de acceso al medio CSMA/CD, basándose en una arquitectura similar desarrollada durante los años setenta por Xerox.

Hay un principio que se cumple en todas las redes de área local: lo que una estación transmite es recibido por todas las demás. Una estación sabe cuándo una trama le va destinada porque lee todas las que le llegan y comprueba la dirección de destino. Tiene que rechazar todas las tramas con direcciones que no sean la suya. Sin embargo, también hay excepciones, y en algunos casos las estaciones también deben capturar tramas dirigidas a direcciones especiales (como las *multicast* y las *broadcast*).

**Configuración de la tarjeta de red en modo promiscuo**

Uno de los casos en el que una estación no rechaza las tramas con direcciones diferentes de la suya es cuando la estación configura la tarjeta de red en modo promiscuo. En este modo, la tarjeta inhabilita su capacidad de filtrado y lee todas las tramas que pasan por la red. Equipos que, generalmente, funcionan en este modo son los puentes (*bridge*s, sistemas destinados a la interconexión de LAN), los analizadores de tráfico (o analizadores de red) o los conmutadores (*switches*). No obstante, casi todas las tarjetas se pueden configurar en modo promiscuo, lo que con frecuencia es aprovechado por los ladrones de información (*crackers*) para leer y copiar información interesante que viaje por la LAN (principalmente contraseñas).




Situación de los niveles de la Red Internet.

De hecho, ésta es la situación por la cual se desarrolló Internet: la interconexión de redes entre ellas, de manera que cualquier ordenador de una red se pueda comunicar con cualquier otro que esté en otra red, superando así las limitaciones físicas y lógicas de las LAN.

Tal como se ha visto en este capítulo y queda reflejado en la figura anterior, el papel principal en esta situación lo juegan los encaminadores. Por lo tanto, para poder acceder a Internet desde un ordenador que está dentro de una LAN, hace falta que en esta LAN haya un encaminador conectado a otras redes.

Las situaciones anteriormente descritas (módem y ADSL) han aparecido *a posteriori*, a medida que han proliferado los ordenadores domésticos aislados.

## 6.4. El DHCP (*Dynamic Host Configuration Protocol*)

El DHCP permite que una estación de una LAN obtenga de forma automática una dirección IP, así como otros parámetros de la conexión, evitando tener que configurarla manualmente.

En accesos PPP no hace falta DHCP porque el propio PPP ya incorpora la asignación de los parámetros de conexión.

Quien proporciona estos parámetros es un servidor DHCP. La única restricción para que un cliente pueda obtenerlos para conectarse a Internet es que los dos estén en la misma LAN. Como el cliente todavía no tiene dirección IP ni sabe quién es su encaminador, la única cosa que puede hacer es enviar un *broadcast* a la LAN donde se encuentra, pidiendo los parámetros de conexión. El servidor DHCP le contestará con los parámetros que pide, y a partir de aquí, la estación ya está bien configurada para poder acceder a toda la Internet.

DHCP permite que esta asignación expire al cabo de un tiempo y haya que refrescarla. De esta manera se asegura que no se desperdicien direcciones. Los tiempos típicos van desde una hora hasta meses. En DHCP se habla de préstamo de una dirección IP. Cuando la estación deja de pedir renovaciones de préstamo (porque se ha apagado, o porque su usuario ha decidido desconectarse de Internet), la dirección IP que se le había asignado pasa a estar disponible.

Los parámetros de conexión que se pueden obtener con el DHCP son:
- dirección IP
- máscara de la red
- encaminador
- servidor DNS.

La asignación de direcciones IP puede ser de tres tipos diferentes:
1) Asignación manual o fija: El servidor DHCP tiene una tabla con parejas de direcciones MAC / direcciones IP. Sólo los clientes con direcciones MAC que están en la lista obtienen dirección IP, y siempre obtienen la misma.
2) Asignación automática: El servidor DHCP proporciona una dirección IP de las que tiene no asignadas, de forma permanente.
   DHCP es el sucesor de BOOTP. Este protocolo, ya obsoleto, proporciona préstamos que no expiran y no soporta opciones.
3) Asignación dinámica: El servidor DHCP proporciona una dirección IP de las que tiene no asignadas, pero con un tiempo de expiración, o sea, "en préstamo". El cliente tiene que pedir la renovación del préstamo una vez transcurrido el tiempo de expiración.

### *6.4.1. Operaciones DHCP*

DHCP pasa por cuatro fases:
- petición de préstamo
- ofrecimiento
- selección
- reconocimiento

Y las operaciones relacionadas con estas fases son:
a) Hallazgo (DHCPDISCOVER): una estación envía un broadcast a la red donde está, preguntando por un servidor DHCP.
b) Ofrecimiento (DHCPOFFER): el servidor DHCP ofrece un préstamo (fijo, automático o dinámico).
c) Petición (DHCPREQUEST): la estación acepta la oferta, o sea, hace la petición formal.
d) Reconocimiento (DHCPACK): el servidor confirma la oferta y comunica la duración del préstamo.

También existe un tipo de petición especial, que es la liberación de préstamo, pero raramente se usa. Lo más habitual es dejar que expire. De hecho, el protocolo no la considera obligatoria.

## 7. NAT

El crecimiento exponencial de Internet hace que la carencia de direcciones se convierta en un problema. Un método para solucionar este problema es la traducción de direcciones de red, también conocida como *NAT (Network Address Translation)*. NAT, definida en el RFC 1631, es el proceso de intercambiar una dirección por otra en la cabecera del paquete IP.

NAT tiene como objetivo solucionar el problema de la falta de direccio nes IP públicas disponibles en Internet. Para hacerlo, utiliza encaminamiento privado en la red local y únicamente publica (traduce) al exterior las direcciones que realmente sean necesarias.

Este proceso se aplica en el encaminador que separa la red interna (privada) de la red externa (pública).

Así pues, la tecnología NAT permite que las redes privadas que utilizan direc ciones IP no registradas se puedan conectar a redes públicas como, por ejemplo, Internet. Cuando una estación de la red privada envíe un paquete hacia la red pública, el encaminador de salida traducirá la dirección privada de esta estación a una dirección pública, es decir, una dirección legal en la red externa. Sin esta traducción, la estación de la red privada nunca llegaría a transmitir información hacia la red externa.

