



UNIVERSIDAD MAYOR DE SAN SIMON  
FACULTAD DE CIENCIAS Y TECNOLOGIA  
DEPARTAMENTO INFORMATICA – SISTEMAS



**TEXTO DE REFERENCIA**  
**REDES DE**  
**COMPUTADORAS**



**Jorge Walter Orellana Araoz**

**2016**

**Tema I**

**INTRODUCCIÓN A REDES DE COMPUTADORAS**

Las Redes de Computadoras y sobretodo el Internet, son aspectos clave de las TICs (Tecnologías de la información y las comunicaciones), teniendo gran influencia sobre el desarrollo tecnológico y científico. Es así, que actualmente no es posible concebir una sociedad donde el intercambio de información mediante las computadoras no sea un procedimiento cotidiano y en ocasiones indispensable. Esto es posible gracias al aporte de la Informática y las Telecomunicaciones.

**Telecomunicaciones**, es el conjunto de medios técnicos que permiten la comunicación a distancia. Normalmente se trata de transmitir información sonora (voz, música) o visual (imágenes estáticas o en movimiento) por ondas electromagnéticas a través de diversos medios (aire, vacío, cable de cobre, fibra óptica, etc.). La información se puede transmitir de forma analógica, digital o mixta, pero esto es transparente al usuario, que la maneja de forma analógica únicamente.

**Telemática** ( fusión de telecomunicaciones e informática) es el uso de las telecomunicaciones para enriquecer las posibilidades de la informática, es decir, del uso de medios de comunicación a distancia para conexiones informáticas (computadora-computadora u computadora-periférico). La información puede transmitirse de forma analógica, digital o mixta, pero esto es transparente al usuario, que la maneja de forma digital únicamente.

Las *redes de computadoras* son dos o más computadoras que se comunican por medio de la telemática. No se considera la comunicación entre un computador y un periférico (impresora, scanner, etc.) independientemente de la distancia a la que dicha comunicación se produzca o el tipo de medios utilizados para ella.

Se puede definir una red como “una interconexión, tanto a nivel físico como lógico, de un conjunto de computadores, periféricos y medios, que permitirían compartir los recursos del sistema por la totalidad de los distintos integrantes”.

**1.1 HARDWARE DE REDES**

Hay muchos parámetros que conforman los tipos de redes, hablaremos de tres dimensiones: la tecnología de transmisión, la topología o disposición en el espacio y la escala o extensión geográfica.

**1.1.1 Tecnología de transmisión.**

Las redes se clasifican en difusión y punto a punto.

• **Redes de difusión (Broadcast)**

En las redes de difusión, el medio de transmisión es compartido por todas las computadoras interconectadas. Cada mensaje transmitido es para un único destinatario, cuya dirección aparece en el mensaje, pero para saberlo cada máquina de la red recibe o “escucha” cada mensaje, analiza la dirección de destino y averigua si va o no dirigido a ella; las normas de buena educación “telemática” establecen que un computador debe descartar sin mas análisis todo mensaje que no vaya dirigido a él; sin embargo, algunos programas llamados “sniffers” se dedican a “husmear” todo lo que pasa por el cable, independientemente de quien sea su destinatario; con un sniffer es muy fácil capturar cualquier cosa.

Cuando un dispositivo puede emitir y muchos pueden recibir se distinguen diferentes tipos de comunicaciones:

- a) Unicast: cuando un dispositivo desea enviar un mensaje solo a otro dispositivo.
- b) Multicast: cuando un dispositivo desea enviar un mensaje a muchos dispositivos.

c) Broadcast: cuando un dispositivo desea enviar un mensaje a todos los dispositivos conectados al enlace.

- **Redes punto a punto (Point-to-point)**

Las redes punto a punto se construyen por medio de conexiones entre pares de computadoras, también llamadas líneas, enlaces, circuitos o canales. Una vez un paquete es depositado en la línea el destino es conocido de forma única y no es preciso en principio que lleve la dirección de destino.

Los enlaces que constituyen una red punto a punto pueden ser de tres tipos de acuerdo con el sentido de la transmisión:

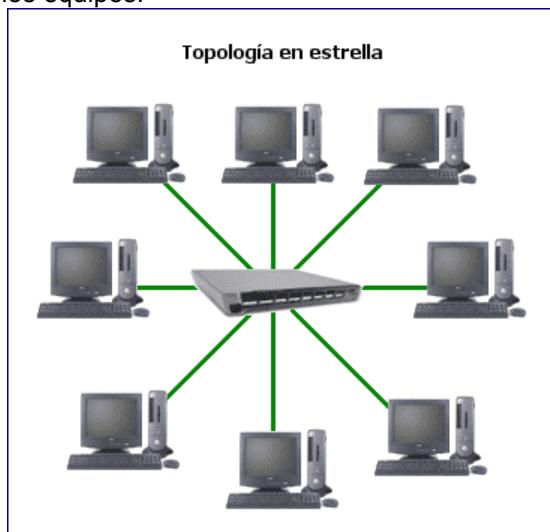
- a) Simplex: En este modo la comunicación es unidireccional. El dispositivo emisor siempre transmite y el dispositivo receptor siempre recibe. Por ejemplo, la comunicación entre un teclado y el ordenador, la tarjeta de video y el monitor, etc.
- b) Dúplex (half dúplex): En este modo cada nodo puede transmitir o recibir, pero no al mismo tiempo. Cuando uno está enviando, el otro está recibiendo y vice versa. Un ejemplo, las radios Walkie-Talkies.
- c) Full-Dúplex: Los dispositivos pueden enviar y recibir simultáneamente. Como la capacidad del enlace se divide entre el número de dispositivos que acceden al medio, la tasa de transferencia se divide también por este factor. La ventaja es que pueden estar transmitiendo continuamente.

### 1.1.2 Topología o Disposición en el Espacio

Se refiere a la forma que tiene la red, física o lógicamente (que no tiene por qué coincidir) y es un aspecto fundamental porque nos da una buena idea de las propiedades de la red. Existen cinco tipos básicos de topologías: bus, anillo, estrella, árbol y malla.

- **Estrella.**

Esta topología se caracteriza por existir en ella un punto central, o más propiamente nodo central, al cual se conectan todos los equipos.



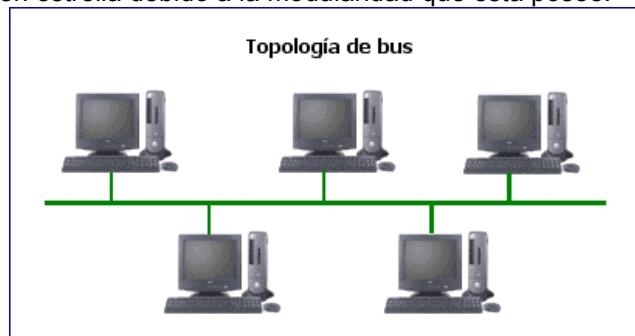
La máxima vulnerabilidad se encuentra precisamente en el nodo central, ya que si este falla, toda la red fallaría. Sin embargo presenta como principal ventaja una gran modularidad, lo que permite aislar una estación defectuosa con bastante sencillez y sin perjudicar al resto de la red.

Para aumentar el número de estaciones, o nodos, de la red en estrella no es necesario interrumpir, ni siquiera parcialmente la actividad de la red, realizándose la operación casi inmediatamente. La topología en estrella es empleada en redes Ethernet y ArcNet.

- **Bus**

Todos los nodos que componen la red quedan unidos entre sí linealmente, uno a continuación del otro. El cableado en bus presenta menos problemas logísticos, puesto que no se acumulan montones de cables en torno al nodo central, como ocurriría en una disposición en estrella. Pero, en contra, tiene la desventaja de que un fallo en una parte del cableado detendría el sistema, total o parcialmente, en función del lugar en que se produzca. Es además muy difícil encontrar y diagnosticar las averías que se producen en esta topología.

Debido a que en el bus la información recorre todo el bus bidireccionalmente hasta hallar su destino, la posibilidad de interceptar la información por usuarios no autorizados es superior a la existente en una Red en estrella debido a la modularidad que ésta posee.



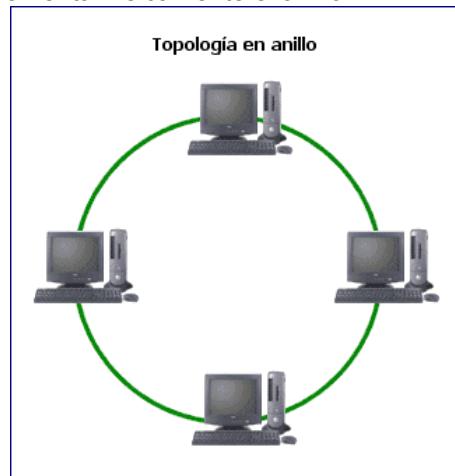
La red en bus posee un retardo en la propagación de la información mínimo, debido a que los nodos de la red no deben amplificar la señal, siendo su función pasiva respecto al tráfico de la red. Esta pasividad de los nodos es debida más bien al método de acceso empleado que a la propia disposición geográfica de los puestos de red. La Red en Bus necesita incluir en ambos extremos del bus, unos dispositivos llamados terminadores, los cuales evitan los posibles rebotes de la señal, introduciendo una impedancia característica (50 Ohm.)

Añadir nuevas estaciones a una red en bus, supone detener al menos por tramos, la actividad de la red. Sin embargo es un proceso rápido y sencillo. Es la topología tradicionalmente usada en redes Ethernet.

- **Anillo**

El anillo, como su propio nombre indica, consiste en conectar linealmente entre sí todos los computadoras, en un bucle cerrado. La información se transfiere en un solo sentido a través del anillo, mediante un paquete especial de datos, llamado testigo (token), que se transmite de un nodo a otro, hasta alcanzar el nodo destino.

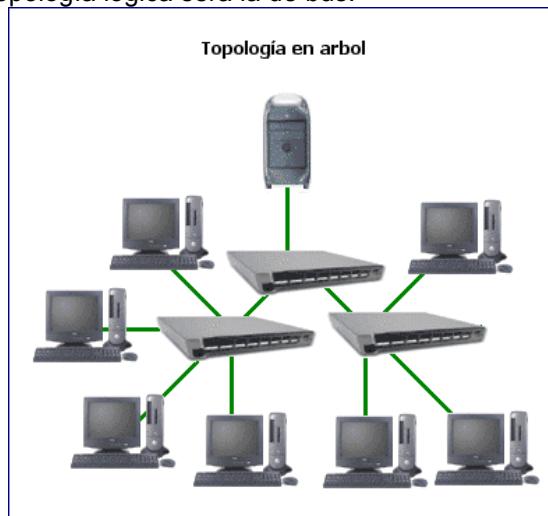
El cableado de la red en anillo es el más complejo, debido por una parte al mayor coste del cable, así como a la necesidad de emplear unos dispositivos denominados Unidades de Acceso Multiestación (MAU) para implementar físicamente el anillo.



A la hora de tratar con fallos y averías, la red en anillo presenta la ventaja de poder derivar partes de la red mediante los MAU's, aislando dichas partes defectuosas del resto de la red mientras se determina el problema. Un fallo, en una parte del cableado de una red en anillo, no debe detener toda la red. La adición de nuevas estaciones no supone una complicación excesiva, puesto que una vez más los MAU's aíslan las partes a añadir hasta que se hallan listas, no siendo necesario detener toda la red para añadir nuevas estaciones. Dos ejemplos de red en anillo serían Token-Ring y FDDI (fibra óptica).

- **Árbol**

Es una generalización de la topología en estrella donde un conmutador puede ser conectado directamente a otro conmutador. Igual que ocurre en la topología en estrella, si los conmutadores son pasivos entonces la topología lógica será la de bus.

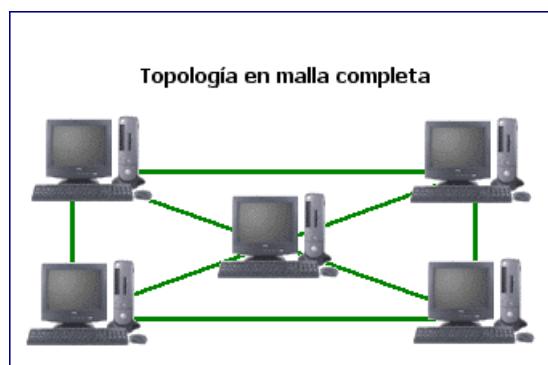


La topología en árbol es muy frecuente porque los situación geográfica de los nodos muchas veces impide que todos los enlaces físicos se conecten a un único conmutador. Tiene por tanto las mismas ventajas e inconvenientes que la configuración en estrella.

- **Malla**

Es la configuración más cara porque cada estación se conecta a cualquier otra mediante un enlace punto a punto. Son necesarios en total  $n(n - 1)/2$  enlaces para  $n$  estaciones. Además, cada estación necesita  $n-1$  interfaces de entrada/salida hacia  $n - 1$  enlaces físicos.

La gran ventaja de la topología en malla radica en que la tasa de transferencia entre cada dos estaciones es la máxima posible, independientemente de que otras estaciones se estén comunicando. Por otra parte, la red es muy robusta frente a perdidas de enlaces porque pueden establecerse muchos caminos alternativos. Otra ventaja es la privacidad o seguridad en las transferencias, puesto que los mensajes no tienen que ser manipulados por ningún otro nodo intermedio.



### 1.1.3. Escala (extensión geográfica)

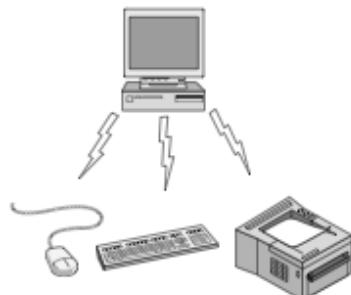
Por su tamaño las redes se clasifican en:

Distancia entre procesadores	Procesadores ubicados en el mismo	Ejemplo
1 m	Metro cuadrado	Red de área personal
10 m	Cuarto	Red de área local
100 m	Edificio	
1 km	Campus	Red de área metropolitana
10 km	Ciudad	
100 km	País	Red de área amplia
1,000 km	Continente	
10,000 km	Planeta	Internet

- **Red de Área Personal (PAN – Personal Area Network)**

Permite que los dispositivos se comuniquen alrededor de una persona (computador conectado con sus periféricos). Casi todos los equipos tienen un monitor, teclado, ratón y la impresora conectados, que se debe hacer con los cables. Una opción es utilizar una red inalámbrica de corto alcance llamada Bluetooth para conectar estos componentes sin necesidad de cables. La idea es que si los dispositivos tienen Bluetooth, entonces no se necesita cables. Solo hay que ponerlos juntos, encenderlos y trabajan. Las redes Bluetooth utilizan el paradigma maestro-esclavo. La computadora (PC) es normalmente el maestro, comunicándose con el ratón, teclado, etc., como esclavos. El maestro le dice a los esclavos qué direcciones usar, cuándo pueden transmitir, por cuánto tiempo pueden transmitir, frecuencias que pueden utilizar, etc.

Bluetooth se puede utilizar en otras configuraciones también, como conectar un auricular a un teléfono móvil sin cables, un reproductor de música digital para el coche, un dispositivo médico incrustado como un marcapasos, bombas de insulina, etc. Las PAN también se pueden construir con otras tecnologías que se comunican a través de distancias cortas, como la RFID en las tarjetas inteligentes y los libros de la biblioteca.



- **Red de Área Local (LAN - Local Area Network)**

Es una red pequeña. Comprende desde dos computadoras conectadas entre sí a través de un único enlace, hasta una red de unos pocos kilómetros de longitud. Es una red privada que opera dentro un edificio, una casa, oficina o una fábrica. Son ampliamente utilizados para conectar computadoras personales y electrónica de consumo para compartir recursos y el intercambio de información.

El alcance limitado de las LANs permite saber el tiempo máximo que un paquete tardará en llegar de un extremo a otro de la red, lo cual permite aplicar diseños que de otro modo no serían posibles, y simplifica la gestión de la red.

Como consecuencia del alcance limitado y del control en su cableado, las redes locales suelen tener un retardo muy bajo en las transmisiones (decenas de microsegundos) y una tasa de errores muy baja.

Las redes LAN cableadas utilizan una gama de diferentes tecnologías de transmisión. La mayoría utiliza cables de cobre, aunque algunos utilizan fibra óptica. Las redes LAN cableadas transmiten a

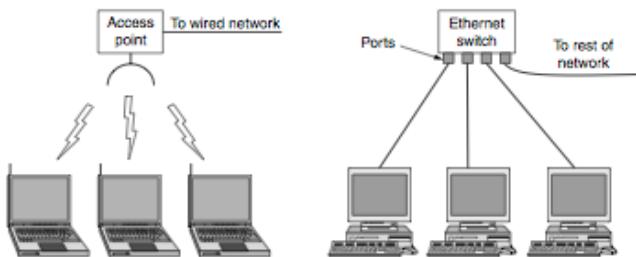
velocidades de 100 Mbps hasta 1 Gbps, tienen bajo retardo (microsegundos o nanosegundos), y tienen pocos errores. Las redes LAN nuevas pueden operar hasta 10 Gbps.

La topología de muchas LANs cableadas se construye a partir de enlaces punto a punto. El estándar es IEEE 802.3, llamado Ethernet, que es el tipo más común de la LAN cableada. Cada computador habla el protocolo Ethernet y se conecta a una caja llamada conmutador (switch) con un enlace punto a punto. El trabajo del switch es transmitir paquetes entre computadores que están conectados a él, mediante la dirección de cada paquete determina a qué computadora debe enviarlo. Ethernet conmutada (switched Ethernet) es una versión moderna del diseño original de Ethernet que transmitía todos los paquetes a través de un único cable lineal.

Las redes LAN inalámbricas son muy populares en los hogares, edificios de oficinas antiguos, cafeterías y otros lugares donde es un problema el instalar cables. En estos sistemas, cada equipo tiene un módem de radio y una antena que utiliza para comunicarse con otros computadores o con un dispositivo central llamado AP (Access Point), router inalámbrico, o estación base, que retransmite paquetes entre los computadores inalámbricos y también entre ellos y la Internet.

El estándar para redes LAN inalámbricas se llama IEEE 802.11, conocida también como WiFi que transmite a velocidades desde los 11 a cientos de Mbps.

En comparación con las redes inalámbricas, las redes cableadas las superan en todas las dimensiones de desempeño. Es más fácil para enviar señales a través de un cable o por medio de una fibra que a través del aire.



#### • Redes de Área Metropolitana (MAN - Metropolitan Area Network)

Una MAN (Metropolitan Area Network) cubre una ciudad. El ejemplo más conocido de MAN es la red de televisión por cable. Cuando Internet comenzó a crecer, los operadores de redes de televisión por cable comenzaron a darse cuenta de que con algunos cambios en el sistema, podrían proveer un servicio de Internet en las partes no utilizadas del espectro. En redes cableadas los medios físicos de gran alcance como la fibra óptica han ampliado las LAN a ciudades y desplazado el concepto de MAN.

Los acontecimientos recientes en el acceso a Internet inalámbrico de alta velocidad han dado lugar a otra MAN, que se ha estandarizado como IEEE 802.16 y es popularmente conocido como WiMAX.

#### • Redes de Área Extensa (WAN - Wide Area Network)

Una WAN (Wide Area Network) se extiende por una amplia zona geográfica, a menudo un país o continente. Las redes de amplio alcance se utilizan cuando no es factible tender redes locales, bien porque la distancia no lo permite por el costo de la infraestructura o simplemente porque es preciso atravesar terrenos públicos en los que no es posible tender infraestructura propia. En todos estos casos lo normal es utilizar para la transmisión de los datos los servicios de una empresa portadora. Las redes WAN se implementan casi siempre haciendo uso de enlaces telefónicos que han sido diseñados principalmente para transmitir la voz humana, ya que este es el principal negocio de las compañías telefónicas. Normalmente la infraestructura está fuera del control del usuario, estando supeditado el servicio disponible a la zona geográfica de que se trate. Conseguir capacidad en redes WAN suele ser caro, por lo que generalmente se solicita el mínimo imprescindible.

La paulatina introducción de fibras ópticas y líneas digitales en las infraestructuras de las compañías portadoras las líneas WAN han reducido apreciablemente su tasa de errores; también se han mejorado las capacidades y reducido los costos. A pesar del inconveniente que en

ocasiones pueda suponer el uso de líneas telefónicas tienen la gran virtud de llegar prácticamente a todas partes, que no es poco. Con la excepción de los enlaces vía satélite, que utilizan transmisión broadcast, las redes WAN se implementan casi siempre con enlaces punto a punto.

La red de telefonía celular es otro ejemplo de WAN que usa la tecnología inalámbrica. Este sistema ya ha pasado por tres generaciones y 4G es la que actualmente se introduce. La primera generación era analógica y de voz solamente. La segunda generación era digital y de sólo voz. La tercera generación es digital y es por tanto de voz como de datos. Cada estación base celular cubre una distancia mucho más grande que una LAN inalámbrica, con un rango medido en kilómetros. Las estaciones base están conectadas entre sí por un enlace principal de red (backbone) que por lo general está cableado. La tasa de transmisión de datos de las redes celulares son a menudo del orden de 1 Mbps hasta unos cuantos cientos Mbps.

- **Internetworking o Trabajo entre Redes.**

Existen muchas redes en el mundo, a menudo con diferente hardware y software. Personas conectadas a una red a menudo quieren comunicarse con personas unidas a una diferente. El cumplimiento de este deseo requiere conectar redes diferentes, y con frecuencia incompatibles. Una colección de redes interconectadas se llama una interconexión de redes (*internet*). Si bien las clasificaciones de redes antes estudiadas tienen interés como medio de sistematizar su estudio, es obvio que en la realidad casi nunca se da uno de esos tipos en estado puro. Por ejemplo, una LAN (que normalmente será una red de tipo broadcast) casi siempre dispondrá de un router que la interconecte a una WAN (que generalmente consistirá en un conjunto de enlaces punto a punto). Esta interconexión de tecnologías diferentes se conoce como “internetworking”. El router que interconecta redes diferentes está físicamente conectado a todas las redes que se desean interconectar.

Cuando una red está formada por la interconexión de varias redes se le denomina *internet*. A principios de los setenta se creó en los Estados Unidos una internet mediante la unión de varias redes que utilizando medios de transmisión diversos empleaban un conjunto común de protocolos en el nivel de red y superiores, denominados TCP/IP. Con el tiempo la denominación Internet (con I mayúscula) terminó convirtiéndose en el nombre propio de dicha red, muy conocida en nuestros días. Internet utiliza redes ISP (proveedor de servicio de Internet) para conectar redes de empresas, redes en el hogar y muchas otras redes.