Hay un conjunto de direcciones reservadas que se pueden utilizar en cualquier red interna. Las direcciones IP privadas definidas en el RFC 1918 son las siguientes:

- 10.0.0.0 a 10.255.255.255
- 172.16.0.0 a 172.31.255.255
- 192.168.0.0 a 192.168.255.255

Hay diferentes formas de operación de NAT, según el tratamiento que hace sobre las direcciones. Los tipos de traducción de direcciones que estudiaremos a continuación son las siguientes:

- NAT estática
- NAT dinámica
- NAT con sobrecarga del TCP (también conocida como *PAT*)
- LSNAT *(Load Sharing NAT)*: NAT con balanceo de carga

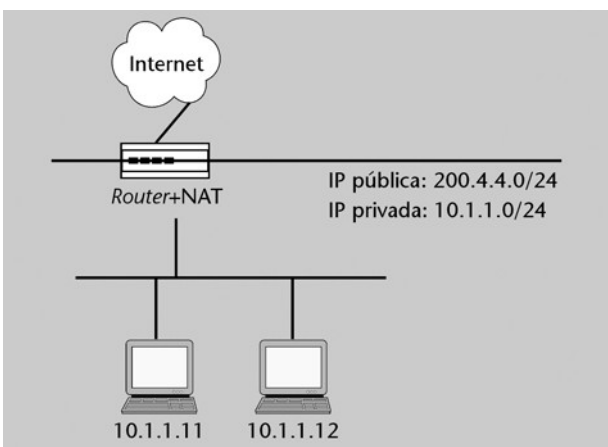
## 7.1. NAT estática

La traducción estática de direcciones es el caso más sencillo de NAT. En este caso, se aplica una traducción de direcciones una a una y, consiguientemente, habrá que definir cada una de las traducciones que se quieran hacer. Esto implicará una gran dificultad de gestión si el número de traducciones es elevado.

Cuando se utiliza NAT estática, cada dirección privada tiene su dirección pública equivalente. Por ello, será necesario que el administrador haga esta relación de equivalencia, una a una.

**Ejemplo de implementación de NAT estática**

La imagen siguiente presenta un escenario con una red interna, la red externa o Internet, y el encaminador que interconecta estas dos redes.



Para configurar la NAT en el encaminador de conexión a Internet habrá que definir la tabla siguiente:

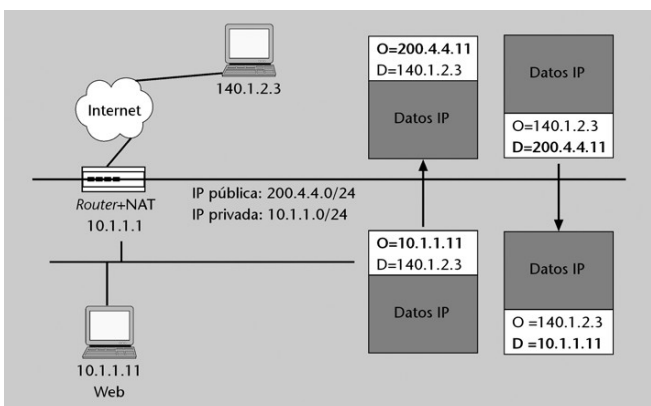
| Tipo de NAT | Dirección privada | Dirección pública |
|---|---|---|
| Estática | 10.1.1.11 | 200.4.4.11 |
| Estática | 10.1.1.12 | 200.4.4.12 |

Como podemos observar, es necesaria una dirección pública para cada estación que tenga una dirección privada. Evidentemente, este tipo de NAT no comporta ningún ahorro de direcciones IP, pero es necesario para todos los servidores que sean accesibles desde Internet. Para conseguir un ahorro real respecto de las direcciones IP públicas habrá que utilizar NAT dinámica.

La relación estática entre la dirección pública y la privada estará en la tabla de NAT hasta que el administrador decida borrarla origen pública correspondiente (200.4.4.11 en este caso). La respuesta de la máquina de destino irá encaminada hacia 200.4.4.11 (dirección IP destino), que se convertirá en 10.1.1.11 al atravesar el encaminador.

El proceso de operación de NAT sobre los paquetes IP es el siguiente: cuando una máquina de la red interna envía un paquete a la red externa, el valor de la dirección IP origen será la dirección privada de la estación origen (por ejemplo, 10.1.1.11). Una vez el paquete llegue al encaminador, éste aplicará la traducción de direcciones correspondiente, y el paquete IP viajará ahora con la dirección IP

El escenario siguiente muestra el modo de operación de NAT y la traducción de las diferentes direcciones:



Es importante destacar que se aplica el mismo proceso cuando el origen del tráfico es externo, ya que cualquier máquina que disponga de una traducción NAT estática será accessible desde el exterior. De esta forma, si una estación externa se quiere conectar con una estación de la red interna (por ejemplo, 10.1.1.11), la dirección destino que utilizará será la dirección pública (200.4.4.11), correspondiente a la dirección interna de la estación. Cuando el encaminador reciba un paquete destinado a esta dirección pública, la traducirá por la dirección privada que corresponde y enviará el paquete hacia la red interna.

### 7.2. NAT dinámica

La traducción dinámica de direcciones tiene lugar cuando el encaminador de salida de la red conoce el conjunto de direcciones internas que tiene que traducir y el rango de direcciones que puede utilizar como direcciones públicas. Y es este dispositivo el que establece la relación entre estos dos conjuntos de direcciones de manera dinámica.

La NAT dinámica asigna dinámicamente la relación entre dirección privada y dirección pública, y se basa en el hecho de que no todas las máquinas de nuestra organización estarán conectadas a Internet al mismo tiempo y, por lo tanto, no hay que disponer de tantas direcciones públicas como direcciones privadas haya.

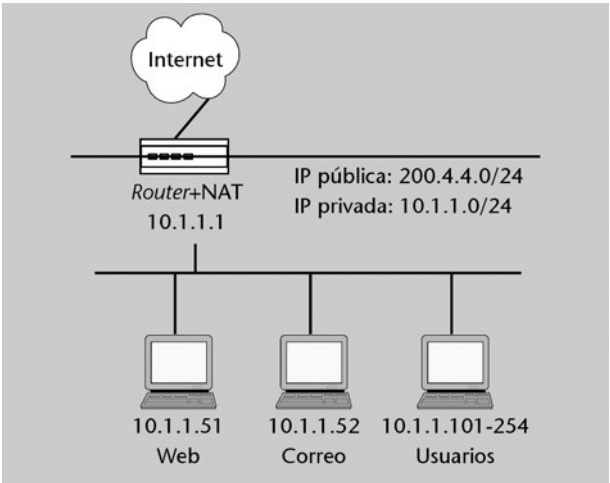
Para definir la NAT dinámica habrá que fijar los rangos de direcciones utilizadas. Esto significa que el administrador de la red tendrá que haber definido previamente la política de asignación de direcciones IP para esta red.

Imaginemos, por ejemplo, la red del apartado anterior en la cual tenemos ahora un número elevado de usuarios que deben conectarse a Internet. Para todos ellos utilizaremos NAT dinámica. Además, también tenemos dos servidores que hay que publicar en Internet, para los cuales utilizaremos NAT estática.

Inferiremos que la política de asignación de direcciones IP es la siguiente:

- Encaminadores: valores de *host* de 1 a 20
- Servidores: valores de *host* de 50 a 100
- Usuarios: valores de *host* de 101 a 254
- El resto de los valores los reservamos para aplicaciones futuras.

Las direcciones públicas y las direcciones privadas son las mismas que en el ejemplo del apartado anterior. Por lo tanto, el escenario de este ejemplo lo podemos representar con el esquema siguiente:




La tabla de traducciones de NAT del encaminador de salida de la red podría ser la siguiente:

| Tipo de NAT | Dirección privada | Dirección pública |
|---|---|---|
| Estática | 10.1.1.51 | 200.4.4.1 |
| Estática | 10.1.1.52 | 200.4.4.2 |
| Dinámica | Rango: 10.1.1.101 a 10.1.1.254 | Rango: 200.4.4.10 a 200.4.4.110 |

En este caso, el rango de direcciones privadas es superior en número al rango de direcciones públicas. Este caso se da cuando tenemos que conectar a Internet un número de máquinas superior al de direcciones públicas disponibles. No habrá ningún problema mientras no haya más máquinas conectadas simultáneamente a Internet que direcciones al rango de direcciones públicas.

Cada vez que una máquina envíe un paquete IP hacia Internet, a su dirección privada se le asignará una dirección pública que se guardará en una tabla de traducciones. Cada traducción tendrá un tiempo de caducidad, después del cual se borrará de la tabla de manera que pueda ser utilizada por nuevas direcciones privadas.

***Ejemplo***

Inferimos que las máquinas 10.1.1.113, 10.1.1.197 y 10.1.1.225 acceden simultáneamente a Internet en un momento determinado. La tabla de traduc ciones del encaminador que aplica la NAT dinámica podría ser la siguiente:

| Dirección privada | Dirección pública |
|---|---|
| 10.1.1.113 | 200.4.4.10 |
| 10.1.1.197 | 200.4.4.11 |
| 10.1.1.225 | 200.4.4.12 |

Cualquier tráfico originado en el exterior de la red privada y destinado a alguna de las direcciones públicas utilizadas en la NAT dinámica, aunque en aquel momento aparezca en la tabla de traducciones dinámicas, será automáticamente descartado en el encaminador. Esto es así para que la correspondencia (IP privada-IP pública) cada vez pueda ser diferente.

En este caso ya conseguimos una primera medida de seguridad: escondemos parte de nuestra infraestructura de red al exterior y únicamente publicamos con NAT estática los servicios que realmente hay que publicar al exterior.

### 7.3. PAT (Port Address Translation)

PAT es un tipo de NAT dinámica que permite reducir al máximo (hasta 1) el número de direcciones IP públicas utilizadas dinámicamente. Para conseguirlo, PAT utiliza el puerto TCP/UDP con el objetivo de identificar el origen de la comunicación (usuario de nuestra red interna).

Así pues, su configuración será similar a la de la NAT dinámica, pero añadiendo el rango de los puertos que se tienen que utilizar. Sin embargo, hay que tener

5. La IANA (Internet Assigned Number Authority) es la organización que, inicialmente, se encargó de distribuir y asignar el encaminamiento público en Internet. Actualmente depende de la organización denomina *ICANN* (Internet Corporation for Assigned Names and Numbers).

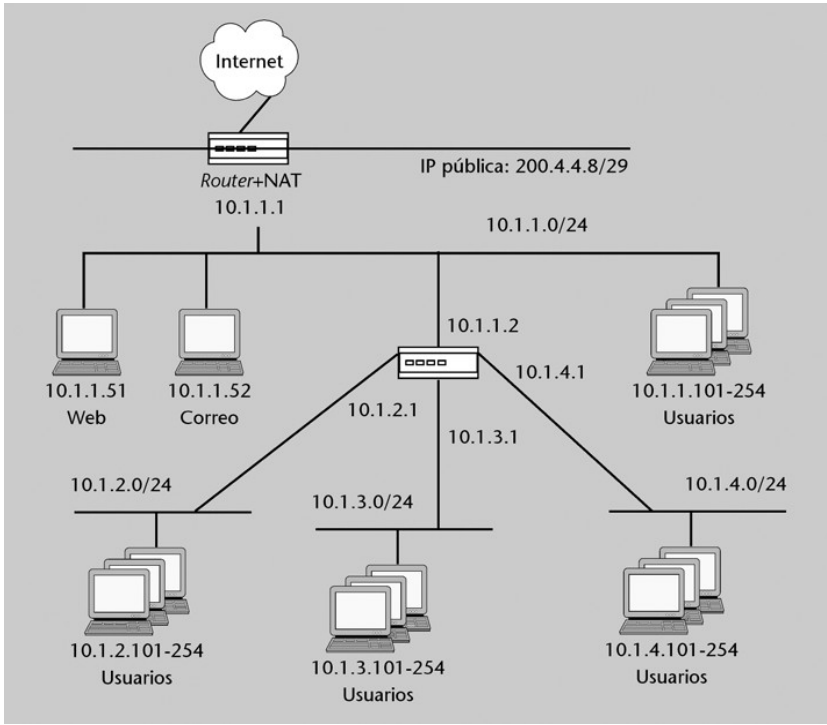
en cuenta que no se puede utilizar cualquier puerto, sino los puertos asignados por la IANA[5] a los clientes.

Recordemos cuáles son estos puertos.

- Del 1 al 1023 (*well-known-ports*): puertos definidos por la IANA para los estándares TCP/IP. Ejemplo: FTP (21), SMTP (25), Telnet (23), web (80).
- Del 1024 al 5000: puertos utilizados por las aplicaciones cliente. Por otra parte, serán los puertos que PAT podrá utilizar.
- Mayores de 5000: puertos para aplicaciones no estándares.