## 1.2 SOFTWARE DE REDES

La preocupación inicial en el diseño de redes fue el hardware y se ocuparon tardíamente del software. Actualmente el software de red es altamente estructurado y tan importante como el hardware

### 1.2.1 Modelo de capas o Arquitectura de capas

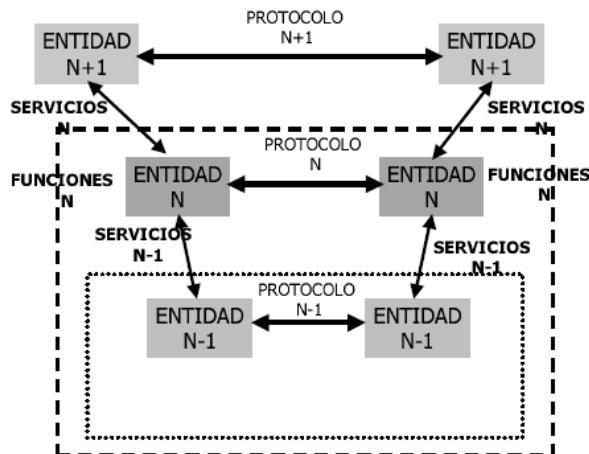
Es una forma de modularidad jerárquica, que es fundamental en el diseño de redes de datos. Los que diseñan el modulo intentan que provea los servicios que se desea, y lo realice en forma eficiente. Quien lo utiliza, lo ve como una “caja negra”, esto implica que quien lo utiliza, estará solamente interesado en los servicios que provee, y en sus entradas y salidas.

Se busca la estandarización de estos módulos, y esta estandarización permite reemplazarlo por uno nuevo que tenga funciones equivalentes, en módulos que tengan más capacidad, o sean más baratos. Los módulos equivalentes en las distintas capas, son llamados peer process, o peer modules. En el modelo la comunicación se realiza entre módulos equivalentes. El modelo de capas exige se establezca una conversación entre capas similares.

Las reglas de esta conversación se llaman el protocolo de la capa N. Las ideas básicas del modelo de capas son las siguientes:

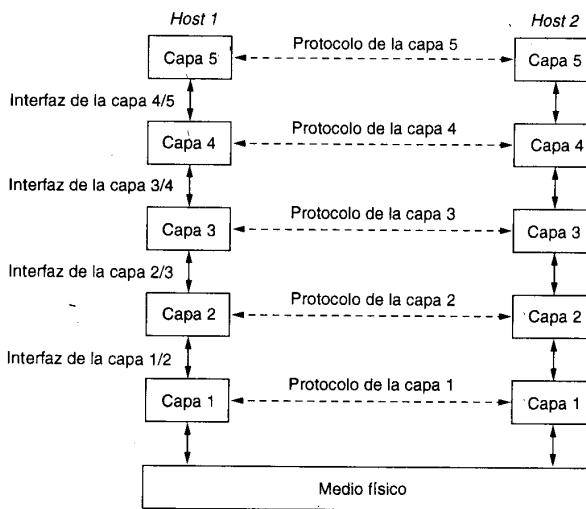
- La capa  $n$  ofrece una serie de servicios a la capa  $n+1$ .
- La capa  $n$  solo “ve” los servicios que le ofrece la capa  $n-1$ .

- La capa  $n$  en un determinado sistema solo se comunica con su homóloga en el sistema remoto (comunicación de igual a igual o “peer-to-peer”). Esa “conversación” se efectúa de acuerdo con una serie de reglas conocidas como *protocolo de la capa n*.



La comunicación entre dos capas adyacentes en un mismo sistema se realiza de acuerdo con una *interfaz*. La interfaz es una forma concreta de implementar un servicio y no forma parte de la arquitectura de la red.

La arquitectura de una red queda perfectamente especificada cuando se describen las capas que la componen, su funcionalidad, los servicios que implementan y los protocolos que utilizan para hablar con sus “iguales”. El conjunto de protocolos que utiliza una determinada arquitectura en todas sus capas se denomina *pila de protocolos* (“protocol stack”); así es frecuente oír hablar de la pila de protocolos OSI, SNA, TCP/IP o DECNET, por ejemplo. Siempre entre capas homónimas, se establece un protocolo punto a punto. Esto permite que los problemas sean menores, ya que se debe resolver esa capa suponiendo que el resto está correcto.

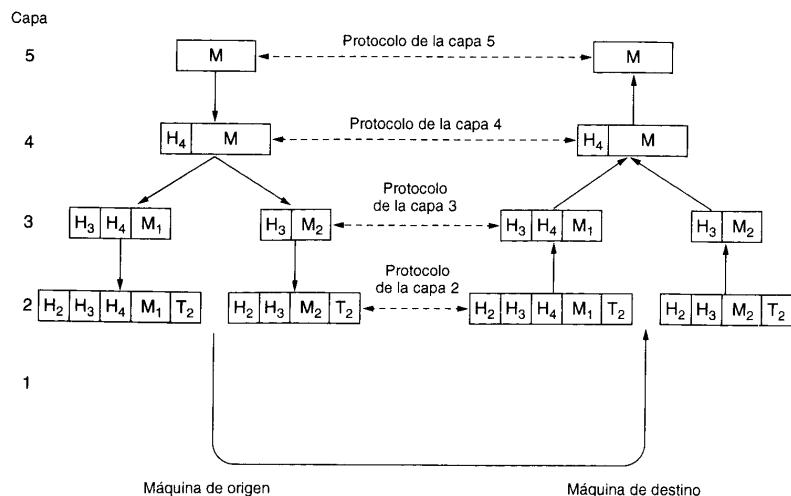


Para comprender mejor cómo funciona el modelo de arquitectura de redes basado en capas hagamos una analogía. Supongamos que un ejecutivo de la empresa A desea enviar de forma urgente un importante informe a un colega suyo en la empresa B. Para esto hablará con aquél notificándole el envío y a continuación pasará a su secretaria el informe con las instrucciones correspondientes. La secretaria llamará a la secretaria de B para averiguar la dirección exacta, pondrá el informe en un sobre y llamará a un servicio de mensajería, que enviará a un motorista para que recoja el paquete y lo lleve al aeropuerto. Cuando el paquete llega al aeropuerto de

destino es recogido allí por otro motorista que lo lleva a la oficina de la empresa B y lo entrega a la secretaria; ésta se ocupará de los trámites administrativos (pagar al mensajero, abrir el paquete, comprobar su contenido, acusar recibo a la secretaria de A, etc.) y lo pasará después a su jefe, el cual una vez estudiado el informe llamará al ejecutivo de A.

Obsérvese que en el proceso anterior existen diferentes niveles claramente diferenciados: los ejecutivos, las secretarias, los motoristas, y por último la empresa de líneas aéreas que se ocupa del transporte físico de la mercancía. En todos los niveles (menos probablemente el más bajo) hay dos entidades, la transmisora (A) y la receptora (B). Si todo ocurre según lo previsto cada entidad sólo hablará con su correspondiente en el otro lado, y con sus entidades vecinas, es decir, el jefe de A sólo habla con el jefe de B y con su secretaria, la secretaria habla con su jefe, con el motorista y con la otra secretaria para confirmar el envío, etc. En ningún caso se contempla que la secretaria de A hable con el ejecutivo de B. Si por ejemplo la secretaria de A es sustituida por enfermedad por otra persona los procedimientos seguirán funcionando, siempre y cuando la secretaria suplente desarrolle la misma función. Las variaciones de carácter interno sólo han de ser conocidas por las entidades contiguas, por ejemplo, el motorista de B podría ser reemplazado por una furgoneta de reparto, y este hecho solo ha de ser conocido por la secretaria de B y por la persona que entrega los paquetes en el aeropuerto. Esto es lo que denominamos una interfaz. Obsérvese que el modelo de capas simplifica considerablemente la tarea de cada una de las entidades, que sólo tiene que preocuparse de una pequeña parte de todo el mecanismo. En esencia se trata de aplicar a la resolución de problemas la vieja fórmula de divide y vencerás.

Cuando un sistema desea enviar un mensaje a un sistema remoto normalmente la información se genera en el nivel más alto; conforme va descendiendo se producen diversas transformaciones, por ejemplo adición de cabeceras, de colas, de información de control, la fragmentación en paquetes más pequeños si es muy grande (o más raramente la fusión con otros si es demasiado pequeño), etc. Todas estas operaciones se invierten en el sistema remoto en las capas correspondientes, llegando en cada caso a la capa correspondiente en el destino un mensaje igual al original.

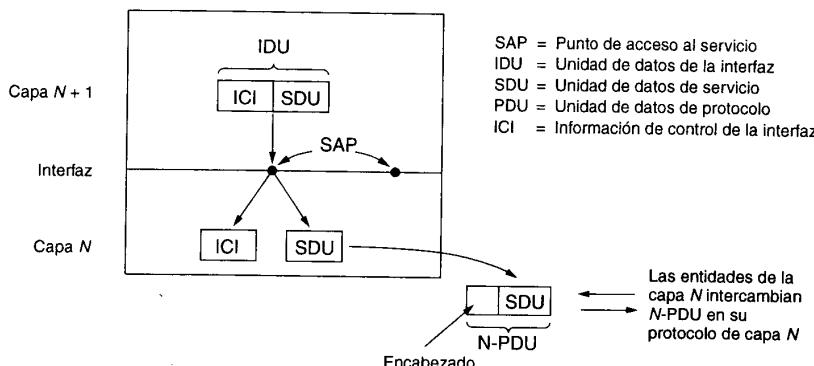


### 1.2.2 Interfaces y servicios

Se llama **Entidad** a los elementos activos en cada capa. Una entidad puede ser un proceso, un componente hardware, o una combinación de ambos. Un computador puede tener una o varias entidades en cada capa (por ejemplo un computador con dos tarjetas de conexión a LAN). **Entidades iguales** o **entidades pares** ("peer entities") son dos entidades diferentes que pertenecen a la misma capa; generalmente estarán en diferentes máquinas, pero podrían estar en la misma.

Las entidades de la capa  $n$  implementan los servicios que utiliza la capa  $n+1$ . En este caso la capa  $n$  actúa como el *proveedor del servicio* y la capa  $n+1$  es el *usuario del servicio*. El uso que la capa  $n$  haga de los servicios de la capa  $n-1$  es algo que no afecta ni cumple a la capa  $n+1$ .

Los servicios están disponibles en los SAPs (Service Access Points). Los SAPs de la capa  $n$  son los puntos donde la capa  $n+1$  puede acceder a los servicios ofrecidos. Cada SAP de cada entidad de la capa  $n$  tiene una dirección que le identifica de forma única en toda la red.



**Interfaz** es el conjunto de reglas que gobiernan el intercambio de información entre capas. En una comunicación la entidad de la capa  $n+1$  intercambia una *IDU* (Interface Data Unit) con la entidad de la capa  $n$  a través del SAP. La IDU esta formada por una *SDU* (Service Data Unit) e información de control. La *SDU* es la información que se transmite a la entidad equivalente (*peer*) en el lado contrario, y de allí a la capa  $n+1$  a través de su SAP. La información de control es necesaria como su nombre indica para que la capa  $n$  haga correctamente su trabajo, pero no es parte de los datos mismos. En la especificación de una arquitectura solo es necesario describir la estructura de la *SDU*, pero no la de la *IDU*; ésta se describe en la interfaz, que puede ser distinta para cada implementación.

Para transferir la *SDU* (Service Data Unit) la entidad de la capa  $n$  puede tener que fragmentarla en varias PDUs (Protocol Data Units). Cada PDU llevará una cabecera que permitirá a la entidad de la capa  $n$  en el otro lado ensamblar de nuevo la *SDU* correctamente.

### 1.2.3 Servicios orientados y no orientados a conexión

En una arquitectura de redes cada capa utiliza los servicios de la capa inmediatamente inferior para comunicar con la correspondiente del otro extremo. En función de como se establezca esa comunicación suelen distinguirse dos tipos de servicios: orientados a conexión y no orientados a conexión.

En el **servicio orientado a conexión**, también llamado CONS (Connection Oriented Network Service), primero se establece el canal de comunicación, después se transmiten los datos, y por último se termina la conexión. Dicha "conexión" se denomina *circuito virtual* (VC, virtual circuit). Una vez establecido el VC el camino físico que van a seguir los datos está determinado; los paquetes deben ir todos por él desde el origen al destino, y llegar en el mismo orden con el que han salido. Dado que el VC establece de forma clara el destino, los paquetes no necesitan contener su dirección. Generalmente se distinguen dos tipos de circuitos virtuales: *comutados*, también llamados SVCs (Switched Virtual Circuits), y *permanentes*, conocidos también como PVCs (Permanent Virtual Circuits). Los SVCs se establecen y terminan a petición del usuario, normalmente cuando hay paquetes que se quieren transmitir. Los PVCs están establecidos (conectados) todo el tiempo que la red está operativa.

En el servicio **no orientado a conexión**, llamado también CLNS (ConnectionLess Network Service) la comunicación se establece de manera menos formal. Cuando una entidad tiene información que transmitir sencillamente la envía en forma de paquetes, confiando que estos llegaran a su destino mas pronto o mas tarde. No se establece previamente un VC ni otro tipo de canal de comunicación extremo a extremo; los paquetes pueden ir por caminos físicos diversos, y

deben incluir cada uno la dirección de destino. Los paquetes pueden ser almacenados por nodos intermedios de la red, y reenviados más tarde. Aunque lo normal es que lleguen en el mismo orden con que han salido, esto no está garantizado como ocurría en el servicio orientado a conexión debido al almacenamiento en nodos intermedios y a la diversidad de caminos físicos posibles. A los paquetes enviados en un servicio no orientado a conexión se les denomina *datagramas*, ya que cada paquete viaja hacia su destino de forma completamente independiente de los demás como si fuera un telegrama.

Generalmente se suelen explicar los modelos orientado y no orientado a conexión con dos analogías: el sistema telefónico y el sistema de correo postal. El sistema telefónico es un ejemplo de servicio orientado a conexión, mientras que el sistema postal es un servicio no orientado a conexión. La analogía es bastante exacta salvo por el hecho de que en redes telemáticas la diferencia en el tiempo de entrega del mensaje entre servicios CONS y CLNS no es tan grande como la anterior comparación podría hacer pensar.

### 1.3 MODELOS DE REFERENCIA

Los modelos de referencia especifican la arquitectura del sistema de transmisión de datos a través de la red. Debido a la gran complejidad de este sistema, los diseñadores de redes usan el concepto de caja negra para esconder a otros diseñadores aquellos aspectos que no son relevantes para resolver una determinada tarea. Este es un concepto que puede ser usado tanto en el desarrollo de software como de hardware y ayuda a diseñar sistemas complejos de forma jerárquica.

En redes, el concepto de caja negra está relacionado con el de capa (layer). El sistema de transmisión se descompone en una serie de capas que usan las capas de nivel inferior para realizar la tarea asignada, y proporcionan a la capa superior una visión más simplificada del problema de la transmisión de datos.

#### 1.3.1 El modelo de referencia OSI

El estándar OSI (Open Systems Interconnection) es un modelo de referencia desarrollado por la ISO (International Standards Organization) como una generalización del modelo TCP/IP que ya había sido implementado satisfactoriamente unos años antes. Tardo 7 años en desarrollarse y se finalizó en 1983. Los protocolos propuestos por la OSI para la ISO son los que empiezan por X. (X.25, X.400, X.500, etc.). Definen como debe producirse la comunicación entre dos dispositivos cualesquier conectados a una red de transmisión de datos.

Con el objetivo de simplificar el diseño, OSI divide el modelo de comunicación en 7 capas. El estándar OSI solo especifica el objetivo de cada capa, no indica cómo debe conseguirse cada objetivo ni cómo son los protocolos que se usan en cada capa.

- **La Capa Física (physical layer)**

Se encarga fundamentalmente de la transmisión del flujo de bits. En concreto realiza la codificación del canal y/o modulación que fueran necesarias, la sincronización de los relojes del emisor y el receptor.

Esta capa transmite los bits entre dos entidades (nodos) directamente conectadas. Puede tratarse de un enlace punto a punto o de una conexión multipunto (una red broadcast, por ejemplo Ethernet). La comunicación puede ser dúplex, semi-dúplex o simplex. Si la información se transmite por señales eléctricas se especifican los voltajes permitidos y su significado (1 ó 0) y análogamente para el caso de fibra óptica. Se especifican las características mecánicas del conector, la señalización básica, etc.

- **La capa de enlace (data link layer)**

La principal función de la capa de enlace es ofrecer un servicio de comunicación fiable a partir de los servicios que recibe de la capa física, también entre dos entidades contiguas de la red. Esto supone que se realice detección y posiblemente corrección de errores. A diferencia de la capa física, que transmitía los bits de manera continua, la capa de enlace transmite los bits en grupos denominados *tramas (frames)* cuyo tamaño es típicamente de unos pocos cientos a unos pocos miles de bytes. En caso de que una trama no haya sido transmitida correctamente se deberá

enviar de nuevo; también debe haber mecanismos para reconocer cuando una trama se recibe duplicada. Generalmente se utiliza algún mecanismo de control de flujo, para evitar que un transmisor rápido pueda “colapsar” a un receptor lento.

Las redes broadcast utilizan funciones especiales de la capa de enlace para controlar el acceso al medio de transmisión, ya que éste es compartido por todos los nodos de la red. Esto añade una complejidad a la capa de enlace que no está presente en las redes basadas en líneas punto a punto, razón por la cual en las redes broadcast la capa de enlace se subdivide en dos subcapas: la inferior, denominada subcapa MAC (Media Access Control) se ocupa de resolver el problema de acceso al medio, y la superior, subcapa LLC (Logical Link Control) cumple una función equivalente a la capa de enlace en las líneas punto a punto.

Ejemplos de protocolos de la capa de enlace son el ISO 7776, la capa de enlace de X.25 (de la ITU) o el ISO HDLC. Como ejemplos de protocolos de la subcapa MAC podemos citar los IEEE 802.3 (Ethernet), IEEE 802.5 (Token Ring) o el ISO 9314 (FDDI). El protocolo de subcapa LLC de todas las redes locales broadcast es el IEEE 802.2.

- **La capa de red (network layer)**

Entra en funcionamiento cuando el nodo origen y destino pertenecen a redes físicas diferentes. En concreto, se encarga del encaminamiento de los datos entre redes.

Esta es la capa que tiene “conciencia” de la topología de la red, y se ocupa de decidir por qué ruta va a ser enviada la información; la decisión de la ruta a seguir puede hacerse de forma estática, o de forma dinámica en base a información obtenida de otros nodos sobre el estado de la red.

De forma análoga a la capa de enlace la capa de red maneja los bits en grupos discretos que aquí reciben el nombre de *paquetes*; motivo por el cual a veces se la llama la capa de paquete. Los paquetes tienen tamaños variables, pudiendo llegar a ser muy elevados, sobre todo en protocolos recientes, para poder aprovechar eficientemente la elevada velocidad de los nuevos medios de transmisión (fibra óptica, ATM, etc.). Por ejemplo en TCP/IP el tamaño máximo de paquete es de 64 KBytes, pero en el nuevo estándar, llamado IPv6, el tamaño máximo puede llegar a ser de 4 GBytes (4.294.967.296 Bytes).

Algunos ejemplos de protocolos utilizados en la capa de red son los protocolos de nivel de paquete y nivel de pasarela CCITT X.25 y X.75, el IP (Internet Protocol), CCITT/ITU-T Q.931, Q.933, Q.2931, y el OSI CLNP (ConnectionLess Network Protocol).

En las redes de tipo broadcast el nivel de red es casi inexistente, ya que desde un punto de vista topológico podemos considerar que en una red broadcast los nodos están interconectados todos con todos, por lo que no se toman decisiones de encaminamiento.

- **La capa de transporte (transport layer)**

La capa de transporte es la primera que se ocupa de comunicar directamente nodos terminales, utilizando la subred como un medio de transporte transparente gracias a los servicios obtenidos de la capa de red. Por esta razón se la ha llamado históricamente la capa host-host. También se suele decir que es la primera capa extremo a extremo.

La principal función de la capa de transporte es fragmentar de forma adecuada los datos recibidos de la capa superior (sesión) para transferirlos a la capa de red, y asegurar que los fragmentos llegan y son recomuestos correctamente en su destino.

En condiciones normales la capa de transporte solicita a la capa de red una conexión diferente por cada solicitud recibida de la capa de sesión, pero puede haber razones de costo que aconsejen multiplexar diferentes conexiones en la capa de sesión sobre una sola conexión en la capa de red.

o, inversamente, razones de rendimiento pueden requerir que una conexión solicitada por la capa de sesión sea atendida por varias conexiones en la capa de red; en ambos casos la capa de transporte se ocupará de hacer la multiplexación más adecuada de forma transparente a la capa de sesión.

La capa de transporte establece el tipo de servicio que recibe la capa de sesión, y en último extremo los usuarios. Éste podría ser por ejemplo un servicio libre de errores que entrega los mensajes en el mismo orden en que se envían; también podría ser un servicio de datagramas, es decir, mensajes independientes sin garantía en cuanto al orden de entrega ni confirmación de la misma, o un servicio broadcast o multicast en que los paquetes se distribuyen a múltiples destinos simultáneamente.

El control de flujo, que ha aparecido en capas anteriores, es necesario también en la capa de transporte para asegurar que un host rápido no sature a uno lento. La capa de transporte realiza también su propio control de errores, que resulta ahora esencial pues algunos protocolos modernos como Frame Relay o ATM han reducido o suprimido totalmente el control de errores de las capas inferiores, ya que con las mejoras en la tecnología de transmisión de datos éstos son menos frecuentes y se considera más adecuado realizar esta tarea en el nivel de transporte. Salvo el caso de transmisiones multicast o broadcast el nivel de transporte se ocupa siempre de una comunicación entre dos entidades, lo cual le asemeja en cierto sentido al nivel de enlace. Por esto existen grandes similitudes entre ambas capas en cuestiones tales como el control de errores o control de flujo.

Ejemplos de protocolos de transporte incluyen el CCITT X.224, también llamado protocolo de transporte OSI TP4 (Transport Protocol 4). En Internet existen dos protocolos de transporte: TCP y UDP.

- **La capa de sesión (session layer)**

La capa de sesión es la primera que es accesible al usuario, y es su interfaz más básica con la red. Por ejemplo, mediante los servicios de la capa de sesión un usuario podría establecer una conexión como terminal remoto de otro computador. En un sistema multiusuario la capa de sesión se ocupa de ofrecer un SAP a cada usuario para acceder al nivel de transporte.

- **La capa de presentación (presentation layer)**

Se preocupa de la forma en que los datos estructurados (números enteros, números en punto flotante, fechas, tablas de conversión de caracteres – por ejemplo, ASCII/Unicode –, etc.) sean traducidos para que puedan ser comprendidos entre aquellas estaciones con diferentes representaciones y/o longitudes para dichos tipos de datos. También se encarga de cifrar/descifrar (validación de passwords) y de comprimir/descomprimir los datos si esto es necesario.

- **La capa de aplicación (application layer)**

La capa de aplicación comprende los servicios que el usuario final está acostumbrado a utilizar en una red telemática, por lo que a menudo los protocolos de la capa de aplicación se denominan *servicios*. Dado que se crean continuamente nuevos servicios, existen muchos protocolos para la capa de aplicación, uno o más por cada tipo de servicio.