PAT permite traducir varias direcciones privadas de una red interna a una única dirección pública. De esta forma, para poder distinguir las diferentes comunicaciones utiliza, además, los identificadores de puerto TCP/UDP.

A continuación, complicaremos el ejemplo para ver que la PAT realmente permite reducir al máximo el número de direcciones públicas necesarias. Pensemos nuevamente en la red anterior, con los dos servidores de web y correo, que continuarán utilizando NAT estática. En esta red, el número de usuarios que requie-



ren acceso a Internet ha aumentado hasta cuatrocientos, separados en cuatro redes diferentes (unos cien usuarios por red). En la ilustración anterior se representa el nuevo escenario.

Además, nuestro proveedor de servicios en Internet ha hecho cambios en el encaminamiento IP y ahora únicamente nos puede ofrecer la dirección IP 200.4.4.8/29 para todo nuestro sistema. Observemos que, aunque parece que la dirección IP es la misma, la máscara asociada ahora es de 29 bits (255.255.255.248), y esto limita el número de direcciones públicas disponibles a seis:

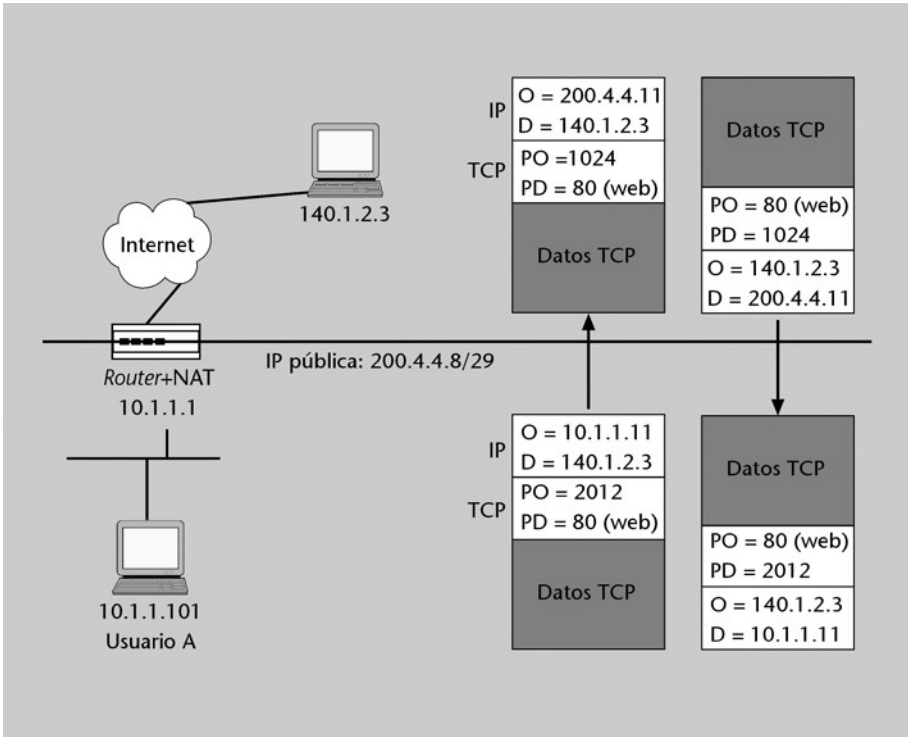
| Direcciones IP disponibles |
|---|
| 200.4.4.9 |
| 200.4.4.10 |
| 200.4.4.11 |
| 200.4.4.12 |
| 200.4.4.13 |
| 200.4.4.14 |

Ante esta limitación de direcciones públicas, decidimos utilizar PAT para que los usuarios de la red tengan acceso a Internet. Utilizaremos únicamente una dirección IP para el PAT, dos para los servidores (NAT estática) y el resto para los posibles nuevos servidores que tengamos que publicar en Internet. Evidentemente, siempre podemos poner más de un servicio en un mismo servidor (por ejemplo, web y correo, en el mismo servidor), con lo cual ahorraríamos nuevamente direcciones IP, pero será necesario tener en cuenta que demasiados servicios pueden colapsar nuestro servidor.

Ante esta elección, la configuración de nuestra NAT, utilizando la misma política de asignación de direcciones IP, podría ser la siguiente:

| Tipo de NAT | Dirección privada | Dirección pública |
|---|---|---|
| Estática | 10.1.1.51 | 200.4.4.9 |
| Estática | 10.1.1.52 | 200.4.4.10 |
| PAT | Rangos:<br>10.1.1.101 a 10.1.1.254<br>10.1.2.101 a 10.1.2.254<br>10.1.3.101 a 10.1.3.254<br>10.1.4.101 a 10.1.4.254 | 200.4.4.14<br>Puertos: 1024 al 5000 |

Analicemos ahora cómo actúa el encaminador que aplica PAT para poder gestio nar tantas conexiones TCP/IP contra servidores externos con una única dirección IP pública:




Cada vez que el encaminador de salida recibe una nueva conexión TCP o UDP contra el exterior, le asigna un puerto libre del rango especificado. De esta manera, todas las conexiones se publican en el exterior con la misma dirección IP, pero son distinguidas según el puerto TCP o UDP asignado. Esta relación se guarda en la tabla dinámica de traducciones para que, al llegar la respuesta del exterior, se pueda redireccionar hacia el cliente correspondiente.

Por ejemplo, pensemos en el caso planteado en la figura anterior, en el que el usuario A quiere establecer una conexión web (puerto 80) contra el servidor 140.1.2.3. Imaginemos que hay dos usuarios más que también quieren establecer conexiones web contra este servidor. Una vez establecidas estas conexiones, la tabla de traducciones PAT del encaminador de salida tendrá un aspecto como el siguiente:

| Origen privado | Destino | Origen público |
|---|---|---|
| 10.1.1.101 : 2012 | 140.1.2.3 : 80 | 200.4.4.14 : 1024 |
| 10.1.2.120 : 3378 | 140.1.2.3 : 80 | 200.4.4.14 : 1025 |
| 10.1.4.220 : 1906 | 140.1.2.3 : 80 | 200.4.4.14 : 1026 |