Ejemplos de protocolos estándar de la capa de aplicación son el X.400 o X.500 de la ITU, los protocolos SMTP, FTP y HTTP de Internet, etc.

### **1.3.2 El modelo de referencia TCP/IP**

En 1969 la agencia ARPA (Advanced Research Projects Agency) del Departamento de Defensa (DoD, Department of Defense) de los Estados Unidos inició un proyecto de interconexión de computadores mediante redes telefónicas. Al ser un proyecto desarrollado por militares en plena

guerra fría un principio básico de diseño era que la red debía poder resistir la destrucción de parte de su infraestructura (por ejemplo a causa de un ataque nuclear), de forma que dos nodos cualesquiera pudieran seguir comunicados siempre que hubiera alguna ruta que los uniera. Esto se consiguió en 1972 creando una red de conmutación de paquetes denominada ARPAnet, la primera de este tipo que operó en el mundo. La conmutación de paquetes unida al uso de topologías malladas mediante múltiples líneas punto a punto dio como resultado una red altamente fiable y robusta.

La ARPAnet fue creciendo paulatinamente, y pronto se hicieron experimentos utilizando otros medios de transmisión de datos, en particular enlaces por radio y vía satélite; los protocolos existentes tuvieron problemas para interoperar con estas redes, por lo que se diseñó un nuevo conjunto o pila de protocolos, y con ellos una arquitectura. Este nuevo conjunto se denominó TCP/IP (Transmission Control Protocol/Internet Protocol) nombre que provenía de los dos protocolos más importantes que componían la pila; la nueva arquitectura se llamó sencillamente *modelo TCP/IP*, los nuevos protocolos fueron especificados por vez primera por Cerf y Kahn en un artículo publicado en 1974. A la nueva red, que se creó como consecuencia de la fusión de ARPAnet con las redes basadas en otras tecnologías de transmisión, se la denominó Internet.

La aproximación adoptada por los diseñadores del TCP/IP fue mucho más pragmática que la de los autores del modelo OSI. Mientras que en el caso de OSI se emplearon varios años en definir con sumo cuidado una arquitectura de capas donde la función y servicios de cada una estaban perfectamente definidas, y solo después se planteó desarrollar los protocolos para cada una de ellas, en el caso de TCP/IP la operación fue a la inversa; primero se especificaron los protocolos, y luego se definió el modelo como una simple descripción de los protocolos ya existentes. Por este motivo el modelo TCP/IP es mucho más simple que el OSI. También por este motivo el modelo OSI se utiliza a menudo para describir otras arquitecturas, como por ejemplo la TCP/IP, mientras que el modelo TCP/IP nunca suele emplearse para describir otras arquitecturas que no sean la suya propia. En el modelo TCP/IP se pueden distinguir cuatro capas.

- **La capa host-red**

Esta capa engloba realmente las funciones de la capa física y la capa de enlace del modelo OSI. El modelo TCP/IP no dice gran cosa respecto a ella, salvo que debe ser capaz de conectar el host a la red por medio de algún protocolo que permita enviar paquetes IP. Podríamos decir que para el modelo TCP/IP esta capa se comporta como una “caja negra”. Cuando surge una nueva tecnología de red (por ejemplo ATM) una de las primeras cosas que aparece es un estándar que especifica de que forma se pueden enviar sobre ella paquetes IP; a partir de ahí la capa internet ya puede utilizar esa tecnología de manera transparente.

- **La capa internet**

Esta capa es el “corazón” de la red. Su papel equivale al desempeñado por la capa de red en el modelo OSI, es decir, se ocupa de encaminar los paquetes de la forma más conveniente para que lleguen a su destino, y de evitar que se produzcan situaciones de congestión en los nodos intermedios. Debido a los requisitos de robustez impuestos en el diseño, la capa internet da únicamente un servicio de conmutación de paquetes no orientado a conexión. Los paquetes pueden llegar desordenados a su destino, en cuyo caso es responsabilidad de las capas superiores en el nodo receptor la reordenación para que sean presentados al usuario de forma adecuada.

A diferencia de lo que ocurre en el modelo OSI, donde los protocolos para nada intervienen en la descripción del modelo, la capa internet define aquí un formato de paquete y un protocolo, llamado IP (Internet Protocol), que se considera el protocolo “oficial” de la arquitectura.

- **La capa de transporte**

Esta capa recibe el mismo nombre y desarrolla la misma función que la cuarta capa del modelo OSI, consistente en permitir la comunicación extremo a extremo (host a host) en la red. Aquí se definen dos protocolos:

El TCP (Transmission Control Protocol) ofrece un servicio CONS fiable, con lo que los paquetes (aquí llamados segmentos) llegan ordenados y sin errores. TCP se ocupa también del control de flujo extremo a extremo, para evitar que por ejemplo un host rápido sature a un receptor más lento. Ejemplos de protocolos de aplicación que utilizan TCP son el SMTP (Simple Mail Transfer Program, correo electrónico) y el FTP (File Transfer Protocol).

El otro protocolo de transporte es UDP (User Datagram Protocol) que da un servicio CLNS, no fiable. UDP no realiza control de errores ni de flujo. Una aplicación típica donde se utiliza UDP es la transmisión de voz y vídeo en tiempo real; aquí el retardo que introduciría el control de errores produciría más daño que beneficio: es preferible perder algún paquete que retransmitirlo fuera de tiempo. Otro ejemplo de aplicación que utiliza UDP es el NFS (Network File System); aquí el control de errores y de flujo se realiza en la capa de aplicación.

- **La capa de aplicación**

Esta capa desarrolla las funciones de las capas de sesión, presentación y aplicación del modelo OSI. La experiencia ha demostrado que las capas de sesión y presentación son de poca utilidad, debido a su escaso contenido, por lo que la aproximación adoptada por el modelo TCP/IP parece mas acertada.

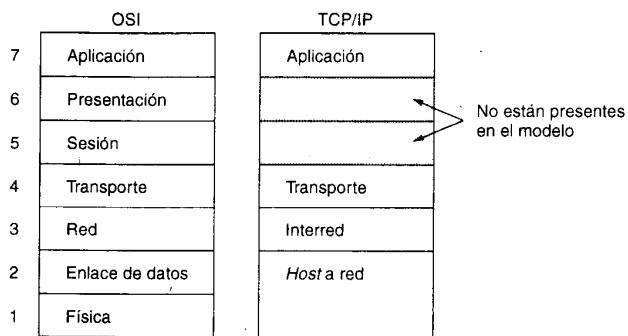
La capa de aplicación contiene todos los protocolos de alto nivel que se utilizan para ofrecer servicios a los usuarios. Entre estos podemos mencionar tanto los "tradicionales", que existen desde que se creó el TCP/IP: terminal virtual (TelNet), transferencia de ficheros (FTP), correo electrónico (SMTP) y servidor de nombres (DNS), como los mas recientes, como el servicio de news (NNTP), el Web (HTTP), el Gopher, etc.

#### **1.3.3 Comparación de los modelos OSI y TCP/IP**

Como ya hemos comentado, la génesis del modelo OSI y TCP/IP fue muy diferente. En el caso de OSI primero fue el modelo y después los protocolos, mientras que en TCP/IP el orden fue inverso. Como consecuencia de esto el modelo OSI es mas elegante y esta menos condicionado por ningún protocolo en particular, y se utiliza profusamente como modelo de referencia para explicar todo tipo de redes. El modelo OSI hace una distinción muy clara entre servicios, interfaces y protocolos, conceptos que a menudo se confunden en el modelo TCP/IP. Podríamos decir que la arquitectura (o el modelo) OSI es mas modular y académico que el TCP/IP.

Pero este mayor nivel de abstracción también tiene sus inconvenientes. Los diseñadores del modelo OSI no tenían experiencia práctica aplicando su modelo para desarrollar protocolos y olvidaron algunas funcionalidades importantes. Por ejemplo, las redes broadcast no fueron previstas inicialmente en la capa de enlace, por lo que se tuvo que insertar a la fuerza la subcapa MAC para incluirlas. Otro problema era que no se había previsto la interconexión de redes diferentes, cosa que fue como ya hemos visto la clave del modelo TCP/IP.

El modelo OSI tiene siete capas, mientras que el modelo TCP/IP sólo tiene cuatro. Aunque es desafortunada la fusión de la capa física y la de enlace en una oscura capa host-red, la fusión de las capas de sesión, presentación y aplicación en una sola en el modelo TCP/IP es claramente mas lógica que la del modelo OSI.



En la práctica los protocolos basados en las normas estándar OSI definidas por la ISO nunca llegaron a tener gran relevancia a nivel mundial, a pesar de que la mayoría de los grandes fabricantes de computadores y compañías telefónicas impulsaron su utilización ofreciendo productos y servicios basados en ellos. Las razones principales que motivaron este fenómeno las podemos resumir en los siguientes puntos:

- Momento inadecuado: Para cuando estaban disponibles productos comerciales basados en protocolos OSI (finales de los ochenta) ya estaban ampliamente difundidos los productos basados en los protocolos TCP/IP; esto era especialmente cierto en entornos académicos (universidades y centros de investigación), que aunque económicamente no eran los mejor dotados sí tenían las mayores redes a nivel mundial.
- Tecnología inapropiada: como ya hemos comentado la elección del modelo de siete capas para el protocolo OSI era algo forzada. Una de las razones que llevaron a elegir este número de capas era que coincidía con el del modelo SNA de IBM, que dominaba el mercado de la informática por aquel entonces; los autores del modelo OSI creían que aproximándose a SNA tenían mayores posibilidades de éxito. La complejidad de la arquitectura OSI (análogamente a la SNA) es considerable, y en muchos aspectos difícil de traducir en programas.
- Implementaciones inadecuadas: en parte como consecuencia de su complejidad, los productos comerciales que aparecían basados en los protocolos OSI eran muy caros y poco fiables. Esto creó un círculo vicioso, ya que al ser caros los usuarios no los compraban, y al no usarse en condiciones reales los nuevos productos no se depuraban; además, las empresas fabricantes tenían que mantener un alto precio del software OSI para compensar los elevados costos de desarrollo y mantenimiento. Como contraste una de las primeras implementaciones de TCP/IP formaba parte del UNIX de Berkeley, era muy buena y además se distribuía gratuitamente. No es extraño pues que rápidamente se asociara OSI con baja calidad, complejidad y costos elevados.
- Mala política: el desarrollo de OSI era patrocinado principalmente por la ISO, la Comunidad Europea y los gobiernos de sus países miembros; las decisiones eran fruto de multitud de reuniones de los diversos comités y grupos de trabajo, y en ocasiones se tomaban en consideración no sólo aspectos técnicos sino también políticos, buscando el compromiso entre sus miembros. Por el contrario el desarrollo de TCP/IP seguía un curso mucho más improvisado e informal, cualquier persona podía (y puede) proponer un nuevo protocolo para su estandarización independientemente de su nacionalidad, prestigio o situación laboral. Haciendo una simplificación podríamos decir que OSI funcionaba como una “democracia parlamentaria” (similar a un gobierno moderno), mientras que TCP/IP era más similar a una ONG, o a un movimiento alternativo.

Aunque por la exposición anterior pueda parecer lo contrario, también existen aspectos negativos en los protocolos TCP/IP:

- Por un lado no se distinguen claramente los conceptos de servicio, interfaz y protocolo.
- El “modelo” TCP/IP fue diseñado con posterioridad al protocolo, intentando imitar la labor de síntesis que se había hecho en el modelo OSI (podríamos decir que es como si se hubieran cortado los patrones después de cosido el traje).

- La “caja negra” que hemos llamado capa host-red y que en el modelo TCP/IP es mas bien una interfaz que una capa, ya que lo único que se especifica de ella es que ha de ser capaz de transmitir paquetes IP. Como consecuencia de esto el modelo TCP/IP no distingue entre la capa física y la de enlace, ya que ambas entran en la “capa” host-red.

Durante la década de los ochenta en Europa las redes académicas de la mayoría de los países (incluido España) utilizaban protocolos OSI por imposición de los respectivos gobiernos y de la Comunidad Europea; a la vista de los problemas ya mencionados de los productos OSI, y la extensión y buen resultado de los protocolos TCP/IP, se empezaron a ofrecer en 1991 servicios basados en TCP/IP, lo cual provocó su inmediata difusión por toda Europa y el estancamiento y casi desaparición de los servicios basados en protocolos OSI.

Consecuentemente con los puntos fuertes y débiles de cada modelo y protocolo, en el curso nos basaremos en una versión modificada del modelo OSI, del cual hemos suprimido la capa de sesión y la de presentación. Sin embargo utilizaremos este modelo para describir fundamentalmente protocolos TCP/IP, si bien también hablaremos de otros mas modernos y que en muchos casos se utilizan como medio de transporte para TCP/IP. En la tabla siguiente hacemos un resumen del modelo y los protocolos más comunes de cada capa.

Capa	Protocolo
Aplicación	TCP/IP (DNS, SMTP, SNMP, NNTP, HTTP)
Transporte	TCP/IP (TCP, UDP) ATM (AAL1, AAL2, AAL3/4, AAL5)
Red	TCP/IP (IP, ICMP, ARP, RARP, OSPF, BGP, IPv6), ATM (Q2931)
Enlace	ISO (HDLC), TCP/IP (SLIP, PPP), ATM, LANs
Física	N-ISDN, B-ISDN (ATM), GSM, SONET/SDH, LANs Cable coaxial, cable UTP, fibra óptica, microondas, radioenlaces, satélite

## 1.4 ESTANDARIZACION DE REDES

Existen muchos vendedores y proveedores de red, cada uno con sus propias ideas de cómo deben hacerse las cosas. Sin coordinación, habría un caos total, y los usuarios no tendrían nada que hacer. La única salida es ponerse de acuerdo sobre algunas normas de las redes. Los estándares permiten que diferentes computadoras puedan comunicarse y también aumentan el mercado de los productos que se adhieren a estos.

### 1.4.1 Quién es quién en el mundo de las telecomunicaciones

La situación legal de las compañías telefónicas del mundo varía considerablemente de un país a otro. En un extremo está Estados Unidos con sus 2000 empresas telefónicas privadas y en el otro extremo están los países en los cuales el gobierno respectivo tiene el monopolio de todas las comunicaciones, como correos, telégrafos, teléfonos y a veces la radio y la televisión. En algunos casos la autoridad de la telecomunicación es una compañía nacionalizada y en otros es simplemente una rama del gobierno PTT (administración de Correos, Telégrafos y Teléfonos). La tendencia a nivel mundial es hacia una liberación y competencia, y alejarse del monopolio gubernamental. La mayoría de los países europeos tiene privatizadas (parcialmente) sus PTTs, pero en otras partes el proceso avanza con lentitud.

Con tantos proveedores diferentes de servicios, es claro que se necesita una compatibilidad a escala mundial para asegurarse de que las personas (y las computadoras) de un país puedan llamar a sus contrapartes en otro. En 1865, los representantes de muchos gobiernos de Europa se reunieron para formar el predecesor de la actual ITU (Unión Internacional de Telecomunicaciones). Su trabajo era estandarizar las telecomunicaciones internacionales. En 1947 la ITU se convirtió en una agencia de las Naciones Unidas. La ITU tiene tres sectores principales: Radiocomunicaciones (ITU-R), Estandarización de telecomunicaciones (ITU-T) y Desarrollo (ITU-D).

De 1956 a 1993, la ITU-T se conocía como CCITT (del francés Comité Consultatif International

Télégraphique et Téléphonique, Comité Consultivo Internacional para la Telegrafía y Telefonía).

La ITU-T tiene alrededor de 200 miembros gubernamentales, 500 miembros de sector, incluyendo compañías telefónicas (por ejemplo, AT&T, Vodafone, WorldCom), fabricantes de equipos de telecomunicación (como Cisco, Nokia, Nortel), fabricantes de computadoras (como Compaq, Sun, Toshiba), fabricantes de chips (como Intel, Motorola, TI), compañía de medios (como AOL Time Warner, CBS, Sony) y otras empresas interesadas (como Boeing, Samsung, Xerox). todos Unidos.

La tarea de la ITU-T es hacer recomendaciones técnicas sobre telefonía, telegrafía y las interfaces de comunicación de datos. Estas recomendaciones suelen convertirse en estándares reconocidos internacionalmente, por ejemplo el V.24 (EIA RS-232), el cual especifica la ubicación y significado de los diversos pines en el conector utilizado para la mayoría de las terminales asíncronas y módems externos y el estándar V.90 para módems de 56 kbps.

#### **1.4.2 Quién es quién en los estándares internacionales**

Los estándares internacionales son producidos y publicados por la ISO (Organización de Estándares Internacionales), fundada en 1946. Sus miembros son ANSI (Estados Unidos), BSI (Gran Bretaña), AFNOR (Francia), DIN (Alemania) y otras 85 organizaciones de estándares nacionales de los países miembro. La ISO emite estándares sobre una gran cantidad de temas. Se han emitido más de 13.000 estándares, entre ellos los estándares de OSI que se refiere a las redes de computadoras. La ISO tiene casi 200 comités técnicos, numerados por el orden de su creación, refiriéndose cada uno a un objeto específico. El TC97 trata con computadoras y procesamiento de información. Cada TC tiene subcomités (SCs) divididos en grupos de trabajo (WGs). En cuanto a estándares de telecomunicación, la ISO y la ITU-T suelen cooperar (la ISO es miembro de la ITU-T).

El NIST (Instituto Nacional de Estándares y Tecnología) es parte del Departamento de Comercio de Estados Unidos. Emite estándares que son obligatorios para compras hechas por el gobierno de Estados Unidos.

Otro representante importante en el mundo de los estándares es el IEEE (Instituto de Ingenieros Eléctricos y Electrónicos), la mayor organización de profesionales del mundo. Además de publicar multitud de periódicos y organizar cientos de conferencias cada año, el IEEE tiene un grupo de estandarización que desarrolla estándares en el área de ingeniería eléctrica y computación. El comité 802 del IEEE ha estandarizado muchos tipos de LANs.

#### **1.4.3 Quién es quién en el mundo de los estándares de Internet**

Cuando se creó ARPANET, el DoD creó un comité informal para supervisarla. En 1983 se dio otro nombre al comité: IAB (Consejo de Actividades de Internet) y se le encomendó una misión un poco más amplia, que era la de mantener a los investigadores de ARPANET y de Internet apuntando más o menos en la misma dirección. El significado del acrónimo "IAB" se cambió a Consejo para la Arquitectura de Internet.

Cuando se necesitaba un estándar (por ejemplo, un nuevo algoritmo de enrutamiento), los miembros del IAB le daban solución y después anuncianan los cambios para que los estudiantes que estuvieran a cargo de la implementación del software pudieran realizarlos. La comunicación se llevaba a cabo mediante una serie de informes técnicos denominados RFCs (Solicitudes de Comentarios). Estos informes se almacenan en línea en [www.ietf.org/rfc](http://www.ietf.org/rfc). Los RFCs se encuentran organizados por el orden cronológico de su creación. Actualmente existen alrededor de 3000.

Para 1989 Internet había crecido tanto que este estilo sumamente informal dejó de ser funcional. Muchos fabricantes ofrecían productos de TCP/IP en ese entonces y no deseaban cambiarlos tan sólo porque algunos investigadores habían concebido una mejor idea. El IAB fue reorganizado en el verano de 1989. Los investigadores fueron transferidos a la IRTF (Fuerza de Trabajo para la Investigación sobre Internet), que fue puesta bajo el mando del IAB, junto con la IETF (Fuerza de

Trabajo para la Ingeniería de Internet). Más tarde se creó la Sociedad de Internet, integrada por gente interesada en Internet. Esta sociedad se asemeja al ACM o al IEEE, es dirigida por administradores electos que designan a los miembros del IAB. El propósito de esta división era que la IRTF se concentrara en proyectos de investigación a largo plazo, en tanto que la IETF se encargaba de proyectos de ingeniería a corto plazo.

Además, se adoptó un proceso de estandarización más formal, con base en la ISO. Para convertirse en Estándar Propuesto, la idea fundamental debía explicarse completamente en un RFC y despertar suficiente interés en la comunidad. Para avanzar a la etapa de Estándar Borrador, una implementación funcional debía haber sido rigurosamente probada por al menos dos sitios independientes durante al menos cuatro meses. Si el IAB se convence de que la idea suena lógica y el software funciona, declara que el RFC es un Estándar de Internet.

Para los estándares Web , el Consorcio World Wide Web (W3C ) desarrolla protocolos y directrices para facilitar el crecimiento a largo plazo de la Web. Se trata de un consorcio industrial liderado por Tim Berners-Lee y creada en 1994 cuando la Web realmente comenzó a despegar. W3C tiene actualmente más de 300 miembros de todo el mundo y ha producido más de 100 recomendaciones W3C, que abarca temas como HTML y privacidad Web.

## Tema II

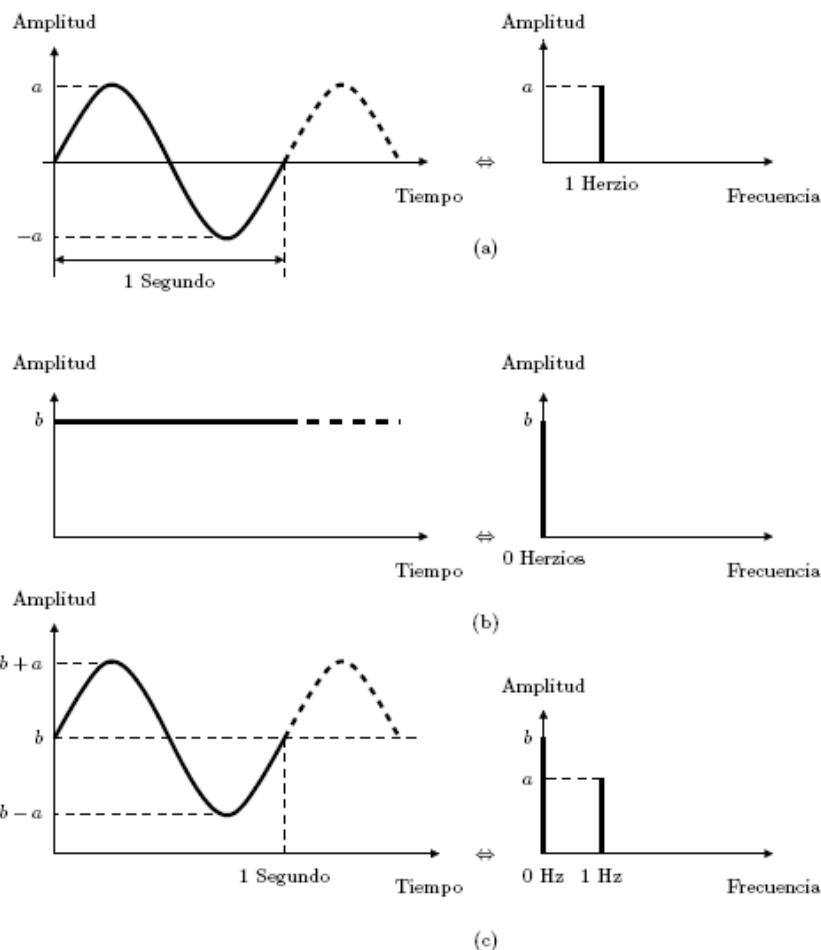
### Nivel Físico

La primera capa dentro de cualquier modelo de red está formada por el medio físico. Aquí más que en ninguna otra capa es necesaria la compatibilidad entre los equipos para que sea posible la comunicación.

#### 2.1. SEÑALES

Las señales (independientemente de si son digitales o no) tienen dos dominios básicos de representación: el del tiempo y el de la frecuencia. En el dominio del tiempo se expresa el valor instantáneo (el voltaje por ejemplo) de la señal en función del tiempo. En el dominio de la frecuencia se expresa la aportación relativa de cada componente de frecuencia durante cierto intervalo de tiempo.

Suponiendo, por ejemplo, que se pretende describir una señal sinusoidal. En la Figura (a), a la izquierda (dominio del tiempo) se ha representado una señal sinusoidal con un periodo de 1 sg., o lo que es lo mismo, con una frecuencia de 1 Hz. Puesto que tiene una amplitud  $a$  (una amplitud pico a pico de  $2a$ ), esta señal está perfectamente descrita por la gráfica de la derecha (dominio de la frecuencia o espectro de la señal), donde se muestra un pulso infinitamente estrecho situado en la frecuencia 1 Hz. y con una altura  $a$ . Es claro que cualquier señal sinusoidal tiene una única componente de frecuencia.



En la Figura (b) se muestra una señal sinusoidal de frecuencia 0 y amplitud  $b > a$ , y en la Figura (c) se muestra una señal que es la suma de las dos señales anteriormente descritas, tanto en el dominio del tiempo como en su correspondiente representación en el dominio de la frecuencia. A principios del siglo XIX, el matemático Jean-Baptiste Fourier se dio cuenta de esta equivalencia y demostró que cualquier señal periódica (que se repite en el tiempo de

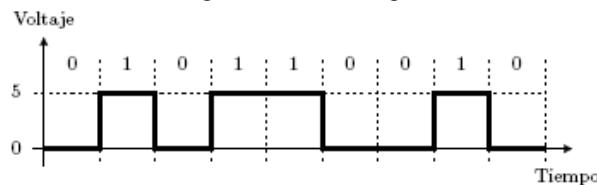
forma infinita), por muy compleja que sea, puede ser expresada como una suma (en principio infinita) de señales sinusoidales (senos y cosenos).

$$g(t) = \frac{1}{2}c + \sum_{n=1}^{\infty} a_n \sin(2\pi n f t) + \sum_{n=1}^{\infty} b_n \cos(2\pi n f t)$$

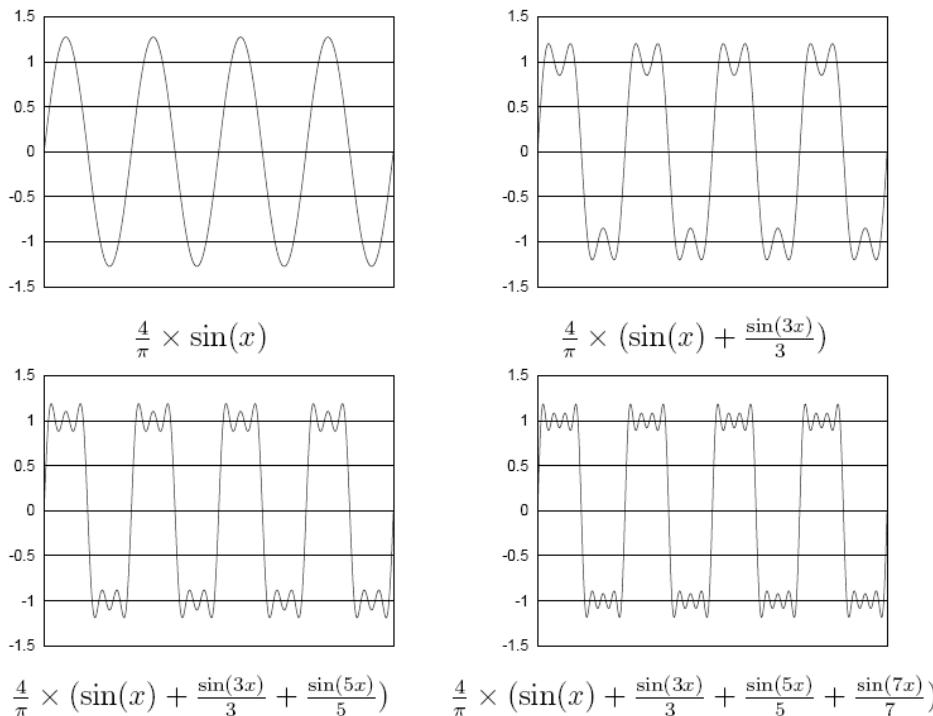
donde  $f = 1/T$  es la frecuencia fundamental,  $a_n$  y  $b_n$  son las amplitudes de seno y coseno de los  $n$ -ésimos (términos) armónicos y  $c$  es una constante. Tal descomposición se conoce como serie de Fourier.

Una señal digital se caracteriza por tener al menos dos estados estables de energía (voltaje, intensidad de corriente o luz, etc.). En uno de esos estados puede permanecer un intervalo de tiempo que tiene un tamaño mínimo pero no un tamaño máximo (pensar en una secuencia infinita de ceros o unos).

Una señal digital invierte (al menos teóricamente) un tiempo infinitamente pequeño en pasar de uno de los estados estables al otro. En su forma más simple, una señal digital solo tiene dos estados lógicos (representados generalmente por los símbolos 0 y 1). Supongamos que la señal digital se representa físicamente mediante un nivel de tensión. Supongamos que uno de los estados lógicos (el 0 por ejemplo) se representa mediante un potencial de 0 V. y que el otro (el 1 por ejemplo) mediante 5 V. Bajo estas condiciones, la secuencia de bits 010110010 se representará físicamente mediante la siguiente señal digital.



Sin pérdida de generalidad para obtener resultados supongamos que la señal digital que queremos transmitir es una secuencia infinita de pulsos de igual duración (010101...). Esta señal digital puede generarse a partir de señales sinusoidales, tal y como Fourier demostró. A la de más baja frecuencia se le llama frecuencia fundamental o simplemente fundamental. Al resto de componentes de frecuencia se las conoce como armónicos.



## 2.2. CAPACIDAD DE UN CANAL.

Como se vio, se puede representar cualquier señal de datos con una *serie de Fourier*. La serie consiste en términos de frecuencias distintas, y se suman los términos para reconstruir la señal.

Ningún medio de transmisión puede transmitir señales sin perder algún poder. Normalmente un medio puede transmitir las frecuencias desde 0 hasta algún límite  $f$ ; las frecuencias mayores se atenúan fuertemente.

La capacidad de un canal es la cantidad de datos por unidad de tiempo que es capaz de transmitir. La capacidad de un canal depende de su ancho de banda, del tipo de modulación utilizada y de la tasa de errores provocados por el ruido del canal. Se entiende por ancho de banda de un canal de transmisión como la anchura de la banda de frecuencias que es capaz de transmitir.

### 2.2.1. Teorema de Nyquist.

Cualquier canal de transmisión tiene un ancho de banda limitado.

Canal de transmisión	Ancho de banda (KHz)
Línea telefónica	3,1
Emisión de radio de onda media (AM)	4,5
Emisión de radio de FM	75
Emisión de televisión PAL	8 000
Red local Ethernet 10 Mbps	10 000
Emisión de televisión de alta definición	30 000

Los bits se transmiten por un canal realizando modificaciones en la onda portadora; por ejemplo en una línea telefónica se podría utilizar una frecuencia de 1 KHz para representar el 0 y una de 2 KHz para el 1, esto se conoce como modulación de frecuencia; si se sincroniza dos equipos para que puedan cambiar la frecuencia de la portadora cada 3,333 milisegundos se podría transmitir datos a 300 bits por segundo, (si dos bits consecutivos son iguales en realidad no hay tal cambio); se dice entonces que se transmite 300 **símbolos** por segundo, o simplemente 300 **baudios**. Si en vez de dos frecuencias se utiliza cuatro, por ejemplo 0,5, 1, 1,5 y 2 KHz, se puede transmitir dos bits por símbolo, al disponer de cuatro estados o niveles posibles; así manteniendo el caudal de 300 símbolos por segundo se transmite 600 bits por segundo; análogamente si se utiliza ocho estados se puede transmitir 900 bits por segundo (tres bits por símbolo), y así sucesivamente; se gana en velocidad, pero a cambio se tiene que ser más precisos en la frecuencia ya que aumenta el número de valores posibles. Además de la frecuencia es posible modular la amplitud y la fase de la onda portadora; en la práctica los módems modernos modulan una compleja combinación de amplitud y fase para extraer el máximo provecho posible de las líneas telefónicas, es decir el máximo número de símbolos por segundo y el máximo número de bits por símbolo.

A pesar de la mejora en eficiencia conseguida con la sofisticación técnica los canales de transmisión tienen un límite. En 1924 Nyquist observó la existencia de un límite fundamental en las transmisiones digitales sobre canales analógicos, que se conoce como **teorema de Nyquist**, y que establece que el número máximo de baudios que puede transmitirse por un canal no puede ser superior al doble de su ancho de banda. Así en el caso de la transmisión de datos por una línea telefónica, con un ancho de banda de 3,1 KHz, el máximo número de baudios que puede transmitirse es de 6.200.

Por ejemplo, si se codifica una información representando un bit por símbolo; eligiendo un valor de amplitud de +1 V para representar el 1 y -1 V para el 0. La secuencia de bits a transmitir, que en principio es aleatoria, puede fluctuar entre dos situaciones extremas: transmitir siempre el mismo valor (11111... ó 00000...) o transmitir una secuencia alterna (010101...); la primera posibilidad genera una corriente continua de frecuencia 0 hertzios, mientras que la segunda produce una onda cuadrada de frecuencia igual a la mitad del número de bits transmitidos (ya que una onda completa estaría formada por dos bits, una cresta y un valle); la gama de frecuencias va pues de cero a la mitad del número de bits, con lo que la anchura de banda es

igual a la mitad del número de bits transmitidos. Se podría repetir el mismo razonamiento para el caso en que se transmita más de un bit por símbolo, es decir que haya más de dos posibles voltajes y se vería como el ancho de banda correspondería a la mitad del número de símbolos por segundo.

El teorema de Nyquist no establece el número de bits por símbolo, que depende del número de estados que se utilicen. Se puede expresar el teorema de Nyquist en forma de ecuación relacionándolo con el caudal máximo de información transmitida: si  $H$  es el ancho de banda y  $V$  el número de niveles o estados posibles, entonces el caudal máximo en bits por segundo  $C$  viene dado por:

$$C = 2H \log_2 V$$

Recordando que:  $\log xy = \frac{\log y}{\log x}$

Por ejemplo, un canal telefónico ( $H=3,1$  KHz) con tres bits por baudio (ocho estados,  $V=8$ ) se tiene un caudal máximo  $C = 2 * 3,1 * (\log(8)/\log(2)) = 18,6$  Kbps

### 2.2.2. Teorema de Shannon – Hartley

El Teorema de Nyquist fija un máximo en el número de símbolos por segundo, pero dado que no dice nada respecto al número de bits por símbolo la capacidad del canal en bits por segundo podría ser arbitrariamente grande utilizando una modulación capaz de transmitir un número lo bastante grande de bits por símbolo.

Sin embargo, a medida que aumenta el número de bits por símbolo se incrementa el número de estados diferentes que el receptor ha de poder discernir, y se reduce la distancia entre éstos. En canales muy ruidosos puede llegar a ser difícil distinguir dos estados muy próximos. Como cabría esperar, el número máximo de estados que el receptor pueda distinguir depende de la calidad del canal de transmisión, es decir de su relación señal/ruido. En 1948 Shannon dedujo una expresión que cuantificaba la capacidad máxima de un canal analógico en función de su ancho de banda y su relación señal/ruido.

El valor de la relación señal/ruido se suele indicar en decibelios (dB), que equivalen a  $10 \log_{10} S/N$  (así 10 dB equivalen a una relación S/R de 10, 20 dB a una relación de 100 y 30 dB a una de 1000). Dado que la percepción de la intensidad del sonido por el oído humano sigue una escala logarítmica la medida en decibelios da una idea más exacta de la impresión que producirá un nivel de ruido determinado. En 1948 Shannon y Hartley generalizaron el teorema de Nyquist al caso de un canal de comunicación con ruido aleatorio, derivando lo que se conoce como la **ley de Shannon-Hartley**, que está expresada en la siguiente ecuación:

$$C = H \log_2 (1 + S/N)$$

$$\frac{S}{N} = 10^{\frac{SRN}{10}}$$

De nuevo aquí  $H$  representa el ancho de banda y  $C$  el caudal de transmisión de la información. Por ejemplo, con un ancho de banda de 3,1 KHz y una relación señal-ruido (SRN) de 36 dB se tiene  $C = 3,1 * (\log(1 + 10^{(36/10)}) / \log(2)) = 37,07$  Kb/s; 36 dB equivale a una relación señal/ruido de 4000 y es el valor máximo que puede obtenerse en una comunicación telefónica, ya que esta es la cantidad de ruido que introduce el proceso de digitalización de un canal telefónico que se utiliza actualmente en la mayoría de las redes telefónicas del mundo. Si la relación señal-ruido desciende a 20 dB (cosa bastante normal) la velocidad máxima baja a 20,6 Kb/s.

### 2.2.3 Latencia

La latencia de un enlace de transmisión  $t$  viene determinada únicamente por el retardo de propagación de las señales a través del enlace. Dicho tiempo depende de la distancia a recorrer  $l$  (longitud del enlace) y de la velocidad de propagación de la señal  $v$  según la relación

$$t = \frac{l}{v}$$

La velocidad de propagación de las señales a través de los enlaces de transmisión es finita y depende de las características físicas de la señal y del medio. La velocidad de una señal eléctrica en un conductor de cobre es de  $2,3 \times 10^8$  metros/segundo. La velocidad de una señal

luminosa en una fibra óptica es de  $2,0 \times 10^8$  metros/segundo y la velocidad de una señal electromagnética en el vacío es de  $3,0 \times 10^8$  metros/segundo.

### **2.3. TÉCNICAS DE TRANSMISIÓN.**

La técnica de Transmisión es la forma en que los datos se transportan a través del medio físico. Existen dos técnicas, la transmisión en **banda base** y la transmisión en **banda ancha**.

#### **2.3.1. Transmisión en Banda Base.**

Las redes que transmiten en banda base utilizan tecnología digital, donde se hace uso de todo el ancho de banda que posee el canal para transmitir los datos digitales que proporcionan las estaciones. Debido a que los usuarios comparten el mismo canal físico para transmitir, es necesario que la transmisión se efectúe a muy alta velocidad con el fin de que la duración de las transmisiones sea pequeña. Esta es la técnica habitual en redes de área local, siendo sus principales ventajas la sencillez de diseño, y el bajo coste frente a las redes de banda ancha. Por otra parte su principal inconveniente es la limitación en distancia debido al deterioro o distorsión que sufren las señales.

La **multiplexación por división de tiempo** (TDM: *time division multiplexing*) sirve para compartir la capacidad disponible de un canal de transmisión de banda base. Se usan dos tipos de TDM:

- Síncrona (o de ciclo fijo): Cada usuario tiene acceso al canal en intervalos de tiempo definidos de manera precisa (sincronizados).
- Asíncrona (o por demanda): Cada usuario tiene acceso aleatorio al canal y, al obtener el acceso, se convierte en el único usuario del canal mientras dure su transmisión.

#### **2.3.2. Transmisión en Banda Ancha.**

Las redes que transmiten en banda ancha utilizan señales portadoras analógicas que transmiten las señales digitales de las estaciones utilizando como adaptadores los correspondientes modems. El medio de transmisión utilizado normalmente es el cable coaxial que permite efectuar transmisiones que va de 300 a 400 MHz. Esta gama de frecuencias se divide en una serie de canales mediante la técnica denominada multiplexación por división de frecuencia (**FDM: frequency division multiplexing**).

La FDM permite la transmisión independiente de hasta 120 señales que pueden utilizar distintas velocidades de transmisión de bits modulados, distintos métodos de acceso y puede ser señales de distinta naturaleza (video, voz, fax, etc.). Se puede considerar que cada canal es por si solo una red de área local independiente. Este tipo de transmisión tiene la ventaja de permitir mayores distancias y la integración de señales de todo tipo, mientras que sus desventajas son su elevado coste, la planificación de la red y su implementación física.

#### **2.3.3. Modulación y codificación**

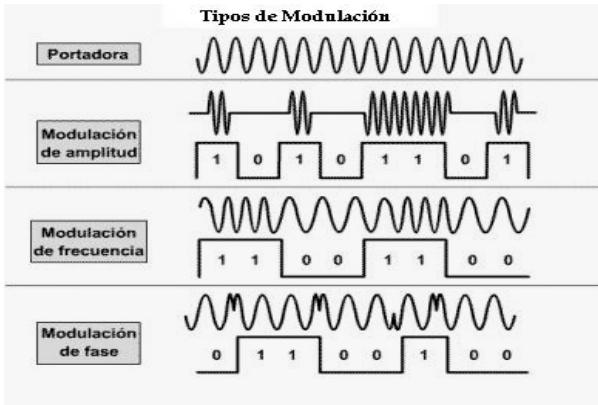
Cuando se envían datos por un canal de transmisión analógico (por ejemplo una línea telefónica) es preciso *modular* la señal en origen y *demodularla* en el destino; el aparato que realiza esta función se llama **módem**. Inversamente, cuando enviamos una señal analógica por un canal de transmisión digital tenemos que *codificarla* en origen y *decodificarla* en destino, para lo cual se utiliza un aparato denominado **códec**.

Por ejemplo un sistema de videoconferencia es un códec, puesto que convierte una señal analógica (la imagen en movimiento captada por la cámara) en una señal digital (la secuencia de bits transmitida por algún medio); también hay un códec presente en cualquier sistema de grabación digital de sonido. Es frecuente referirse a los códecs como conversores analógico-digital o conversores A/D, aunque en telecomunicaciones suele preferirse la denominación códec.

La modulación es el envío de una señal, que toma el nombre de moduladora, a través de otra señal denominada portadora, de características óptimas para la transmisión a distancia. La señal moduladora generalmente controla algún parámetro de la señal portadora, de tal forma que ambas pueden unirse y separarse en los momentos que corresponda.

- **Modulación de Señales con Portadora Analógica.**

Este tipo de modulación utiliza señal portadora sinusoidal de alta o muy alta frecuencia, estando alguno de sus parámetros controlados por la señal a transmitir. Existen tres tipos básicos y combinaciones de ellos:

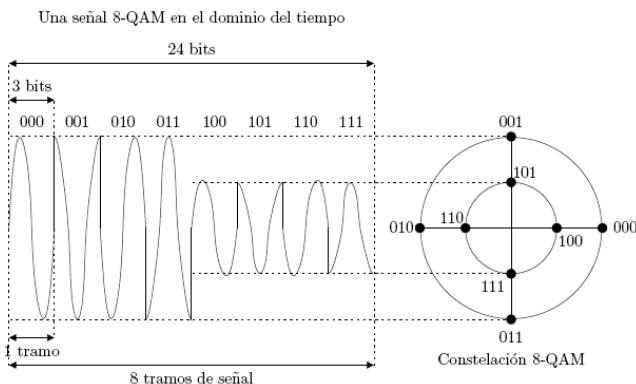


**Modulación en amplitud (AM) - Amplitude Shift Keying (ASK)** Es el caso en que la amplitud de la portadora está controlada por la señal moduladora. Se utiliza fundamentalmente para transmisiones telefónicas, de radio y televisión. El gran inconveniente de este tipo de modulación está en la poca protección que presenta al ruido.

**Modulación en frecuencia (FM) - Frequency Shift Keying (FSK)** Este tipo de modulación se realiza variando la frecuencia de la portadora en función de la amplitud de la moduladora. Ocupa mayor ancho de banda pero goza de una mejor resistencia al ruido. Se utiliza fundamentalmente en transmisiones de radio de alta calidad.

**Modulación en fase (PM) - Phase Shift Keying (PSK)** Es un tipo de modulación utilizada fundamentalmente para moduladora digital, y consiste en controlar la fase de la señal portadora a través de la señal moduladora. Goza de gran inmunidad al ruido.

**Modulación de amplitud en cuadratura - 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 por cada símbolo. Por ejemplo, podemos utilizar cuatro amplitudes diferentes y cuatro fases diferentes para conseguir  $4 \times 4 = 16$  símbolos diferentes (QAM-16).



- **Modulación de Señales con Portadora Digital.**

Estos tipos de modulación se caracterizan por transmitir un tren de impulsos. Según se controle uno u otro parámetro, de los impulsos, aparecen los distintos tipos de modulación. Toda transmisión con portadora digital posee características opuestas para la multiplexación de señales en el tiempo. Los tipos de modulación con portadora digital son:

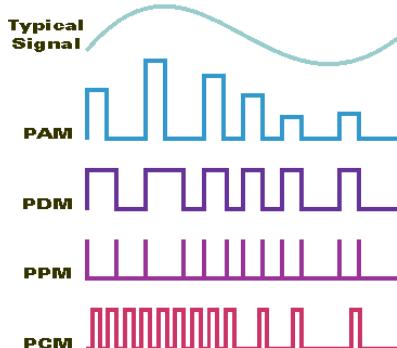
**Impulsos modulados en amplitud (PAM):** Este tipo de modulación tiene su origen en el concepto de muestreo de una señal. En él, cada cierto tiempo se genera un impulso cuya amplitud es proporcional a la amplitud de la señal moduladora.

**Impulsos modulados en posición (PPM):** Este tipo de modulación trata de ahorrar potencia en la transmisión de un tren de impulsos haciendo que todos ellos sean de la misma amplitud. La posición de cada impulso con respecto a un origen contiene la información que se transmite.

**Impulsos modulados en duración (PDM):** Este tipo contiene la información a transmitir por medio de la duración de cada impulso.

**Modulación por impulsos codificados (MIC):** Consiste en la relación existente entre la señal moduladora y grupos de impulsos de igual amplitud, duración y posición en el tiempo.