### 7.4. LSNAT (*Load Sharing NAT*)

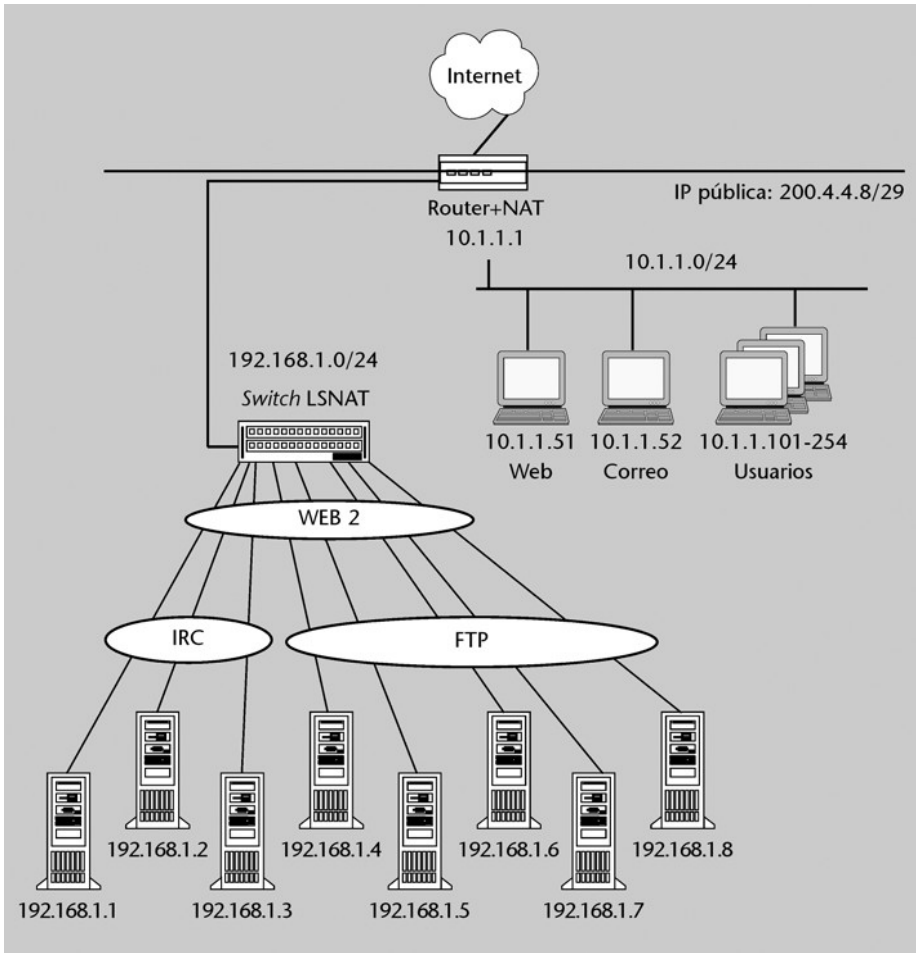
Este tipo de NAT está pensado para solucionar el problema que presentan los servidores de aplicaciones sobresaturados, como los portales o los buscadores. Si se da el caso de que el servicio que ofrecemos en Internet recibe tantas solicitudes que no las puede absorber, la primera solución que se nos puede plantear es la de comprar un servidor más potente. Normalmente, esto tiene un coste muy elevado y un problema grave: ¿y si falla este único servidor?

La solución a los dos problemas es replicar nuestro servidor; es decir, hay que comprar más servidores, pero de las mismas prestaciones que el que ya tenemos, opción que seguramente es más económica que comprar un servidor más potente. Una vez hecho esto, aplicaremos un sistema que publique todos los servidores en Internet como si fueran un único servidor, y repartiremos la carga entre todos; de esta manera, en caso de que uno falle, redirigirá las peticiones hacia el resto, de modo que el usuario no reciba sus consecuencias.

En este caso tendremos *n* servidores con diferentes direcciones privadas, los cuales estarán identificados por la misma dirección pública o *virtual* IP *address*.

Normalmente la LSNAT requiere una alta carga del procesador, por lo cual se suele hacer en un conmutador de altas prestaciones, y no en el encaminador perimetral.

LSNAT permite la distribución de la carga entre distintos servidores de aplicaciones de manera que, en Internet, estos servidores son publicados con una dirección pública única.




Con estos datos, la configuración LSNAT del conmutador de altas prestaciones sería la siguiente:

| Tipo de NAT | Dirección privada | Dirección pública |
|---|---|---|
| LSNAT | Rango: 192.168.1.1 a 192.168.1.8 | 200.4.4.11 : 80 |
| LSNAT | Rango: 192.168.1.4 a 192.168.1.8 | 200.4.4.12 : 21 |
| LSNAT | Rango: 192.168.1.1 a 192.168.1.3 | 200.4.4.13 : IRC |

Cualquier usuario que quiera acceder al servicio web utilizará la dirección virtual IP (VIP) 200.4.4.11. Internamente, esta dirección se corresponde con ocho direcciones privadas distintas, una para cada servidor web. Es decir, que el conmutador de altas prestaciones elegirá la dirección privada más adecuada, aunque este proceso será completamente transparente al usuario, que pensará que se conecta a un único servidor web.

## 8. IPv6

Pv6 es la evolución del IP de siempre (la versión 4). La principal mejora que aporta, que de hecho es su razón de ser, es el incremento de direcciones IP disponibles porque en lugar de 32 bits utiliza 128 para representarlas.

Cuando se empezó a hablar del IP de nueva generación, inicialmente se le dio este nombre, *IP Next Generation*, o *IPng*, en honor de la serie de televisión *Star Trek: the next generation*. El nombre definitivo que recibió, no obstante, fue IP versión 6, o IPv6. IP versión 5 fue un intento que no triunfó de protocolo de *streaming* para audio y vídeo.

IPv4 permite $2^{32}$, o sea, 4.294.967.296 direcciones. Aunque parece un número bastante grande como para no tener que sufrir, lo cierto es que, como se ha visto, el esquema jerárquico de las direcciones y la manera como se distribuyeron las primeras redes han hecho que casi no queden libres. Por eso se propuso evolucionar IP hacia un protocolo que permitiera aún más direcciones. Los 128 bits de IPv 6 permiten $2^{128}$, o sea, $5 \cdot 10^{28}$ direcciones, número suficiente para que todos los dispositivos imaginables en el mundo (y más) que se quieran conectar a Internet lo puedan hacer.