#### Modulation

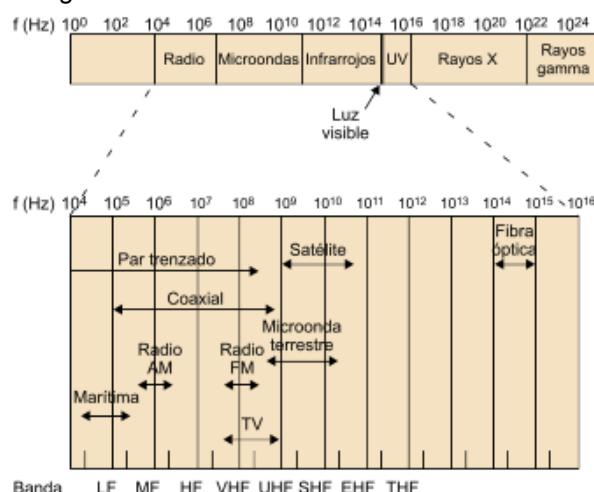


## 2.4. MEDIOS DE TRANSMISIÓN.

Una de las formas más comunes de transmitir una señal es mediante ondas electromagnéticas. Las dos principales ventajas del uso de ondas electromagnéticas como medio de transmisión son:

- Su velocidad de propagación es muy alta (300.000 km/s en el vacío).
- Se transmiten prácticamente en cualquier tipo de medio.

Las ondas electromagnéticas se clasifican en función de su frecuencia

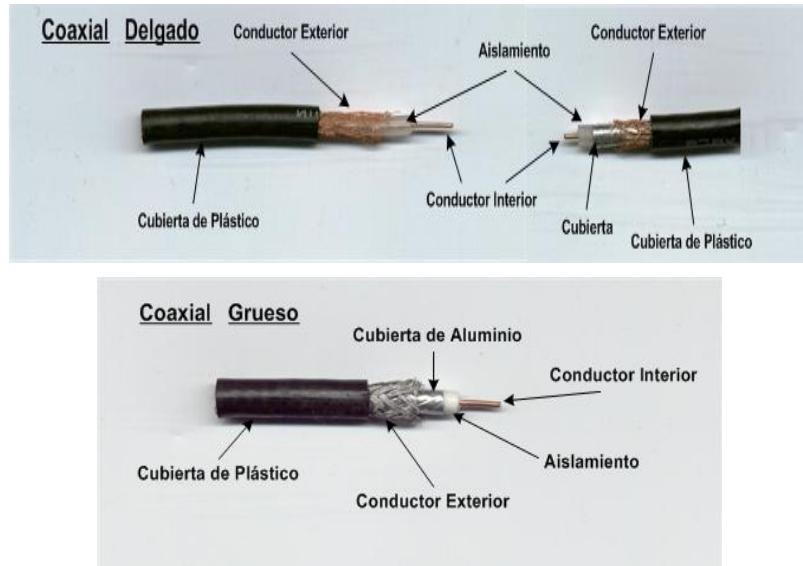


Los medios de transmisión se clasifican en guiados (hacen uso de un medio físico para la propagación de la señal) y no guiados (utilizan el aire o el vacío como medio de propagación).

### 2.4.1. Medios de Transmisión Guiados

- **Cable Coaxial.**

Un cable coaxial consta de un par de conductores de cobre o aluminio, formando uno de ellos un alma central, rodeado y aislado del otro mediante pequeños hilos trenzados o una lámina metálica cilíndrica. La separación y aislamiento entre los dos conductores se realiza generalmente con anillos aislantes (teflón o plástico), espaciados regularmente a una cierta distancia, o un revestimiento completo del material mencionado.



Los puertos para coaxial delgado suelen estar compuestos de un único conector hembra de tipo BNC. Lo único necesario es cortar el cable a la medida necesaria, instalar dos conectores BNC macho en los extremos y conectarlos a la tarjeta de red del ordenador mediante un derivador conocido como "T". La impedancia del cable es de 50 ohmios.



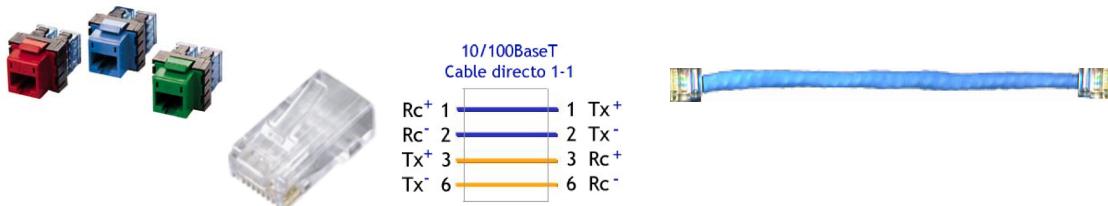
Sus inconvenientes principales son: atenuación, ruido térmico y ruido de intermodulación.

- **Cable de Par Trenzado.**

Cada circuito de transmisión lo configura un par de hilos de cobre aislados por medio de un material plástico, trenzados o torsionados entre sí con el fin de disminuir posibles interferencias. Dependiendo del número de trenzas por unidad de longitud y de la calidad del aislante exterior, los pares trenzados se clasifican en categorías que van desde la 1 a la 7 siendo esta última la de mayor calidad. En la actualidad, son los más utilizados para redes de área local.

Las conexiones en este cable se establecen mediante conectores del tipo RJ45, son muy parecidos a los empleados en telefonía, que son los RJ11, pero más anchos y con 8 pinos o puntos de conexión. En las redes de par trenzado lo más usual es que cada ramal termine en una roseta RJ45 hembra y que la tarjeta también sea una hembra. Para establecer la conexión a la red no hay más que tender un cable entre ellas, provisto en sus extremos por dos conectores machos de RJ45. Este cable es conocido como latiguillo (Patch Cord).

Categoría	Ancho de banda (MHz)	Aplicaciones
Categoría 1	0,4 MHz	Líneas telefónicas y módem de banda ancha.
Categoría 2	4 MHz	Transmite datos hasta 4 Mbps. Se usó en las redes ARCnet (arco de red) y Token Ring (configuración de anillo).
Categoría 3	16 MHz	Velocidad de hasta 10 Mbps.
Categoría 4	20 MHz	Velocidad de 16 Mbps. Se uso en redes Token Ring.
Categoría 5 (5e)	100 MHz	Velocidades de hasta 1000 Mbps. Se usa para redes ATM, 1000BASE T, 10BASE T, 100BASE T y token ring.
Categoría 6 (6a)	250 MHz / 500 MHz	Velocidades de 1 GB a 10GB. Es adecuado para redes 1000BASE T, 10GBASE T.
Categoría 7 (7a)	600 MHz / 1200 MHz	Ethernet de cable de cobre 10G

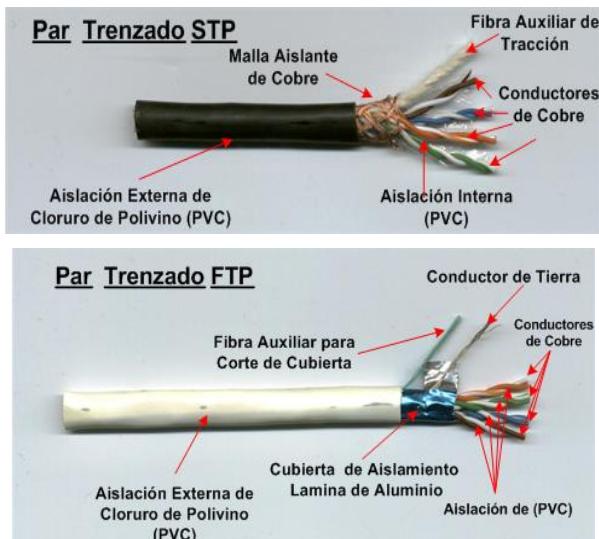


El cable de par trenzado empleado en redes de área local, puede ser de tres tipos según para el tipo de red a la que esté destinado, estos tipos de cable son:

Par trenzado de cuatro pares, sin blindaje (**UTP Unshielded Twisted Pair**) de 100 ohmios, 22/24 AWG, que suelen usarse en Ethernet de 10 Mbps, y en medio ambiente ausente de contaminación electromagnética, es difundido su empleo para realizar cableados Ethernet en el interior de oficinas y ambientes protegidos de los factores medioambientales, como las lluvias, exposición a rayos solares y humedad.



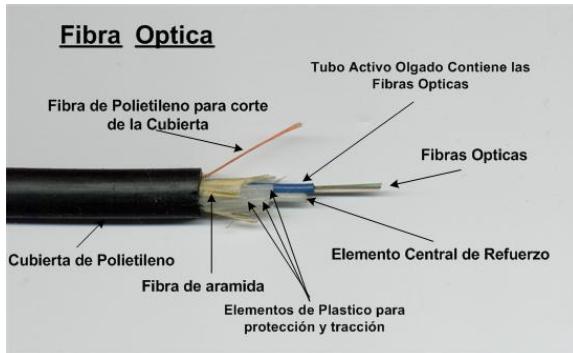
Par trenzado de cuatro pares con blindaje (**STP Shielded Twisted Pair**, o **FTP Foiled Twisted Pair**) de 150 ohmios, 22 AWG, para Fast Ethernet a 100 Mbps, o para Ethernet de 10 Mbps en medio ambiente sujeto a contaminación electromagnética cual es el caso de plantas industriales, existe en el mercado, cables con cubierta de PVC de alta densidad, que les permiten tener un alto grado de seguridad ante los factores medioambientales.



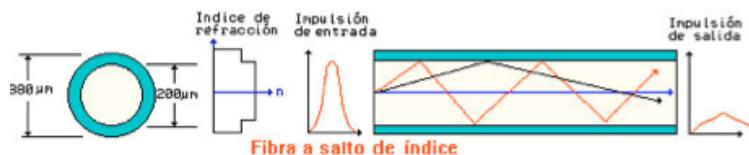
#### • **Cable de Fibra Óptica.**

Se trata de un medio, muy flexible y muy fino, que conduce energía de naturaleza lumínica (fotones) con muy bajas pérdidas. Su forma es cilíndrica con tres secciones radiales: núcleo, revestimiento y cubierta. El núcleo está formado por una o varias fibras muy finas de cristal o plástico. Cada fibra está rodeada por su propio revestimiento que es un cristal o plástico con diferentes propiedades ópticas distintas a las del núcleo. Todo este conjunto se recubre para proteger el contenido de aplastamientos, abrasiones, humedad, etc.

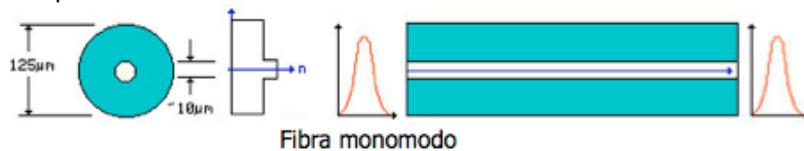
Es un medio muy apropiado para largas distancias e incluso para LAN's extendidas. Sus beneficios frente a cables coaxiales y pares trenzados son que permite mayor ancho de banda, menor tamaño y peso, menor atenuación, total aislamiento electromagnético, mayor separación entre repetidores.



Los rayos de luz que inciden en el núcleo de la fibra con ángulos diferentes, se van reflejando en la capa que lo recubre hasta llegar a su destino. A este tipo de propagación se le llama multimodal.

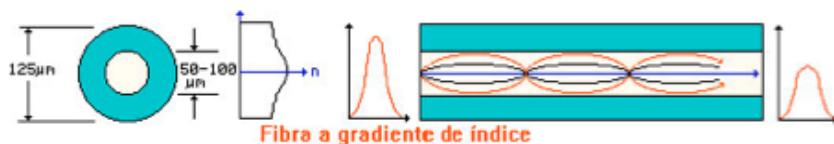


Si se reduce el radio del núcleo suficientemente, sólo es posible la transmisión de un rayo, el rayo axial, siendo monomodal, alcanzándose en este caso el mayor ancho de banda (2GHz). Las fibras monomodo disponibles actualmente pueden transmitir datos a 100 Gbps hasta 100 kilómetros sin amplificación.



El inconveniente del modo multimodal es que dependiendo al ángulo de incidencia de los rayos, estos tomarán caminos diferentes y tardarán más o menos tiempo en llegar al destino, con lo que se puede producir una distorsión en los pulsos (desfase entre los modos), con lo que se ve limitada la velocidad de transmisión.

Hay un tercer modo de transmisión que es un paso intermedio entre los anteriores y que consiste en cambiar el índice de refracción del núcleo. A este modo se le llama multimodo de índice gradual.



Los emisores de luz utilizados son: LED (de bajo coste, con utilización en un amplio rango de temperaturas y con larga vida media) y Láser (más caro, pero más eficaz y permite una mayor velocidad de transmisión). Para aumentar la capacidad de la fibra se utiliza la técnica D-WDM (Multiplexación por Longitud de Onda-Densa). Hay 2 tipos de conectores de Fibra Óptica, 568SC y 568ST. La posición correspondiente a los dos conectores del 568SC en su adaptador, se denominan como A y B. Esto ayuda a mantener la polaridad correcta en el sistema de cableado y permite al adaptador implementar polaridad inversa acertada de pares entre los conectores, lo que no ocurre en los conectores ST.



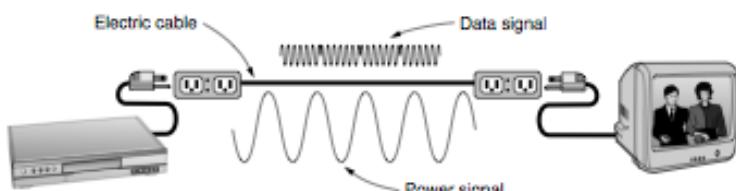
Comparación de los diferentes medios de transmisión			
Tipo de cable/característica	Par trenzado	Coaxial	Fibra óptica
Ancho de banda	Moderado	Grande	Muy grande
Longitud	Pequeña	Moderada	Muy alta
Fiabilidad de la transferencia	Moderada	Alta	Muy alta
Seguridad	Baja	Moderada	Alta
Complejidad de la instalación	Sencillo	Moderado	Complejo
Coste	Bajo	Moderado	Alto

- **Líneas de energía eléctrica – Power Lines**

Las redes telefónicas y de televisión por cable no son las únicas fuentes de cableado que pueden ser reutilizados para la comunicación de datos. Hay una especie aún más común de cableado que son las líneas de energía eléctrica . Las líneas eléctricas entregan energía eléctrica a las casas y el cableado eléctrico dentro de las casas distribuye la energía a las tomas eléctricas.

El uso de las líneas de energía para la comunicación de datos es una idea vieja. Las líneas eléctricas han sido utilizados por las compañías eléctricas durante muchos años, para la comunicación de baja velocidad como medición remota, también en el hogar para controlar dispositivos (por ejemplo, el estándar X10). En los últimos años ha habido interés en la comunicación de alta velocidad a través de estas líneas, tanto dentro del hogar como en una LAN y fuera del hogar para el acceso a Internet de banda ancha.

Sólo se tiene que conectar un televisor en la pared, lo que se debe hacer de todos modos, ya que necesitan la energía eléctrica, y pueden enviar y recibir las películas a través del cableado eléctrico. La señal de datos se sobrepone a la señal de energía de baja frecuencia (en el cable activo o “caliente” ) ya que ambas señales utilizan el cableado al mismo tiempo .



La dificultad con el uso de corriente eléctrica doméstica para una red es que fue diseñado para distribuir señales de energía eléctrica y que es muy diferente a la distribución de señales de datos, por lo que el cableado del hogar hace un trabajo horrible. Las señales eléctricas son enviadas a 50-60 Hz y el cableado atenúa las frecuencias más altas (MHz) que son señales necesarias para la comunicación de datos de alta velocidad. Las propiedades eléctricas de los cables varían de una casa a otra y cambian según los aparatos se encienden y se apagan, lo que provoca que las señales de datos reboten en el cableado. Las corrientes transitorias cuando los aparatos se encienden y apagan, crean ruido eléctrico en un amplio rango de frecuencias. Y sin el cuidado de trenzar los cables, el cableado eléctrico actúa como una fina antena, que capta señales externas e irradia sus propias señales. Este comportamiento significa que para cumplir los requisitos normativos, la señal de datos debe excluir frecuencias licenciadas como las bandas de radioaficionados.

A pesar de estas dificultades, es práctico para enviar al menos 100 Mbps a través del cableado eléctrico del hogar usando esquemas de comunicación que resisten frecuencias deterioradas y ráfagas de errores. Muchos productos utilizan diferentes estándares propietarios para la creación de redes de la línea eléctrica, mientras las normas internacionales aun están bajo desarrollo.

#### 2.4.2. Medios de Transmision No guiados (Inalámbrica).

Las ondas infrarrojas, microondas y ondas hertzianas son las llamadas radiaciones electromagnéticas que se utilizan en el campo de las telecomunicaciones inalámbricas (sin cable). La ventaja de estas ondas es que no son visibles para el ojo humano, a pesar de que pueden servir para la comunicación de información.

En los últimos años, el mercado de las comunicaciones inalámbricas se ha popularizado debido a las ventajas de las redes sin hilos: movilidad, flexibilidad, facilidad de instalación, escalabilidad, dinamismo en los cambios de la topología, y la posibilidad de llegar donde no llega el cable. Como principales inconvenientes se puede destacar su elevado coste inicial y su seguridad. Dentro del mundo de las comunicaciones sin hilos, se puede distinguir dos grandes grupos:

- Sistemas de comunicación con telefonía móvil.
- Redes inalámbricas.

Actualmente, la gran mayoría de la población tiene un teléfono móvil, y su uso ha experimentado un crecimiento exponencial en todo el mundo durante los últimos años. Inicialmente, la telefonía móvil, que sólo servía para mantener conversaciones telefónicas, evolucionó con la posibilidad de realizar envíos de pequeños mensajes de texto (SMS). Más adelante se posibilitó el acceso a Internet, primero a través del protocolo WAP, después el aumento de las velocidades de acceso a Internet con GPRS y finalmente, con las tecnologías UMTS y HSPA, que permiten disponer de terminales móviles (teléfonos, smartphones, tabletas, computadores de bolsillo...) multimedia con múltiples servicios y aplicaciones con conexiones a altas velocidades.

Por otro lado, la evolución de las redes inalámbricas de área local WiFi, gracias a la popularización de los accesos domésticos a Internet, se ha extendido en multitud de hogares y empresas, así como la utilización del protocolo Bluetooth para interconectar diferentes accesorios y dispositivos en ordenadores, impresoras, ratones, teclados, etc.

#### **2.4.2.1. Sistemas de comunicación de telefonía móvil**

En 1985 surgió en Europa la primera generación (1G) de teléfonos móviles después de adaptar el AMPS (advanced mobile phone system) a los requisitos europeos denominados TACS (total access communications system). TACS engloba todas las tecnologías de comunicaciones móviles analógicas. Con ella se podía transmitir voz pero no datos. Actualmente, esta tecnología está obsoleta.

Debido a la sencillez y las limitaciones de la primera generación surgió el GSM (global system for mobile communications), que marcó el comienzo de la segunda generación (2G) de la telefonía móvil. Esta tecnología puede transmitir datos, además de las bien conocidas conversaciones de voz, a una velocidad de 9,6 kbit/s; la transmisión de datos se inició con el servicio de mensajería corta o mensajes SMS.

Después, surgió la tecnología WAP, que consistía en unas páginas web pensadas para verlas con las pantallas monocromáticas de los teléfonos móviles. Las primeras conexiones se hacían con llamadas al proveedor telefónico, que transmitía los datos como si fuera un módem tradicional a una velocidad de 9,6 kbits/s.

En el 2001 surgió la denominada segunda generación y media (2,5G). En la 2,5G están incluidas todas las tecnologías que permiten una mayor capacidad de transmisión de datos. Dentro de esta generación nació la tecnología GPRS (general packet radio service), que permitía acceder a Internet a través del protocolo TCP/IP. La velocidad de comunicación era de 54 kbits/s de bajada y 9,6 kbits/s de subida, y el servicio se pagaba por los datos descargados, no por el tiempo de conexión. Con la finalidad de facilitar las comunicaciones inalámbricas debido a su generalización en todo el mundo, la ITU (International Telecommunication Union) adoptó diferentes interfaces de acceso radioeléctrico llamadas IMT-2000 (international mobile telecommunications-2000) o comunicaciones móviles internacionales.

Después surgieron las tecnologías 3G, que se definen dentro del IMT-2000 (de la ITU) que marca el estándar para que todas las redes 3G sean compatibles unas con otras. El 3GPP (3rd generation partnership project) trabajó con el UMTS (universal mobile telecommunications system), una de las tecnologías que utilizan los móviles de tercera generación (3G). Esta tecnología permite descargar datos a una velocidad de hasta 2 Mbits/s, lo que fomenta la aparición de nuevas aplicaciones y servicios. La tecnología UTMS se puede utilizar para conexiones a Internet, correo electrónico, FTP (transferencia de archivos), telnet (terminal remoto), videoconferencias, comercio electrónico, etc.

La evolución de UMTS fue hacia las tecnologías HSDPA (high speed downlink packet access) (3,5G y 3,75G) con terminales HSUPA y a los teléfonos de cuarta generación (4G) para mejorar el rendimiento por el uso simultáneo de aplicaciones y aumentar la cobertura de tecnologías 3G.

La cuarta generación (4G) es la última tecnología móvil, con velocidades de transmisión de 50 Mbps de subida y 100 Mbps de bajada y con una meta de 1Gbps; y utiliza diferentes tecnologías (HSPA+ y LTE - Long Term Evolution).

#### 2.4.2.2. Redes inalámbricas

Las diferentes tecnologías sin hilos (Wireless) se suelen agrupar basándose en el radio de acción (alcance) de cada una de ellas:

- Redes personales sin hilos (WPAN, wireless personal area network). Este concepto se aplica cuando la distancia que se quiere cubrir es del orden de unos cuantos metros. La familia de estándares más representativos son el 802.15.1 (Bluetooth) y el 802.15.4 (Zigbee).
- Redes locales sin hilos (WLAN, wireless local area network). Permiten dar servicios a distancias del orden de un centenar de metros (un piso, una planta de un edificio, unas cuantas calles, etc.). El estándar más destacado en este campo es el 802.11 (WiFi).
- Redes metropolitanas sin hilos (WMAN, wireless metropolitan area network). Permiten dar servicios a distancias del orden de unos cuantos kilómetros (un barrio, una ciudad, ...). El estándar más destacado en este campo es el 802.16 (WiMAX).
- Redes de gran alcance sin hilos (WWAN, wireless wide area network). Tienen una cobertura más amplia. La familia de estándares más representativos es la de GSM, GPRS, UMTS, HSPA, LTE.

##### • Bluetooth

Bluetooth es la tecnología inalámbrica más popular. Se trata de un protocolo basado en el estándar de comunicaciones IEEE 802.15, pensado para la transmisión de datos y voz sin hilos entre dispositivos, mediante una radiofrecuencia. Al inicio del desarrollo de los productos Bluetooth de primera generación, se tuvo en cuenta lo siguiente:

- El sistema tenía que ser universal.
- El emisor había de consumir poca energía, ya que se usaba en equipos que funcionan con batería.
- La conexión debía permitir la transmisión de datos y voz (aplicaciones multimedia).
- Tenía que ser de bajo coste.

De acuerdo con estos requisitos, Ericsson desarrolló en 1994 una tecnología que utiliza un canal de comunicación con frecuencia de 2,4 GHz, un máximo de 720 Kb/s (1 Mbps de velocidad bruta), un rango óptimo de 10 metros (opcionalmente de 100 metros con repetidores). En 1999 se creó el SIG (grupo de interés especial) de Bluetooth, formado por las empresas Ericsson, Intel, IBM, Toshiba y Nokia. Este SIG trabaja para definir, desarrollar, promover y publicar el protocolo Bluetooth. Esta tecnología es propietaria, es decir, que sólo puede producirla quien tiene la patente. Por eso sólo puede introducir esta tecnología en sus productos quien pertenece al SIG de Bluetooth.

Estos dispositivos se clasifican como "Clase 1", "Clase 2" o "Clase 3" en referencia a su potencia de transmisión:

Clase	Potencia máxima permitida (mW)	Potencia máxima permitida (dBm)	Alcance (aproximado)
Clase 1	100 mW	20 dBm	~30 metros
Clase 2	2.5 mW	4 dBm	~10-5 metros
Clase 3	1 mW	0 dBm	~1 metro

En la mayoría de los casos, la cobertura efectiva de un dispositivo de clase 2 se extiende cuando se conecta a un transceptor de clase 1. Esto es así gracias a la mayor sensibilidad y potencia de transmisión del dispositivo de clase 1, es decir, la mayor potencia de transmisión del dispositivo de clase 1 permite que la señal llegue con energía suficiente hasta el de clase 2. Por otra parte la mayor sensibilidad del dispositivo de clase 1 permite recibir la señal del otro

pese a ser más débil. Los dispositivos con Bluetooth también pueden clasificarse según su ancho de banda:

Versión	Ancho de banda
Versión 1.2	1 Mbit/s
Versión 2.0 + EDR	3 Mbit/s
Versión 3.0 + HS	24 Mbit/s
Versión 4.0	24 Mbit/s

Bluetooth se utiliza en un gran número de productos tales como teléfonos, impresoras, módems y auriculares. Su uso es adecuado cuando puede haber dos o más dispositivos en un área reducida sin grandes necesidades de ancho de banda. Su uso más común está integrado en teléfonos y PDA, bien por medio de unos auriculares Bluetooth o en transferencia de archivos, además se puede realizar y confeccionar enlaces o vincular distintos dispositivos entre sí.

Bluetooth simplifica el descubrimiento y configuración de los dispositivos, ya que estos pueden indicar a otros los servicios que ofrecen, lo que permite establecer la conexión de forma rápida (sólo la conexión, no la velocidad de transmisión).



El protocolo Bluetooth establece tres niveles de seguridad:

- Nivel 1: no hay seguridad. El dispositivo funciona en modo promiscuo, permitiendo que cualquier otro equipo Bluetooth se conecte al dispositivo.
- Nivel 2: seguridad a nivel de servicio. Soporta autoidentificación, encriptación y autorización una vez se ha establecido el canal de comunicación.
- Nivel3: seguridad a nivel de enlace. Las medidas de seguridad se implantan antes de que el canal de comunicación se haya establecido. Proporciona encriptación y autoidentificación.

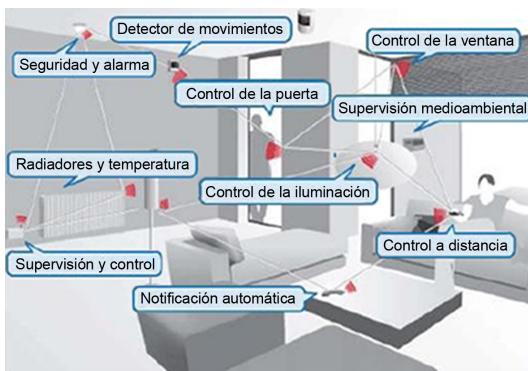
#### • ZigBee

ZigBee es la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica de bajo consumo, basado en el estándar WPAN IEEE 802.15.4. Su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de las baterías.

La especificación 1.0 de ZigBee se aprobó en el 2004 y está disponible en miembros del grupo de desarrollo (ZigBee Alliance). En diciembre del año 2006 se publicó la especificación actual.

ZigBee utiliza la banda de frecuencias ISM para usos industriales, científicos y médicos. En Europa utiliza la banda de 868 MHz, 915 MHz en Estados Unidos y 2,4 GHz en todo el mundo. El desarrollo de esta tecnología se basa en su sencillez y bajo coste. Un nodo Zig- Bee más completo requiere menos del 10% del hardware que necesita un nodo Bluetooth.

Los protocolos ZigBee están definidos para su uso en aplicaciones con requerimientos muy bajos de transmisión de datos y consumo energético, como control industrial, albergar sensores empotrados, recolectar datos médicos, detectar humo o intrusos y en domótica o teleasistencia. Una red en su conjunto utilizará una cantidad muy pequeña de energía, de forma que cada dispositivo individual pueda tener una autonomía de hasta cinco años antes de necesitar un recambio en su sistema de alimentación.



Las desventajas de ZigBee son: tasa de transferencia muy baja, sólo manipula textos pequeños en comparación con otras tecnologías, trabaja de manera que no es compatible con Bluetooth porque no llega a tener las mismas tasas de transferencia, ni la misma capacidad de soporte para nodos, tiene una menor cobertura porque pertenece a redes sin hilos del tipo WPAN.

Una red ZigBee puede constar de un máximo de 65.535 nodos distribuidos en subredes de 255 nodos (frente a los 8 nodos máximo de una subred Piconet Bluetooth). Puede alcanzar una velocidad de 250 Kbps. En una red ZigBee existen tres tipos de dispositivos:

- Coordinador ZigBee (Coordinator,ZC). El tipo de dispositivo más completo. Tiene que existir uno por cada red. Sus funciones son las de encargarse de controlar la red y los caminos que tienen que seguir los dispositivos para conectarse entre ellos. Requiere de memoria y capacidad de computación.
- Router ZigBee (ZR). Interconecta dispositivos separados en una topología de la red. Ofrece un nivel de aplicación para ejecutar un código de usuario.
- Dispositivo final (End Device,ZED). Tiene la funcionalidad necesaria de comunicarse al nodo padre (el coordinador o el router) pero no puede transmitir información destinada a otros dispositivos.

Un nodo ZigBee, tanto activo como pasivo, reduce su consumo gracias a que puede estar dormido la mayor parte del tiempo (incluso muchos días seguidos). Cuando se requiere su uso, el nodo ZigBee es capaz de despertar en un tiempo muy reducido y dormir otra vez cuando no lo necesiten. Un nodo cualquiera despierta aproximadamente en 15 milisegundos.

ZigBee permite tres topologías de red:

- Topología en estrella: el coordinador se sitúa en el centro.
- Topología en árbol: el coordinador será la raíz del árbol.
- Topología en malla: al menos uno de los nodos tendrá más de dos conexiones.

La seguridad de las transmisiones y los datos son puntos clave en la tecnología ZigBee. Ésta utiliza el modelo de seguridad de la subcapa MAC IEEE 802.15.4, la cual especifica cuatro servicios de seguridad:

- control de accesos (el dispositivo mantiene una lista de los dispositivos comprobados por la red),
- datos encriptados (se usa la encriptación con un código de 128 bits),
- integración de tramas (los datos se protegen para que no sean modificadas por otros),
- secuencia de refresco (se comprueba que las tramas no han sido reemplazadas por otras). El controlador de red comprueba estas tramas de refresco y su valor, para ver si son las esperadas.

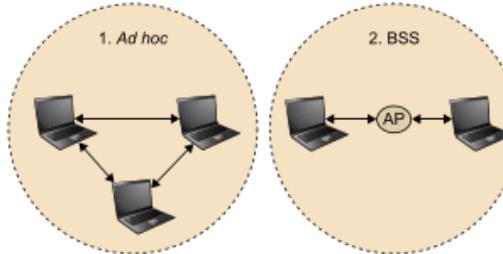
#### • WiFi (Wireless Fidelity)

Esta tecnología inalámbrica en sus distintas versiones (802.11a, b, g, n y ac) puede ofrecer desde 11 Mbits/s hasta 1 Gbits/s, y tiene distintas aplicaciones, especialmente en entornos locales (de corta distancia) como aeropuertos, hoteles, estaciones de servicio, centros comerciales, convenciones, pequeños pueblos..., en los que se ofrece acceso a Internet.

En Wi-Fi se utilizan las ondas portadoras de radio para transmitir la información. Los datos se superponen a la onda portadora de radio y se pueden extraer en el receptor final en un proceso conocido como modulación/demodulación.

Si las ondas se transmiten a diversas frecuencias, puede haber varias ondas portadoras de radio al mismo tiempo, sin que se interfieran unas con otras. Los puntos de acceso (access point) reciben la información, la guardan, y la retransmiten entre la red sin hilos y la red

cableada, esta red se llama de Infraestructura o BSS (Basic Service Set) . Si los dispositivos WiFi se comunican sin ningún punto de acceso, sino entre ellos, se crea una red llamada ad hoc.



La especificación concreta del sistema WiFi es la IEEE 802.11 y a lo largo del tiempo fueron aprobados diferentes estándares y otros quedaron en borradores.

Standard	Year	Band	Bandwidth	Modulation	Antenna Technology	Data Rate
802.11b	1999	2.4 GHz	20 MHz	CCK	—	11 Mb/s
802.11a	1999	5 GHz	20 MHz	OFDM	—	54 Mb/s
802.11g	2003	2.4 GHz	20 MHz	CCK, OFDM	—	54 Mb/s
802.11n	2009	2.4 GHz, 5 GHz	20 MHz, 40 MHz (up to 64-QAM)	OFDM	MIMO with up to four spatial streams, beamforming	600 Mb/s
802.11ac	—	5 GHz	40 MHz, 80 MHz, 160 MHz	OFDM (up to 256-QAM)	MIMO, MU-MIMO with up to eight spatial streams, beamforming	6.93 Gb/s
802.11ad (WiGig)	—	2.4 GHz, 5 GHz, 60 GHz	2.16 GHz	SC/OFDM	Beamforming	6.76 Gb/s

### 802.11a

La revisión 802.11a fue aprobada en 1999. El estándar 802.11a, opera en la banda de 5 GHz y utiliza 52 subportadoras orthogonal frequency-division multiplexing (OFDM) con una velocidad máxima de 54 Mbit/s. 802.11a tiene 12 canales sin solapa, 8 para red inalámbrica y 4 para conexiones punto a punto.

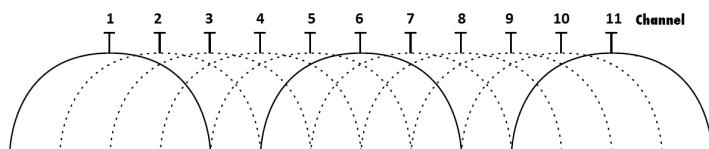
### 802.11b

La revisión 802.11b fue ratificada en 1999, tiene una velocidad máxima de transmisión de 11 Mbps y utiliza el mismo método de acceso definido en el estándar original CSMA/CA. El estándar 802.11b funciona en la banda de 2,4 GHz. Debido al espacio ocupado por la codificación del protocolo CSMA/CA, en la práctica, la velocidad máxima de transmisión con este estándar es de aproximadamente 5,9 Mbit/s sobre TCP y 7,1 Mbit/s sobre UDP.

### 802.11g

En junio de 2003, se ratificó un tercer estándar de modulación, 802.11g, que es la evolución de 802.11b. Este utiliza la banda de 2,4 GHz pero opera a una velocidad teórica máxima de 54 Mbit/s, que en promedio es de 22,0 Mbit/s de velocidad real de transferencia. Es compatible con el estándar b y utiliza las mismas frecuencias. Los equipos que trabajan bajo el estándar 802.11g llegaron al mercado muy rápidamente, incluso antes de su ratificación que fue dada aprox. el 20 de junio del 2003. Esto se debió en parte a que para construir equipos bajo este nuevo estándar se podían adaptar los ya diseñados para el estándar b. Actualmente todavía se venden equipos con esta especificación, con potencias de hasta medio vatío, que permite hacer comunicaciones de hasta 50 km con antenas parabólicas o equipos de radio apropiados

Los estándares IEEE 802.11b y 802.11g utilizan la banda de frecuencias de 2,4 a 2,5 GHz. Se ha definido once canales dentro de esta banda de señal. Pero no se utilizan todos, porque se sobreponen y producen interferencias. Los más utilizados son los canales 1, 6 y 11, ya que no son adyacentes y no reciben interferencias. Esta configuración sólo se hace habitualmente en el punto de acceso, ya que los dispositivos clientes detectan la señal.



### **802.11n**

En enero de 2004, el IEEE anunció la formación de un grupo de trabajo 802.11 para desarrollar una nueva revisión del estándar 802.11. La velocidad real de transmisión podría llegar a los 300Mbps. El alcance de operación de las redes es mayor gracias a la tecnología MIMO Multiple Input – Multiple Output, que permite utilizar varios canales a la vez para enviar y recibir datos gracias a la incorporación de varias antenas. El 802.11n puede trabajar en dos bandas de frecuencias: 2,4 GHz (la que emplean 802.11b y 802.11g) y 5 GHz (la que usa 802.11a). Gracias a ello, 802.11n es compatible con dispositivos basados en todas las ediciones anteriores de Wi-Fi. Además, es útil que trabaje en la banda de 5 GHz, ya que está menos congestionada y en 802.11n permite alcanzar un mayor rendimiento.

El estándar 802.11n fue ratificado por la organización IEEE el 11 de septiembre de 2009 con una velocidad de 600 Mbps en capa física. Actualmente existen varios productos que cumplen el estándar N con un máximo de 300 Mbps (80-100 estables).

### **802.11ac**

Es un estándar inalámbrico de red que trabaja en la banda de 5 GHz. Fue desarrollado entre 2011 y 2013 y se aprobó en enero de 2014. De acuerdo con un estudio, se espera que los dispositivos con la especificación 802.11ac se popularicen para el año 2015 con un estimado de mil millones extendidos por todo el mundo.

Esta especificación tiene una velocidad de al menos 1 gigabit por segundo en multiestaciones WLAN y una velocidad de al menos 500 megabits por segundo (500 Mbit/s) en un solo enlace. Esto se logra mediante la ampliación de los conceptos de 802.11n relacionados con ancho de banda de RF más amplio (hasta 160 MHz), más canales MIMO espaciales (hasta 8), MIMO multi-usuario y la modulación de alta densidad (hasta 256-QAM).

Para utilizar tanto las frecuencias de 2.4GHz o 5GHz no hace falta ninguna autorización (o licencia) de la administración competente, y por lo tanto, es una tecnología muy utilizada actualmente en entornos locales como edificios, oficinas, hospitales, etc., etc.

Las ondas de radio son emitidas mediante antenas, que simplemente son conductores por los que circula una corriente eléctrica de una cierta frecuencia. Cuanto mayor es la frecuencia, mas pequeña puede ser la antena. Las ondas de radio se transmiten en todas las direcciones del espacio lo que las hace ideales para transmisiones de tipo broadcast. Se utilizan antenas internas o externas, existen dos tipos de ellas:

- Omnidireccionales. Tienen poco alcance, pero permiten un radio de cobertura de 360°, unos 300 metros en el exterior.
- Unidireccionales. Tienen más alcance, pero sólo en una sola dirección.

Cuanto mayor es la frecuencia de la señal a transmitir, más factible es la transmisión unidireccional. Por tanto, para enlaces punto a punto se suelen utilizar microondas (altas frecuencias del orden de GHz) y para enlaces con varios receptores posibles se utilizan las ondas de radio a bajas frecuencias.

Uno de los problemas a los cuales se enfrenta actualmente la tecnología Wi-Fi es la progresiva saturación del espectro radioeléctrico, debido a la masificación de usuarios, esto afecta especialmente en las conexiones de larga distancia (mayor de 100 metros). En realidad Wi-Fi está diseñado para conectar computadoras a la red a distancias reducidas, cualquier uso de mayor alcance está expuesto a un excesivo riesgo de interferencias.

Un muy elevado porcentaje de redes son instalados sin tener en consideración la seguridad convirtiendo así sus redes en redes abiertas (o completamente vulnerables ante el intento de acceder a ellas por terceras personas), sin proteger la información que por ellas circulan. El acceso no autorizado a un dispositivo Wi-Fi es muy peligroso para el propietario por varios motivos. El más obvio es que pueden utilizar la conexión. Pero además, accediendo al Wi-Fi se puede monitorizar y registrar toda la información que se transmite a través de él (incluyendo información personal, contraseñas....). Existen varias alternativas para garantizar la seguridad de estas redes. Las más comunes son la utilización de protocolos de cifrado de datos para los estándares Wi-Fi como el WEP, el WPA, o el WPA2 que se encargan de codificar la información transmitida para proteger su confidencialidad, proporcionados por los propios dispositivos inalámbricos:

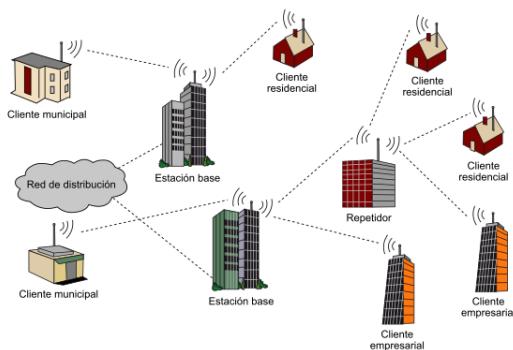
- WEP, cifra los datos en la red de forma que sólo el destinatario deseado pueda acceder a ellos. Los cifrados de 64 y 128 bits son dos niveles de seguridad WEP. WEP codifica los datos mediante una “clave” de cifrado antes de enviarlo al aire. Este tipo de cifrado no está muy recomendado debido a las grandes vulnerabilidades que presenta ya que cualquier cracker puede conseguir sacar la clave, incluso aunque esté bien configurado y la clave utilizada sea compleja.
- WPA: presenta mejoras como generación dinámica de la clave de acceso. Las claves se insertan como dígitos alfanuméricos.
- Filtrado de MAC, de manera que sólo se permite acceso a la red a aquellos dispositivos autorizados. Es lo más recomendable si solo se va a usar con los mismos equipos, y si son pocos.
- Ocultación del punto de acceso: se puede ocultar el punto de acceso de manera que sea invisible a otros usuarios.
- WPA2, es una mejora relativa a WPA. En principio es el protocolo de seguridad más seguro para Wi-Fi en este momento. Sin embargo requieren hardware y software compatibles, ya que los antiguos no lo son.

Sin embargo, no existe ninguna alternativa totalmente fiable, ya que todas ellas son susceptibles de ser vulneradas.

#### • WiMax

Actualmente hay una gran demanda de servicios de acceso de banda ancha (altas velocidades de transmisión) en Internet y otras aplicaciones de voz y datos. De manera cableada actualmente se ofrecen servicio con las líneas ADSL, y de manera inalámbrica WiMax es la solución que sirve tanto a los operadores de telecomunicaciones como a los usuarios.

El estándar WMAN de banda ancha apareció promovido y desarrollado por el grupo WiMax (wireless interoperability for microwave access) –acceso inalámbrico de banda ancha–, que tiene dos miembros muy representativos, como Intel y Nokia. La etiqueta WiMax está asociada globalmente al propio nombre del estándar IEEE 802.16. La tecnología WiMax supone una evolución con respecto al WiFi. Permite la conectividad entre puntos fijos, nómadas y móviles, y eventualmente la conectividad móvil sin necesidad de tener una línea punto a punto con una estación base.



WiMax utiliza bandas de frecuencia con y sin licencia gubernamental. La banda que no necesita ninguna autorización administrativa está entre 2,4 y 5 GHz. Estas bandas se tienen que utilizar con mucha cautela, ya que existe la posibilidad de una gran interferencia.

La norma inicial IEEE 802.16, publicada en diciembre del 2001, sirvió para fomentar la operatividad entre los sistemas local multipoint distribution system (LMDS). Inicialmente, el rango de frecuencias era entre 10 y 66 GHz con necesidad de visión directa (entre emisor y receptor). A principios del 2003, con la aparición del 802.16a para ratificar el estándar inicial 802.16, se amplió el rango de frecuencias hacia las bandas de 2 a 11 GHz. En el año 2004 aparece el estándar 802.16-2004 o IEEE 802.16d, también conocido como WiMAX, para cubrir las carencias del IEEE 802.16a.

802.16		802.16a	802.16e
Espectro	10-66 GHz	< 11 GHz	< 6 GHz
Funcionamiento	Sólo con visión directa	Sin visión directa (NLOS)	Sin visión directa (NLOS)
Ancho de banda	32-134 Mbps	Hasta 75 Mbps con canales de 20 MHz	Hasta 1.5 Mbps con canales de 5 MHz
Modulación	QPSK, 16QAM y 64 QAM	OFDM con 256 subportadoras	Mismo que 802.16a
Movilidad	Sistema fijo	Sistema fijo	Movilidad pedestre
Ancho del espectro	20, 25 y 28 MHz	Selección entre 1,25 y 20 MHz	El mismo que 802.16a con los canales de subida para ahorrar potencia
Distancia	2-5 km aprox.	5-50 km aprox.	2-5 km aprox.

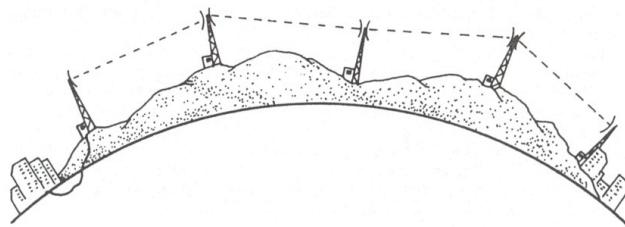
El actual estándar es 802.16m-2011 que tiene una tasa de transferencia de 100 Mbit/s móvil y 1 Gbit/s sobre punto fijo. También se conoce como Mobile WiMAX Release 2 o WirelessMAN-Advanced.

#### 2.4.2.3. Microondas terrestres.

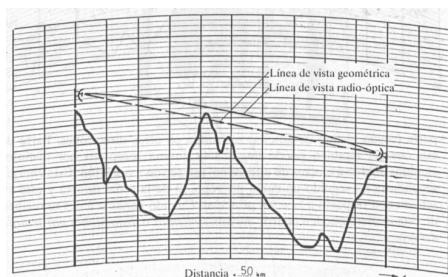
Las microondas, con frecuencias del orden de 1 GHz a 100 GHz, permiten un ancho de banda muy elevado, ya que la frecuencia de la portadora es muy grande. Esto permite la multicanalización de muchos mensajes. Las distancias que se permiten oscilan de 50 a 100 km. en transmisiones por la superficie terrestre y se suelen utilizar antenas parabólicas, con conexiones intermedias si la distancia a cubrir es muy larga.

Se suelen utilizar en sustitución del cable coaxial o las fibras ópticas ya que se necesitan menos repetidores y amplificadores, aunque se necesitan antenas alineadas. Se usan para transmisión de telefonía fija y televisión y son de utilización frecuente en la telefonía móvil, para enlazar las BTS (antenas fijas) con las BSC (dispositivos de comunicación móvil) y éstas con las MSC (central de conmutación móvil).