IPv6 fue adoptado por la IETF en 1994, y parecía que su despliegue sería rápido porque era muy necesario. En realidad, sin embargo, eso no ha sido así. En diciembre del 2005 sólo una pequeña parte de las máquinas en Internet utilizaban IPv6. Por un lado, migrar de v4 a v6 todas las máquinas existentes y la necesaria cohabitación entre los dos estándares durante un tiempo no está resultando fácil; y por el otro, las técnicas de NAT y CIDR aplicadas con bastante éxito han contribuido a aligerar la presión sobre IPv4.

Sea como sea, se prevé que antes del 2010 todo Internet ya utilice IPv6.

# Glosario

***American Standard Code for Information Interchange*** *m* Codificación de fuente que permite distinguir 128 mensajes o caracteres distintos. El conjunto de caracteres está dividido en letras mayúsculas y minúsculas no acentuadas, cifras, signos de puntuación y caracteres de control de transmisión. sigla: ASCII

**analógico -a** *adj* Tipo de señal, información o parámetro cuya variación puede tomar un valor cualquiera dentro de un conjunto infinito de valores. Son ejemplos de ello todos los parámetros físicos, como la posición, la velocidad, la temperatura, etc.

**ARP** *m* Protocolo para el mapeado automático de direcciones MAC con direcciones IP. *en* address resolution protocol

**ASCII** *m Véase* American Standard Code for Information Interchange.

**binario –a** *adj* Tipo de señal, información o parámetro cuya variación puede tomar un valor cualquiera dentro de un conjunto de dos valores. Normalmente, los valores se representan como 0 y 1, como sí o no, o como activado o desactivado.

**bit (binary digit)** *m* Volumen mínimo de información. Puede tener dos valores, normalmente asociados a 0 y 1, o bien a sí o no, o bien a activado o desactivado.

***bridge*** *m Véase* puente.

***byte*** *m* Conjunto de 8 bits.

**canal de transmisión** *m* Conjunto de sistemas pasivos (medios de transmisión) y de sistemas activos (amplificadores, ecualizadores, repetidores) que transportan la señal entre el transmisor y el receptor.

**circuito de datos** *m* Conjunto formado por el canal de datos y los equipos de codificación y descodificación externos.

**codificación de fuente** *f* Código utilizado por dos equipos terminales de datos. Podéis ver también *código*.

**código detector de errores** *m* Transformación que se aplica a la trama para obtener el valor de los bits que se han añadido y detectar de esta forma si una trama llega con errores. Los códigos más utilizados son la paridad, la paridad longitudinal y transversal y los códigos CRC.

**código** *m* Conjunto de palabras código con su correspondiente representación binaria. Podéis ver también *palabra código*.

**concentrador** *m* Equipo para la interconexión de terminales de una misma LAN. Todos los equipos interconectados por medio de un concentrador pertenecen a un mismo dominio de colisión. *en* hub

**conmutador** *m* Equipo para la interconexión de terminales de una misma LAN. Hay algún tipo que permite la interconexión de diferentes concentradores. *en* switch

**control de flujo** *m* Mecanismo que utiliza el nivel de enlace para evitar que las tramas lleguen antes de lo que es capaz de procesar. Normalmente se utiliza un mecanismo de ventana.

**cortafuegos** *m* Enrutador con capacidad de filtrado de paquetes de acuerdo con unas reglas establecidas. Estas últimas se definen en cada instalación de acuerdo con los problemas de seguridad de la red en que se instala. *en* firewall

***cracker*** *m* En argot, individuo que se dedica a entrar en sistemas ajenos mediante la red, aprovechando defectos de su configuración o gestión, o bien defectos de los sistemas que la componen. Si bien el término *hacker* está muy difundido para esta definición, no es correcto, ya que su significado original se refiere simplemente a alguien experto en programación.

**DCE (*Data Circuit-terminating Equipment*)** *m Véase* equipo terminal de circuito de datos.

**diagrama de tiempo** *m* Representación espaciotemporal de la comunicación entre dos entidades. Permite representar gráficamente un protocolo y el tiempo invertido en cada una de las acciones.

**digital** *adj* Tipo de señal, información o parámetro cuya variación puede tomar un valor cualquiera dentro de un conjunto finito de valores.

**distancia de Hamming** *f* Número de bits de diferencia que hay entre dos palabras código.

**DTE (*Data Terminal Equipment*)** *m Véase* equipo terminal de datos.

**elemento activo** *m Véase* sistema activo.

**elemento pasivo** *m Véase* sistema pasivo.

**encaminador** *m* Dispositivo de la capa de red que utiliza una o más métricas para encontrar el camino óptimo por el cual debe viajar la información. Pasa la información de una red a otra según la información de la capa de red. *en* router

**enlace multipunto** *m* Situación en la que una comunidad de computadores se comunican mediante un enlace compartido. Para poder comunicarse sin interferirse entre sí uno de los mismos se encarga de arbitrar las comunicaciones.

**equipo terminal de circuito de datos** *m* Equipo que hace la conversión de bits a señales eléctricas o electromagnéticas (codificación de canal). sigla: DCE sin.: terminal de circuito de datos

**equipo terminal de datos** *m* Equipo receptor y generador de datos. Presenta los datos al destinatario (por ejemplo un operador humano) y le facilita su introducción. Efectúa la codificación y la descodificación de fuente. sigla: DTE. sin.: terminal de datos

**Ethernet** *m* Estándar *de facto* parcialmente compatible con el IEEE802.3. Ambos utilizan el algoritmo CSMA/CD 1 –persistente a 10 Mbps o 100 Mbps. Ethernet (red de éter) hace alusión tanto al hecho de que antiguamente se creía que el espacio que separaba a los astros estaba formado por un material (éter) capaz de transportar la luz, como al hecho de que la arquitectura de su red competidora, la Token Ring, necesitaba bastantes más elementos, aparte del éter.

***firewall*** *m Véase* cortafuegos.

***framing*** *f Véase* sincronización de trama.

**fuente de alimentación** *f* Equipo eléctrico y/o electrónico que da energía eléctrica al equipo electrónico que la necesita. La fuente de alimentación obtiene la energía de la red eléctrica (220 V) en equipos de sobremesa y de potencia media o alta, o de baterías para equipos portátiles.

***full-duplex*** *f* Capacidad para la transmisión y recepción de datos de manera simultánea entre una estación origen y una estación destino.