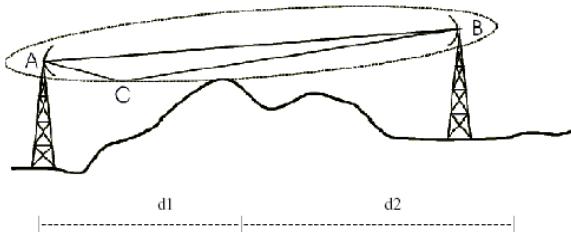
La principal causa de pérdidas es la atenuación, que aumenta con el cuadrado de la distancia (con cable coaxial y par trenzado son logarítmicas). La atenuación aumenta con la lluvia y la niebla y para ciertas frecuencias la pérdida de señal puede hacer que se interrumpa la comunicación. Las interferencias es otro inconveniente, ya que al proliferar estos sistemas, puede haber solapamiento de señales.



Cuando las distancias son extremadamente grandes, el número de repetidores sería también grande, por esta razón, se utilizan los satélites de comunicaciones soportados sobre satélites artificiales geoestacionarios.



De acuerdo a esto, los circuitos de radio enlace pueden establecerse si la condición de *línea de vista* existe. Sin embargo, debido a la refracción de la atmósfera, las ondas de radio que viajan paralelas a la superficie terrestre sufren una ligera curvatura hacia abajo, extendiendo el alcance de su línea de vista en un 5%, es costumbre referirse a la línea de vista *radio-óptica*.

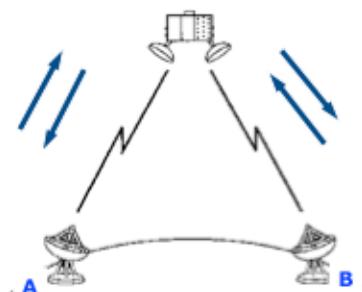


Una condición para la validez de las leyes ópticas es que solamente las ondas ionosféricas se propaguen desde el transmisor hacia el receptor (*propagación de espacio libre*). Para lograr esto, es necesario que cierto espacio alrededor del eje de la línea de vista radio-óptica esté libre de obstáculos. Este espacio puede ser descrito por la zona de Fresnel<sup>1</sup> que es el espacio comprendido en un elipsoide de revolución el cual la distancia entre las antenas es el eje mayor y el eje menor el cual varía directamente con la distancia e inversamente a la frecuencia, los puntos focales del elipsoide están ubicados en las antenas.

Los principales factores que afectan a la propagación de señales de radio en este rango de frecuencias son la lluvia, Niebla, Hielo y nieve, Gases atmosféricos y Vegetación

#### 2.4.2.4. Satélites.

Un satélite de comunicaciones hace la labor de repetidor electrónico. Una estación terrena A transmite al satélite señales de una frecuencia determinada (canal de subida *upstream*). Por su parte, el satélite recibe estas señales y las retransmite a otra estación terrena B mediante una frecuencia distinta (canal de bajada, *downstream*). La señal de bajada puede ser recibida por cualquier estación situada dentro del cono de radiación del satélite, y puede transportar voz, datos o imágenes de televisión. De esta manera se impide que los canales de subida y de bajada se interfieran, ya que trabajan en bandas de frecuencia diferentes.



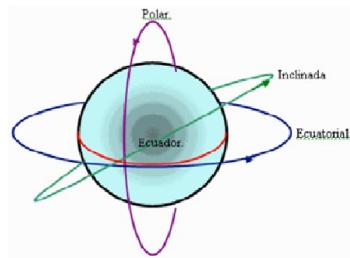
La capacidad que posee una satélite de recibir y retransmitir se debe a un dispositivo conocido como transpondedor. Los transpondedores de satélite trabajan a frecuencias muy elevadas, generalmente en la banda de los Gigahertzios.

Los satélites se pueden clasificar según:

- Su Distancia de la Tierra (Geoestacionaria, Geosíncrona, de Baja Altura, de Media Altura y Excéntricas).
- Su Plano Orbital con respecto al Ecuador (Ecuatorial, Inclinada y Polar).
- La Trayectoria Orbital que describen (Circular y Elíptica).

Elemento a Comparar	Orbita Geo	Orbita Meo	Orbita Leo
Altura (km)	36.000	6.000-12.000	200-3000
Período Orbital (Hr)	24	5-12	1.5
Velocidad (Km/hr)	11.000	19.000	27.000
Retraso (ida y vuelta) (ms)	250	80	10
Período de Visibilidad	Siempre	2-4 Hr	<15 min
Satélites necesarios para cobertura global	3	10-12	50-70

<sup>1</sup> Fresnel, Agustin Jean; 1788-1822, Físico Francés.



La mayoría de los satélites de comunicaciones están situados en una órbita denominada geoestacionaria, que se encuentra a 36000 Km. sobre el ecuador. Esto permite que el satélite gire alrededor de la tierra a la misma velocidad que ésta, de modo que parece casi estacionario. Así, las antenas terrestres pueden permanecer orientadas hacia una posición relativamente estable (sector orbital) ya que el satélite mantiene la misma posición relativa con respecto a la superficie de la tierra. Las características de estos medios de transmisión son:

Existe un retardo de unos 0.5 segundos en las comunicaciones debido a la distancia que han de recorrer las señales. Los cambios en los retrasos de propagación provocados por el movimiento en ocho de un satélite geoestacionario necesita transmisiones frecuentes de tramas de sincronización.

Los satélites tienen una vida media de diez a quince años, pero pueden sufrir fallos que provocan su salida de servicio. Es, por tanto, necesario disponer de un medio alternativo de servicio en caso de cualquier eventualidad.

Las estaciones terrenas suelen estar lejos de los usuarios y a menudo se necesitan caros enlaces de alta velocidad. Las estaciones situadas en la banda de bajas frecuencias (la banda C) están dotadas de grandes antenas (de unos 30 metros de diámetro) y son extremadamente sensibles a las interferencias. Por este motivo suelen estar situadas lejos de áreas habitadas. Las estaciones que trabajan en la banda Ku disponen de una antena menor y son menos sensibles a las interferencias. Utilizar un enlace de microondas de alta capacidad sólo ayudaría a complicar los problemas de ruido que presente el enlace con el satélite.

Banda	Frecuencias (GHz) Enlace ascendente	Frecuencias (GHz) Enlace descendente	Utilización
Banda L 1,5-2 GHz	1,61 – 1,66 1,93 – 2,01	1,452 – 1,61	Comunicaciones móviles por satélite,
Banda S 2-3 GHz	2,025- 2,11 2,655 – 2,69	2,29 – 2,5	Comunicaciones móviles por satélite, fijas, radiodifusión.
Banda C 6/4 GHz	5,925-6,425 (500 MHz)	3,7 – 4,2 (500 MHz)	Intelsat, Satélites nacionales: Westar, Satcom, Comstar, (USA), Anik (Canada), STW, Chinasat (CHINA), Palapa (Indonesia) Telecom I (Francia), CS-2 (Japón)
	5,725 – 6,275 (550 MHz)	3,4 – 3,9 (500 MHz)	Molinya, Intersputnik (URSS)
	5.850 – 7,075 (1255 MHz)	3,5 – 4,2, 4,5 – 4,8 (1100 MHz)	Bandas ampliadas, CAMR-79,85,88
Banda X 8/7 GHz	7,925 – 8,425 (500 MHz)	7,25 – 7,75 (500 MHz)	Satélites gubernamentales y militares
Banda Ku 13/11 GHz	12,75 – 13,25 (500 + 500 MHz)	10,7 - 12,75 (2.005 MHz)	Intelsat, Eutelsat, satélites nacionales, DBS
Banda K 18/12 GHz	17,3 – 18,4		Radiodifusión, fijos
Banda Ka 30/20 GHz	27,5 – 31	17,2 – 21,2	Japón, Europa, USA Enlaces intersatelitales

Las comunicaciones con el satélite pueden ser interceptadas por cualquiera que disponga de un receptor en las proximidades de la estación. Es necesario utilizar técnicas de encriptación para garantizar la privacidad de los datos.

Los satélites geoestacionarios pasan por períodos en los que no pueden funcionar. En el caso de un eclipse de Sol en el que la tierra se sitúa entre el Sol y el satélite, se corta el suministro de energía a las células solares que alimentan el satélite, lo que provoca el paso del suministro de energía a las baterías de emergencia, operación que a menudo se traduce en una reducción de las prestaciones o en una pérdida de servicio. En el caso de tránsitos solares, el satélite pasa directamente entre el Sol y la Tierra provocando un aumento del ruido térmico en la estación terrena, y una pérdida probable de la señal enviada por el satélite. Los satélites geoestacionarios no son totalmente estacionarios con respecto a la órbita de la tierra. Las

desviaciones de la órbita ecuatorial hace que el satélite describa una figura parecida a un ocho, de dimensiones proporcionales a la inclinación de la órbita con respecto al ecuador. Estas variaciones en la órbita son corregidas desde una estación de control.

Actualmente hay un problema de ocupación de la órbita geoestacionaria. Cuando un satélite deja de ser operativo, debeirse irse a otra órbita, para dejar un puesto libre. La separación angular entre satélites debe ser de 2 grados (anteriormente era de 4). Esta medida implicó la necesidad de mejorar la capacidad de resolución de las estaciones terrenas para evitar detectar las señales de satélites próximos en la misma banda en forma de ruido.



**Tema III**

## CONTROL DE ACCESO AL MEDIO (MAC)

### SUB CAPA 1.5

Las redes por su tecnología se clasifican en redes broadcast y redes formadas por enlaces punto a punto. En este último caso la información se envía al computador situado al otro lado del enlace, que está claramente identificado y el medio de transmisión normalmente está siempre disponible.

En las redes broadcast hay una complejidad añadida. Dado que el canal de comunicación es compartido entre varios computadores, es preciso habilitar mecanismos que permitan a cada uno de ellos utilizarlo para enviar sus tramas al computador de destino. El hecho de compartir el canal generará conflictos o incluso pérdida de tramas (colisiones) en algunos casos; los protocolos deberán establecer los mecanismos adecuados para resolver dichos conflictos y permitir que los computadores retransmitan en caso necesario las tramas que no hayan podido ser enviadas correctamente.

Debido a esta característica singular de las redes broadcast la capa de enlace tiene en ellas una complejidad mayor que en las redes punto a punto, por lo que el modelo OSI se suele dividir en este caso en dos subcapas: la inferior, que se ocupa de controlar esta nueva función de acceso al medio de transmisión que hemos comentado, se denomina subcapa MAC (Medium Access Control); la superior, conocida como subcapa LLC (Logical Link Control) corresponde a las funciones de la capa de enlace comunes a todo tipo de redes que veremos en el siguiente tema.

#### **3.1. COLISIÓN**

Una colisión es una interferencia en la transmisión de un computador, producida por otro u otros computadores, o por las condiciones del medio de transmisión. Puede haber más de dos computadores involucrados en la colisión. Un computador puede interferir consigo mismo.

Se produce una colisión cuando, a pesar de que un equipo pretende transmitir una cierta serie de bits, las alteraciones que se producen en el medio de transmisión no se corresponden con esa secuencia de bits. El computador que transmite es la única capaz de detectar que se está produciendo una colisión, porque es la única que conoce qué es lo que pretende emitir. Para detectar una colisión, el computador debe comparar (bit a bit) lo que pretende enviar, con lo que realmente está siendo enviado por el medio. El computador tiene que tener la capacidad de poder "escuchar" (recibir) mientras está enviando. Los factores que intervienen en una colisión son la velocidad de propagación, distancia, velocidad de transmisión y longitud del mensaje.

#### **3.2. ESTRATEGIAS O METODOS DE ACCESO AL MEDIO**

Todas las redes locales consisten en una colección de dispositivos que deben compartir la capacidad de transmisión de la red. Para ello, se hace necesaria la existencia de algún método o estrategia para controlar el acceso al medio de transmisión evitando los posibles conflictos o errores que puedan aparecer. El protocolo de control de acceso al medio de transmisión es el factor que más caracteriza el funcionamiento de una red de área local. De él dependen los parámetros básicos del funcionamiento de la red como son el rendimiento, la fiabilidad y la gestión de la red. Para regular el control de acceso al medio, se pueden utilizar estrategias estáticas o dinámicas

##### **Estrategias estáticas**

Dividen el uso del recurso de modo fijo. Son ineficientes, porque cuando un computador no tiene nada para transmitir, los otros no pueden aprovechar la capacidad ociosa.

- TDM (time division multiplexing)
- FDM (frequency division multiplexing)

##### **Estrategias dinámicas**

Se clasifican, a su vez, en

- Aleatorias o de Contención: Todos los nodos tienen el control en cada momento (CSMA-Ethernet)
- Distribuidas : Sólo un nodo tiene el control en cada momento (Token bus, token ring)
- Centralizadas: Un único nodo tiene el control, y periódicamente invita a los otros a transmitir (Sondeo

o Polling)

### 3.2.1. Asignación estática de canales

La manera tradicional de asignar un solo canal entre múltiples usuarios competidores es dividir su capacidad mediante el uso de uno de los esquemas de multiplexación como el FDM (Multiplexación por División de Frecuencia). Si hay  $N$  usuarios, el ancho de banda se divide en  $N$  partes de igual tamaño, y a cada usuario se le asigna una parte. Debido a que cada usuario tiene una banda de frecuencia privada, ahora no hay interferencia entre ellos. Cuando sólo hay una pequeña cantidad fija y constante de usuarios, cada uno tiene un flujo estable o una carga de tráfico pesada, esta división es un mecanismo de asignación sencillo y eficiente. Las estaciones de radio de FM son un ejemplo inalámbrico. Cada estación recibe una parte de la banda de FM y la utiliza la mayor parte del tiempo para difundir su señal. Dividir el único canal disponible en varios subcanales estáticos es ineficiente por naturaleza. El problema básico es que, cuando algunos usuarios están inactivos, su ancho de banda simplemente se pierde. No lo están usando, y a nadie más se le permite usarlo.

Por otro lado, si se usara la multiplexación por división de tiempo (TDM) y a cada usuario se le asignara una  $N$ -ésima ranura de tiempo, en caso de que un usuario no utilizara la ranura asignada, simplemente se desperdicia.

### 3.2.2. Asignación dinámica de canales

El trabajo hecho en esta área se basa en cinco supuestos clave, que se describen a continuación:

- **Tráfico independiente.** El modelo consiste en  $N$  estaciones independientes (computadoras, teléfonos), cada una con un programa o usuario que genera tramas para transmisión. El número esperado de tramas que se generan en un intervalo de longitud  $Dt$  es de  $\lambda Dt$ , donde  $\lambda$  es una constante (la tasa de llegada de tramas nuevas). Una vez que se ha generado una trama, la estación se bloquea y no hace nada sino hasta que la trama se haya transmitido con éxito.
- **Canal único.** Hay un solo canal disponible para todas las comunicaciones. Todas las estaciones pueden transmitir en él y pueden recibir de él. Se asume que las estaciones tienen una capacidad equivalente, aunque los protocolos pueden asignarles distintos roles (prioridades).
- **Colisiones observables.** Si dos tramas se transmiten en forma simultánea, se traslanan en el tiempo y la señal resultante se altera. Este evento se llama colisión. Todas las estaciones pueden detectar una colisión que haya ocurrido. Una trama en colisión se debe volver a transmitir después. No hay otros errores, excepto aquéllos generados por las colisiones.
- **Tiempo continuo o ranurado.** Se puede asumir que el tiempo es continuo, en cuyo caso la transmisión de una trama puede comenzar en cualquier momento. Por el contrario, el tiempo se puede ranurar o dividir en intervalos discretos (llamados ranuras). En este caso las transmisiones de las tramas deben empezar al inicio de una ranura. Una ranura puede contener 0, 1 o más tramas, correspondientes a una ranura inactiva, una transmisión exitosa o una colisión, respectivamente.
- **Detección de portadora o sin detección de portadora.** Con el supuesto de detección de portadora, las estaciones pueden saber si el canal está en uso antes de intentar usarlo. Si se detecta que el canal está ocupado, ninguna estación intentará utilizarlo. Si no hay detección de portadora, las estaciones no pueden detectar el canal antes de intentar usarlo. Simplemente transmiten. Sólo después pueden determinar si la transmisión tuvo éxito.

## 3.3. PROTOCOLOS DE ACCESO MÚLTIPLE

Se conocen muchos algoritmos para asignar un canal de acceso múltiple. En la siguiente parte se estudiará una muestra representativa de los más interesantes y se darán algunos ejemplos de cómo se usan comúnmente en la práctica.

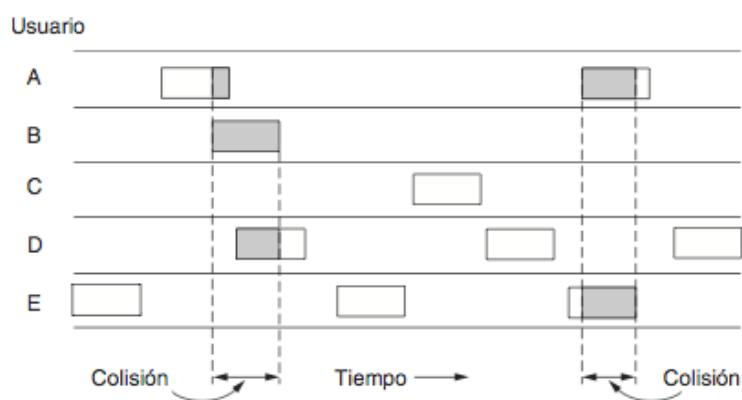
### 3.3.1. Protocolo sin detección de portadora: ALOHA

En 1970, cuando la red ARPANET solo llevaba unos meses en funcionamiento, un equipo de la Universidad de Hawái quería poner en marcha una red para interconectar terminales ubicados en las islas de Kauai, Maui y Hawaii, con un computador situado en Honolulu, en la isla de Oahu. Lo normal habría sido utilizar enlaces telefónicos, pero la baja calidad y el elevado costo de las líneas hacían

prohibitiva esta opción. El equipo estaba decidido a llevar a cabo su proyecto a toda costa, pero no a cualquier costo. Consiguieron varios transmisores de radio taxis viejos y construyeron módems de forma artesanal. Con esto pusieron en marcha una red de radioenlaces entre las islas. Si se hubiera asignado un canal diferente para la comunicación en cada sentido entre Oahu y las otras tres islas habrían hecho falta seis canales; en vez de eso asignaron solamente dos: uno a 413,475 MHz para las transmisiones de Oahu a las demás islas y otro a 407,350 MHz para el sentido inverso. Cada canal tenía un ancho de banda de 100 KHz y una capacidad de 9,6 Kb/s. En caso de haber creado seis canales en el mismo ancho de banda la capacidad de cada uno habría sido proporcionalmente menor; creando solo dos se disponía de una mayor capacidad a costa de tener que compartirlos entre las tres islas. Las transmisiones desde Oahu no planteaban problemas pues había un único emisor. Sin embargo el canal de retorno era compartido por tres emisores (Kauai, Maui y Hawaii), por lo que podía suceder que dos emisores transmitieran simultáneamente, con lo que se producía una colisión con lo que ambas tramas se perdían; había pues que establecer reglas que especificaran como se resolvía una situación de este tipo; estas reglas es lo que denominamos un protocolo de acceso al medio o protocolo MAC (Medium Access Control).

La solución que adoptaron fue muy simple. Cuando un emisor quiere transmitir una trama simplemente la emite, sin preocuparse en ningún momento de si el canal está libre; una vez ha terminado se pone a la escucha, esperando recibir confirmación de que la información ha sido recibida correctamente por el destinatario. Si pasado un tiempo razonable no se recibe confirmación el emisor supone que ha ocurrido una colisión; en este caso espera un tiempo aleatorio (para no colisionar nuevamente) y a continuación reenvía la trama. Este protocolo MAC, que fue el primero en implementarse, se denominó Aloha. La red creada en Hawái se denominó ALOHANET. Aloha es una palabra Hawaiana que se utiliza para saludar, tanto al llegar como al despedirse. Seguramente esta ambigüedad pareció apropiada a sus inventores para indicar el carácter multidireccional o *broadcast* de la transmisión, por contraste con los enlaces punto a punto utilizados hasta entonces donde se sabe con certeza si la información va o viene.

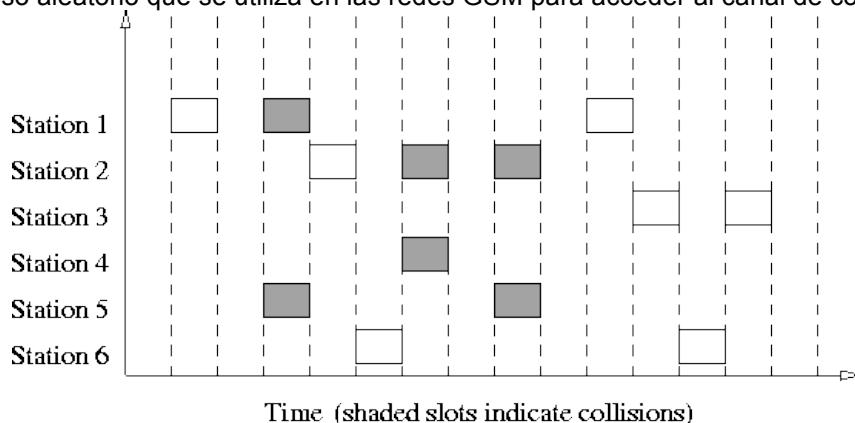
En el protocolo Aloha original (puro) la emisión de tramas por parte de cada computador se hace de forma completamente caótica y basta que dos tramas colisionen o se solapen solamente en un bit para que ambas sean completamente inútiles, a pesar de lo cual tanto la primera como la segunda serán irremediablemente transmitidas, ya que los emisores sólo se percatarán del problema después de haber terminado la transmisión; además la segunda trama podría colisionar con una tercera, y así sucesivamente; en una red Aloha cuando el tráfico crece las colisiones aumentan de manera no lineal y el rendimiento decae rápidamente. En la siguiente figura se presenta un ejemplo de la generación de tramas en un sistema ALOHA. Todas las tramas son de la misma longitud porque la velocidad real de transmisión (throughput) de los sistemas ALOHA se maximiza al tener tramas con un tamaño uniforme en lugar de tramas de longitud variable. Cada vez que dos tramas traten de ocupar el canal al mismo tiempo, habrá una colisión y ambas se dañarán. Si el primer bit de una trama nueva se traslape con el último bit de una trama casi terminada, ambas se destruirán por completo (es decir, tendrán sumas de verificación incorrectas) y ambas tendrán que volver a transmitirse más tarde. La suma de verificación no distingue (y no debe) entre una pérdida total y un error ligero. El uso del canal es aproximadamente el 18%.