***Global System for Mobile Communication*** *m* Sistema de telefonía inalámbrico. Transmite la voz digitalmente. Mediante sistemas predictivos (compresión) consigue reducir la capacidad necesaria a 13 kbps. sigla: GSM

**GSM** *m Véase* Global System for Mobile Communication.

***hacker*** *m Véase cracker*.

***half duplex*** *f* Capacidad para la transmisión de datos en una única dirección a la vez entre una estación origen y una estación destino.

***hardware*** *m* Conjunto de sistemas electrónicos capaces de llevar a cabo una tarea. Normalmente son sistemas basados en la ejecución de un código ejecutable *(software)*.

**hercio (Hz)** *m* Unidad de medida de frecuencia. Es equivalente a uno partido por segundo ($s^1$) o ciclos partido por segundo (s).

***hub*** *m Véase* concentrador.

**ICMP** *m* Protocolo de nivel de red utilizado sólo para tareas de gestión del nivel de red. *en* Internet control message protocol.

**IEEE** *m Véase* Institute of Electrical and Electronics Enginneers.

***initial sequence number*** *m Véase* ISN.

**inserción de caracteres** *f* Mecanismo de transparencia que se utiliza en las transmisiones orientadas a carácter.

**inserción de ceros** *f* Mecanismo de transparencia que se utiliza en las transmisiones orientadas a bit.

**Institute of Electrical and Electronics Enginneers** *m* Organización profesional que se encarga, entre otros temas, del desarrollo de estándares de comunicaciones y de red. Los estándares LAN del IEEE son los estándares predominantes en redes de área local de hoy día. sigla: IEEE

***integrated services digital network*** *f Véase* RDSI.

**interfaz** *f* Punto de acceso a un sistema determinado. Una interfaz debe llevar asociada una definición del método de acceso (protocolo, instrucciones, parámetros, etc.).

**interferencia intersimbólica** *f* Solapamiento entre símbolos consecutivos de una señal de datos. sigla: ISI

***Internet control message protocol*** *m Véase* ICMP.

**Internet protocol** m *Véase* IP.

**IP** *m* Protocolo de nivel de red utilizado en Internet. *en* Internet protocol

**ISI** *f Véase* interferencia intersimbólica.

**ISN** *m* Número de secuencia inicial que es asignado al principio de la conexión. A partir de este momento, el TCP numera los *bytes* consecutivamente a partir de este número. *en* initial sequence number

**LAN** *f* red de área local. *en* local area network

**LAN virtual** *f* Grupo de dispositivos de una LAN que se configuran de manera que se puedan comunicar como si realmente estuvieran conectados al mismo

medio físico cuando, en realidad, están situados en segmentos de LAN distintos. VLAN están basadas en conexiones lógicas, no físicas. sigla: VLAN

**línea de transmisión** *f* Medio guiado de transmisión. El par de hilos de cobre, el coaxial y la fibra óptica son ejemplos de ello.

**lista de recepción** *f* Lista en la cual el secundario guarda las tramas para poder entregarlas en orden.

**lista de transmisión** *f* Lista donde el primario guarda las tramas que se han enviado y que están pendientes de confirmar, para poder retransmitirlas en caso de error.

***local area network*** *f* *Véase* LAN.

**MAC** *m* Protocolo de control de acceso al medio utilizado por las estaciones de una misma red de área local como, por ejemplo, Ethernet o IEEE802.5 (Token Ring). *en* medium access control

**máscara de red** *f* Máscara de 32 bits utilizada en IP para indicar los bits de una dirección IP que se utilizan como una dirección de subred. Muchas veces, simplemente se le denomina *máscara*. *en* subnet mask

***maximum segment size*** *f* *Véase* MSS.

**mecanismo de transparencia** *m* Mecanismo que se utiliza para no confundir los datos de la trama con el identificador de final de trama.

**medio de transmisión** *m* Soporte físico por el cual se pueden propagar ondas y en general una señal.

***medium access control*** *m* *Véase* MAC.

**mensaje** *m* Secuencia de palabras código que representan una información determinada.

**módem** *m* Contracción de *modulador/demodulador*. DCE que genera (y recibe) una codificación de canal del tipo modulación digital.

**MSS** *f* Define la longitud máxima de datos que el TCP enviará. Se especifica durante el establecimiento de la conexión. *en* maximum segment size

**multiplexar** *v tr* Conjuntar diferentes señales (o secuencias de bits) en una sola.

**NAT** *f* *Véase* network address translation.

***network address translation*** *m* Técnica que proporciona una dirección pública a usuarios de una red que trabajan con enrutado privado. De esta manera los usuarios pueden acceder a Internet. sigla: NAT

**nivel de enlace** *m* Nivel que hay inmediatamente por encima del nivel físico. Su función más importante es transmitir información de manera fiable utilizando el flujo no fiable de bits del nivel físico.

***open systems interconnection*** *m* Modelo de comunicación entre dispositivos que cooperan entre sí. Define una arquitectura de siete capas de funciones de comunicación. sigla: OSI

**OSI** *m Véase* open systems interconnection.

**palabra código** *f* En el contexto de los códigos detectores de errores, conjunto de *P* + Q bits que resultan de la transformación que aplica el código a los *P* bits que se deben proteger.

**PAT** *f Véase* port address translation.

***piggybacking*** *m* Técnica que consiste en enviar las confirmaciones dentro de un campo de las tramas de información.

***port address translation*** *m* Junto con NAT, PAT utiliza direcciones de puerto para hacer una correspondencia entre direcciones IP locales y globales. sigla: PAT

**primario** *m* Estación que envía las tramas de información. En el protocolo HDLC, es la estación que puede enviar comandos.

**protocolo** *m* Formato y conjunto de reglas que se definen para la comunicación entre dos entidades del mismo nivel, en la arquitectura de comunicaciones.

**protocolo no orientado a la conexión** *m* Protocolo que no tiene fase de inicialización ni de desconexión.

**protocolo orientado a la conexión** *m* Protocolo que tiene una fase de inicialización y una fase de desconexión.

**puente** *m* Tipo de conmutador utilizado para interconectar las LAN. *en* bridge

**ráfaga de error** *f* Número de bits que hay entre el primer bit erróneo de una trama y el último, los dos incluidos.

**RDSI** *f* red digital de servicios integrados. *en* integrated services digital network

***read only memory*** *f Véase* ROM.

**recuperación automática de errores** *f* Técnica que consiste en solicitar la retransmisión de una trama cuando se detecta un error. Los protocolos más conocidos que utilizan esta técnica son el *Stop* & *Wait*, el *Go back N* y la retransmisión selectiva.

**redundancia** *f* Bits adicionales que añade el código detector de errores.

**ROM** *f* Memoria sólo de lectura utilizada para almacenar datos y/o programas pequeños. *en* read only memory

**router** *m Véase* encaminador.

**secundario** *m* Estación –en el contexto del módulo– que envía las tramas de confirmación. En el protocolo HDLC, se trata de la estación que envía las respuestas.

***shielded twisted pair*** *m* Medio de transmisión formado por dos cables de cobre aislados trenzados uno con otro. Entre la cubierta exterior de plástico y el par, se coloca un apantallamiento metálico. sigla: STP

**sincronización de trama** *f* Mecanismo que utiliza el nivel de enlace para determinar el inicio y el final de una trama dentro del flujo de bits o caracteres que llega del nivel físico. en *framing*

**sistema activo** *m* Sistema que puede emitir una potencia más alta de la que recibe mediante la utilización de una fuente de alimentación y circuitería electrónica. También podéis consultar *sistema pasivo*. sin.: elemento activo

**sistema pasivo** *m* Sistema que emite menos potencia de la que recibe, normalmente a causa de pérdidas de transmisión (radiación, resistencia, desadaptación, etc.). No tiene circuitería electrónica, a menudo es simplemente una línea de material conductor de energía (línea de cobre, fibra óptica, etc.). sin.: elemento pasivo

***software*** *m* Conjunto de instrucciones (programa ejecutable) que desarrollan una tarea determinada y que son interpretadas por el *hardware*.

**STP** *m Véase* shielded twisted pair.

***subnet*** *f Véase* subred.

***subnet mask*** *f Véase* máscara de red.

**subred** *f* Se habla de subred en redes IP cuando una red tiene una dirección de subred particular. Las redes y subredes son redes que el administrador de la red acostumbra a segmentar arbitrariamente para obtener una estructura de enrutado jerárquica, y mantener la misma dirección de red para el conjunto. *en* subnet

***switch*** *m Véase* conmutador.

**TCP/IP** *m Véase* transmission control protocol / Internet protocol.

**terminal de circuito de datos** *f Véase* equipo terminal de circuito de datos.

**terminal de datos** *f Véase* equipo terminal de datos.

**trama** *f* Estructura en la cual el nivel de enlace organiza los campos de control y de información para que puedan ser identificados dentro del flujo de bits del nivel físico.

***transmission control protocol / Internet protocol*** *m* Modelo de comunicación entre dispositivos que cooperan entre ellos y que se intercambian información. Es la base de funcionamiento de Internet. sigla: TCP/IP

***unshielded twisted pair*** *m* Medio de transmisión formado por dos cables de cobre aislados trenzados uno con otro. La cubierta exterior del par es de plástico. sigla: UTP

**UTP** *m Véase* unshielded twisted pair.

***vef*** *Véase* velocidad de transmisión efectiva o eficaz.

**velocidad de modulación** *f* Número de símbolos transmitidos por unidad de tiempo. Normalmente se mide en símbolos/s o baudios.

**velocidad de propagación** *f* Velocidad a la que se propaga una señal eléctrica o electromagnética dentro de su medio de transmisión. La velocidad de propagación de la luz en el vacío (o velocidad de la luz) es $c = 3 \cdot 10^8$ m/s. Para medios conductores (señales eléctricas) la velocidad de propagación puede descender en algunos casos hasta 2/3 *c*.

**velocidad de transmisión efectiva o eficaz** *f* Velocidad media a la que se transmiten los bits de información. Se define como: $vef = E \cdot vt$, donde *E* es la eficiencia del enlace, y *vt*, la velocidad de transmisión máxima.

**velocidad de transmisión** *f* Número de bits transmitidos por unidad de tiempo. Normalmente se mide en bits/s o bps (bits *per second*).

**ventana de recepción** *f* Número máximo de tramas que tiene que almacenar el secundario.

**ventana de transmisión** *f* Número de tramas que el primario está autorizado a transmitir sin confirmar, dado que todas las anteriores están confirmadas.

**ventana óptima** *f* Ventana mínima que permite alcanzar una eficiencia del protocolo del 100%.

**VLAN** *f Véase* LAN virtual.

# Bibliografía

**Amato, V.** (1999). *Cisco Networking Academy Program: Second-Year Companion Guide*. Cisco Press. Black, U. (1997). *Tecnologías emergentes para redes de computadoras* (2.ª ed.). Pearson Educación. García, J. (1997). *Redes de alta velocidad*. Editorial Ra-ma.

**Benedetto, S.; Biglier, E.; Castellani, V.** (1987). *Digital Transmission Theory*. Englewood Cliffs: Prentice Hall.

**Bertsekas, D.; Gallager, R.** (1992). *Data Networks* (2.ª ed.) Englewood Cliffs: Prentice Hall.

**Carlson, A. B.** (1986). *Communication systems. An introduction to signals and noise in electrical communication* (3.ª ed.). Nueva York: McGraw-Hill.

**Chapman, D.B.; Zwicky, E.D.** (1995). *Building Internet Firewalls*. Sebastopol: O'Reilly & Associates, Inc.

**Comer, D.E.** (1995). "Principles, Protocols and Architecture". *Internetworking with TCP/IP* (vol. 1). Hertfordshire: Prentice Hall.

**Halsall, F.** (1998). *Comunicación de datos, redes de computadores y sistemas abiertos* (4.ª ed.). México: Pearson Educación.

**Pierce, J. R.; Noll, A. M.** (1995). *Señales: la ciencia de las telecomunicaciones*. Barcelona: Reverté.

**Sloman, M; Kramer, J.** (1987). *Distributed Systems and Computer Networks*. Englewood Cliffs: Prentice Hall.

**Stallings, W.** (2000). *Comunicaciones y redes de computadores* (6.ª edición). Madrid: Prentice Hall.

**Stevens, W.R.** (1998). "The Protocols" *TCP/IP Illustrated* (vol. I). Wilmington: Addison- Wesley, 1994.

**Tanenbaum, A.S.** (2003). *Redes de computadoras* (4.ª ed.). México: Pearson Educación.

**Thomas, S.A.** (1996). *IPng and the TCP/IP protocols, implementing the next generation Internet*. Nueva York: John Wiley & Sons, Inc.

**Wright, G.R.; Stevens, W.R.** (1995). "The implementation". *TCP/IP Illustrated* (vol. II). Wilmington: Addison-Wesley.