### 3.3.2. Protocolo sin detección de portadora: ALOHA RANURADO

En 1972 Roberts propuso una mejora al protocolo Aloha que consistía en dividir el tiempo para la emisión de tramas en intervalos de duración constante. De alguna manera los computadores estarían sincronizadas y todos sabrían cuando empezaba cada intervalo. Esto reduce la probabilidad de colisión, ya que al menos limita su efecto a un intervalo concreto. A esta versión mejorada de Aloha se la denomina Aloha ranurado, porque utiliza tiempo ranurado o a intervalos.

Los protocolos Aloha aún se utilizan hoy en día (normalmente Aloha ranurado) en situaciones donde no es posible o no es práctico detectar las colisiones en tiempo real, por ejemplo algunas redes de satélite o el canal de acceso aleatorio que se utiliza en las redes GSM para acceder al canal de control.



### 3.3.3. Protocolo con detección de portadora: CSMA 1-persistente

En Aloha los computadores se ponen a transmitir sin preguntar antes si el canal está libre. Hay protocolos más diplomáticos, que antes de transmitir miran si alguien ya lo está haciendo. Esto permite hacer un uso más eficiente del canal y llegar a mayores grados de ocupación, ya que no se interrumpe la transmisión en curso. Estos protocolos se denominan de acceso múltiple con detección de portadora o **CSMA** (*Carrier Sense Multiple Access*); la denominación “detección de portadora” hace referencia a esa consulta previa sobre la ocupación del canal.

En su nivel más primitivo el protocolo CSMA hace lo siguiente: cuando tiene una trama que enviar primero escucha el canal para saber si está libre; si lo está envía la trama; en caso contrario espera a que se libere y en ese momento la envía. Este protocolo se denomina **CSMA 1-persistent**e porque hay una probabilidad 1 (es decir certeza) de que la trama se transmita cuando el canal esté libre.

En una situación real con tráfico intenso es muy posible que cuando un computador termine de transmitir haya varios esperando para enviar su trama; con CSMA 1-persistent todas esas tramas serán emitidas a la vez y colisionarán, pudiéndose repetir el proceso varias veces con la consiguiente degradación del rendimiento. En realidad la colisión ocurre aunque no empiecen a transmitir exactamente a la vez: basta con que dos computadores empiecen a transmitir con una diferencia de tiempos menor que la distancia que los separa, ya que en tal caso ambos detectarán el canal libre en el momento de iniciar la transmisión. A pesar de sus inconvenientes el CSMA 1-persistent supone un avance respecto al ALOHA ranurado, ya que toma la precaución de averiguar antes si el canal está disponible, con lo que se evitan un buen número de colisiones.

### 3.3.4. Protocolo con detección de portadora: CSMA no persistente

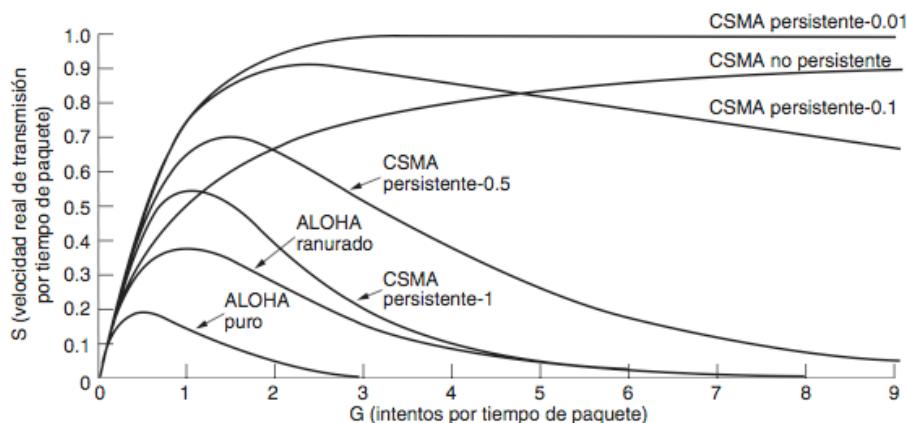
En un intento por resolver el problema de colisiones de CSMA 1-persistent se puede adoptar la estrategia siguiente: antes de enviar se escucha, si el canal está libre se transmite, pero si está ocupado, en vez de estar a la escucha, pendientes de usarlo en cuanto se libere, se espera un tiempo aleatorio después del cual se repite el proceso; a este protocolo se le denomina **CSMA no persistente**. Este protocolo tiene una menor eficiencia que CSMA 1-persistent para tráficos moderados, pues introduce una mayor latencia; sin embargo se comporta mejor en situaciones de tráfico intenso ya que evita las

colisiones producidas por los computadores que se encuentran a la espera de que termine la transmisión de una trama en un momento dado.

### 3.3.5. Protocolo con detección de portadora: CSMA p-persistente

CSMA p-persistente intenta combinar las ventajas de CSMA 1-persistente y CSMA no persistente. Este protocolo se aplica con tiempo ranurado o a intervalos y funciona de la siguiente manera: cuando el computador tiene algo que enviar primero escucha el canal, si está libre transmite, en caso contrario espera; cuando el canal se libera transmite con una probabilidad  $p$  (o no transmite con una probabilidad  $q=1-p$ ); si no transmite en el primer intervalo el proceso se repite en el siguiente, es decir transmite con una probabilidad  $p$ , o no transmite con una probabilidad  $q$ . El proceso se repite hasta que finalmente la trama es transmitida o bien otro computador utiliza el canal, en cuyo caso espera un tiempo aleatorio y empieza de nuevo el proceso desde el principio.

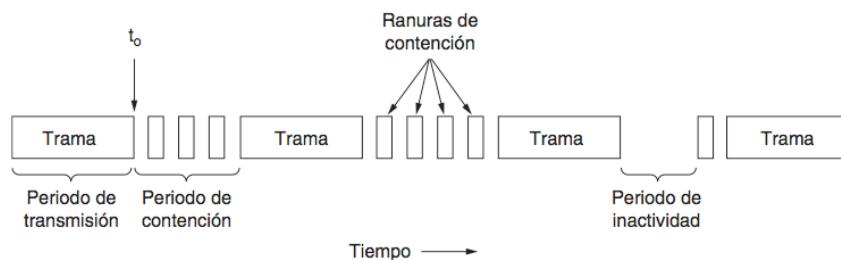
Ajustando el valor del parámetro  $p$  el funcionamiento de este protocolo se puede regular en todo el rango entre el de CSMA 1-persistente y el de CSMA no persistente. Su eficiencia es en general superior a la de ambos.



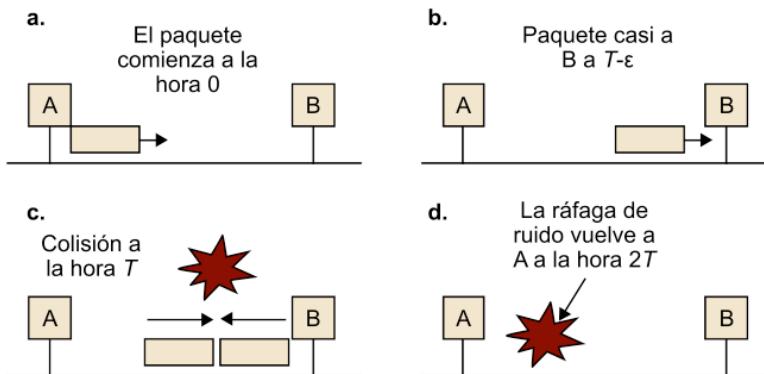
### 3.3.6. Protocolo con detección de portadora: CSMA con detección de colisión

En los protocolos que se han descrito hasta ahora una vez se había empezado a transmitir una trama el computador seguía transmitiendo aun cuando detectara una colisión. En ese caso sería lógico y más eficiente parar de transmitir, ya que la trama será errónea e inútil. Esta mejora es la que incorporan los protocolos conocidos como **CSMA/CD** (Carrier Sense Multiple Access with Collision Detection, acceso múltiple con escucha de portadora y detección de colisiones) que se utiliza en la red local IEEE 802.3, también conocida como Ethernet, en sus múltiples variantes.

CSMA/CD utiliza el modelo conceptual de la siguiente figura. En el punto marcado como  $t_0$ , una estación ha terminado de transmitir su trama. Cualquier otra estación que tenga una trama por enviar puede intentar hacerlo ahora. Si dos o más estaciones deciden transmitir en forma simultánea, habrá una colisión. Si una estación detecta una colisión, aborta la transmisión, espera un tiempo aleatorio e intenta de nuevo (suponiendo que ninguna otra estación ha comenzado a transmitir durante ese lapso). Por lo tanto, el modelo de CSMA/CD consistirá en períodos alternantes de contención y transmisión, con períodos de inactividad que ocurrirán cuando todas las estaciones estén en reposo (por ejemplo, por falta de trabajo).



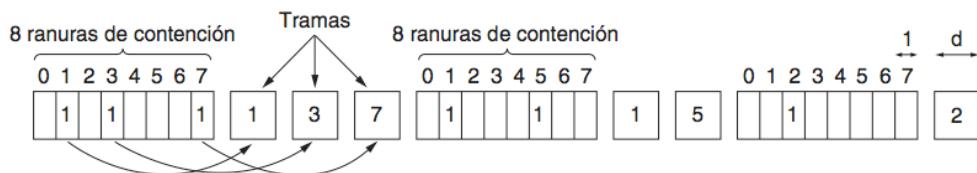
En una red CSMA/CD la única circunstancia en la que puede producirse una colisión es cuando dos computadores empiezan a transmitir a la vez, o con una diferencia de tiempo lo bastante pequeña como para que la señal de uno no haya podido llegar al otro antes de que éste empiece a transmitir. Suponer que se tiene dos computadores A y B situados en extremos opuestos de la red y que la señal tarda un tiempo  $\tau$  en propagarse de uno a otro extremo a otro; cabría pensar que si A empieza a transmitir pasado ese tiempo  $\tau$  ya puede estar seguro de que no observará colisiones, ya que sus señal ha llegado al otro extremo de la red; pero en el caso más desfavorable B podría haber empezado a transmitir justo en el instante  $\tau - \epsilon$ , o sea inmediatamente antes de que le haya llegado la trama de A; por lo que sólo después de un tiempo  $2\tau$  puede A estar seguro de no colisionar con ningún otro computador, habiéndose entonces "apoderado" del canal de transmisión. Dado que el período de incertidumbre en CSMA/CD se reduce a ese intervalo  $2\tau$  estas redes se suelen modelar como un sistema ALOHA ranurado con intervalos de tamaño  $2\tau$ .



### 3.3.7. Protocolos sin colisiones: Bitmap

En cualquiera de los protocolos que se ha visto hasta ahora puede haber competencia entre computadores por acceder al medio. Dicha competencia produce colisiones, que en la práctica suponen una disminución del rendimiento ya que las transmisiones que se producen durante la colisión son inútiles; estos efectos se agravan a medida que aumenta el tráfico en la red, ya que la probabilidad de colisiones aumenta. Las cosas mejoran a medida que se refina el protocolo, pero incluso con CSMA/CD cuando la ocupación del canal es elevada el rendimiento empieza a bajar.

Suponiendo que la red tiene N computadores, numerados de 0 a N-1. Para empezar a funcionar se establece una ronda "exploratoria" de N intervalos en la que por riguroso turno cada computador, empezando por el 0, tiene la posibilidad de enviar un bit con el valor 1 ó 0 para indicar si tiene alguna trama que transmitir. Pasados N intervalos todos los computadores han podido manifestar su situación, y todos saben quien tiene tramas para transmitir. Por ejemplo, si se tienen 8 computadores, y que después de la ronda inicial se sabe que los computadores 1, 3 y 7 tienen tramas para transmitir. Entonces toma la palabra el computador 1, que transmite la trama que tenía pendiente. Después vendrá el 3 y por último el 7. Agotados los turnos que había solicitados se inicia otra ronda de sondeo para saber quien tiene tramas pendientes de transmitir, y así sucesivamente.



Puede suceder que a algún computador le surja la necesidad de transmitir una trama justo después de haber dejado pasar su turno; en tal caso tendrá que esperar a la siguiente vuelta.

Desde el punto de vista del rendimiento este protocolo genera una trama adicional de N bits. Si la red no tiene tráfico se generará una trama bitmap que estará continuamente dando vueltas por la red. Si la carga en la red es baja (una trama transmitida por vuelta) la eficiencia es  $d/(N+d)$ , donde  $d$  es el tamaño de la

trama de información transmitida y  $N$  el número de computadores. Si la red está saturada cada computador tendrá una trama que enviar y la eficiencia será  $Nd/(Nd+N)$ , o sea  $d/(d+1)$ . Se ve que el rendimiento de este protocolo aumenta a medida que lo hace el tráfico en la red, justo lo contrario de lo que ocurría con los protocolos basados en colisiones.

En situaciones de saturación donde todos los computadores tienen tramas que transmitir, y suponiendo que todas las tramas tienen el mismo tamaño el protocolo bitmap produce un reparto equitativo, por lo que resulta equivalente a utilizar multiplexación por división en el tiempo para repartir el canal entre los computadores de la red.

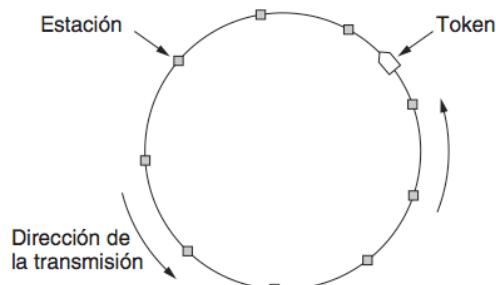
El protocolo bitmap resulta más eficiente y más homogéneo en su comportamiento a medida que la carga de la red aumenta. Los protocolos como se han descrito, en los que se emite un paquete indicando el deseo de transmitir información, se denominan *protocolos de reserva*.

### 3.3.8. Protocolos sin colisiones: Paso de token

La esencia del protocolo de mapa de bits es que permite que cada estación transmita una trama por turno, en un orden predefinido. Otra forma de lograr lo mismo es pasar un pequeño mensaje conocido como token de una estación a otra, en el mismo orden predefinido. El token representa el permiso para enviar. Si una estación tiene una trama puesta en cola para transmitirla cuando recibe el token, puede enviar esa trama antes de pasar el token a la siguiente estación. Si no tiene una trama puesta en cola, simplemente pasa el token.

En un protocolo token ring, la topología de la red se utiliza para definir el orden en el que las estaciones envían información. Las estaciones están conectadas una con otra en un solo anillo. Así, el proceso de pasar el token a la siguiente estación consiste en recibir el token proveniente de una dirección y transmitirlo hacia la otra dirección. Las tramas también se transmiten en la dirección del token. De esta forma, circularán alrededor del anillo y llegarán a la estación de destino. Sin embargo, para evitar que la trama circule en forma indefinida (como el token), una estación necesita quitarla del anillo. Esta estación puede ser la que envió originalmente la trama, después de que haya pasado por un ciclo completo, o la estación destinada a recibir la trama.

Cabe mencionar que no se necesita un anillo físico para implementar el paso del token. El canal que conecta a las estaciones podría ser también un solo bus extenso. Así, cada estación puede usar el bus para enviar el token a la siguiente estación en la secuencia predefinida. Al poseer el token, una estación puede usar el bus para enviar una trama, como antes. A este protocolo se le conoce como token bus.



El desempeño del protocolo de paso de token es similar al del protocolo de mapa de bits, aunque las ranuras de contención y las tramas de un ciclo están ahora entremezcladas. Después de enviar una trama, cada estación debe esperar a que las  $N$  estaciones (incluyéndose a sí misma) envíen el token a sus estaciones vecinas y que las otras  $N - 1$  estaciones envíen una trama, si es que la tienen. Una sutil diferencia es que, como todas las posiciones en el ciclo son equivalentes, no hay parcialidad por las estaciones de menor o de mayor numeración. Para token ring, cada estación también envía el token sólo hasta su estación vecina antes de que el protocolo lleve a cabo el siguiente paso. No es necesario propagar cada token a todas las estaciones antes de que el protocolo avance al siguiente paso.

Las redes de token ring han surgido como protocolos MAC con cierta consistencia. Uno de los primeros protocolos de este tipo (conocido como "Token Ring", que se estandarizó como IEEE 802.5) fue popular en la década de 1980 como alternativa a la Ethernet clásica. En la década de 1990, una red token ring mucho más veloz conocida como FDDI (Interfaz de Datos Distribuidos por Fibra) fue vencida por la

Ethernet conmutada. En la década de 2000, una red token ring llamada RPR (Anillo de Paquetes con Recuperación) se definió como el IEEE 802.17 para estandarizar la mezcla de anillos de área metropolitana que usaban los ISP.

### 3.3.9. Protocolos sin colisiones: Protocolo de cuenta atrás binaria

El protocolo bitmap requiere reservar un intervalo de un bit para cada computador. Con un número elevado de computadores esto puede suponer un costo elevado que lo haga impracticable. Se tiene una alternativa que resuelve ese inconveniente, el protocolo denominado *cuenta atrás binaria*.

Suponiendo que se tiene una red con 16 computadores. Cada uno recibirá una dirección codificada en 4 bits. Suponiendo también que los computadores 0010, 0100, 1001 y 1010 desean transmitir tramas. El protocolo de cuenta atrás binaria procede de la siguiente forma:

1. En el primer intervalo los cuatro computadores que desean transmitir envían a la red el primer bit de su dirección; el medio de transmisión está diseñado de tal forma que retransmite el OR de todos los bits transmitidos, es decir en este caso los cuatro computadores reciben un 1.
2. Al haber recibido un 1 los computadores 0010 y 0100 (que tienen un 0 en su primer bit) reconocen que hay computadores superiores a ellos en la competición y se retiran; los dos "finalistas" envían a la red su segundo bit, que es cero para ambos; la red retransmite un cero.
3. Al haber recibido un cero los dos computadores siguen compitiendo y envían su tercer bit, un cero para 1001 y un 1 para 1010; la red retransmite un 1 y el computador 1001 se retira al ver que hay uno que le supera.
4. El computador ganador, el 1010, envía su trama.



El proceso se repite para los tres computadores restantes, y así sucesivamente hasta que eventualmente todos envían su trama. La eficiencia de este protocolo es  $d/(d + \ln N)$ , que para tráficos reducidos supera al bitmap; además, el mecanismo de selección suministra la dirección del computador transmisor que a menudo es parte de la información que se pretende transmitir, con lo que incluso este overhead se aprovecha y la eficiencia puede ser del 100%.

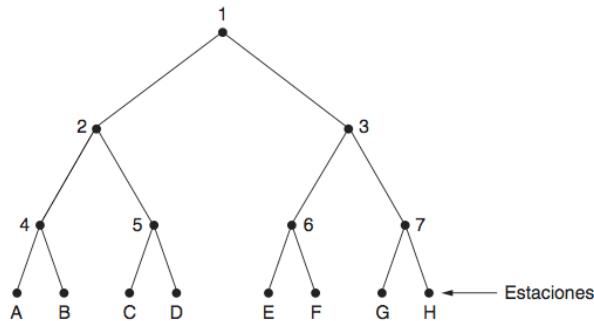
### 3.3.10. Protocolos de contención limitada: El protocolo de recorrido de árbol adaptable

Se ha visto que los protocolos con contención (es decir con colisiones) son ideales cuando los niveles de tráfico son bajos, ya que tienen retardos pequeños y no introducen sobrecarga (overhead); todos los datos transmitidos son tramas de información útil. En cambio, cuando el tráfico aumenta, es preferible perder una parte de la capacidad del canal en habilitar mecanismos que habiliten "turnos de palabra", ya que de lo contrario no es posible utilizar el canal al máximo de sus posibilidades.

Cabría pensar en un protocolo ideal que contuviera lo mejor de ambos mundos. Debería ser lo bastante astuto como para funcionar de forma "caótica" (es decir con colisiones) a bajos niveles de tráfico, y poner en marcha mecanismos de arbitraje riguroso en caso de que el tráfico aumente por encima de ciertos niveles considerados peligrosos, es decir, debería ser autoadaptativo. Este tipo de protocolos se denomina protocolos de contención limitada.

En caso de que la red tenga poco tráfico estos protocolos se comportarán según alguno de los protocolos con colisiones que hemos visto. Pero cuando se superen determinados umbrales de ocupación el protocolo dividirá el canal en intervalos asignando uno a cada computador, en riguroso turno. Este comportamiento es equivalente a realizar multiplexación por división en el tiempo sobre el canal. En la práctica suelen ser unos pocos computadores los que generan la mayor parte del tráfico, por lo que lo ideal es identificar a los culpables y aislarlos en intervalos propios, independientes del resto de los computadores; de esta forma esos computadores con tráfico elevado consiguen un buen rendimiento sin perjudicar a la mayoría "silenciosa". Precisamente la pronta identificación de esos culpables es la clave del funcionamiento de estos protocolos. Los computadores no necesariamente han de ser identificados individualmente, es suficiente detectar un grupo con tráfico elevado (que presumiblemente contendrá algún "sospechoso") y aislarlo del resto. Uno de los protocolos que funciona con este principio es el denominado protocolo de recorrido de árbol adaptable.

Una manera muy sencilla de llevar a cabo la asignación necesaria es usar el algoritmo desarrollado por el ejército de Estados Unidos para hacer pruebas de sífilis a los soldados durante la Segunda Guerra Mundial (Dorfman, 1943). En esencia, el ejército tomaba una muestra de sangre de  $N$  soldados. Se vaciaba una parte de cada muestra en un solo tubo de ensayo. Luego se examinaba esta muestra mezclada en busca de anticuerpos. Si no se encontraban, todos los soldados del grupo se declaraban sanos. Si se encontraban anticuerpos, se preparaban dos nuevas muestras mezcladas, una de los soldados 1 a  $N/2$  y otra de los demás. El proceso se repetía en forma recursiva hasta que se determinaban los soldados infectados. Para la versión de computadora de este algoritmo (Capetanakis, 1979) es conveniente considerar a las estaciones como hojas de un árbol binario, como se muestra en la siguiente figura. En la primera ranura de contención después de la transmisión exitosa de una trama (ranura 0), se permite que todas las estaciones intenten adquirir el canal. Si una de ellas lo logra, qué bueno. Si hay una colisión, entonces durante la ranura 1, sólo aquellas estaciones que queden bajo el nodo 2 del árbol podrán competir. Si alguna de ellas adquiere el canal, la ranura que sigue después de la trama se reservará para las estaciones que están bajo el nodo 3. Por otra parte, si dos o más estaciones bajo el nodo 2 quieren transmitir, habrá una colisión durante la ranura 1, en cuyo caso será el turno del nodo 4 durante la ranura 2.



Si ocurre una colisión durante la ranura 0, se examina todo el árbol para localizar todas las estaciones listas. Cada ranura de bits está asociada a un nodo específico del árbol. Si ocurre una colisión, continúa la búsqueda en forma recursiva con el hijo izquierdo y el derecho del nodo. Si una ranura de bits está inactiva o si sólo una estación que transmite en ella, se puede detener la búsqueda de su nodo, ya que se han localizado todas las estaciones listas (si hubiera existido más de una, habría ocurrido una colisión).

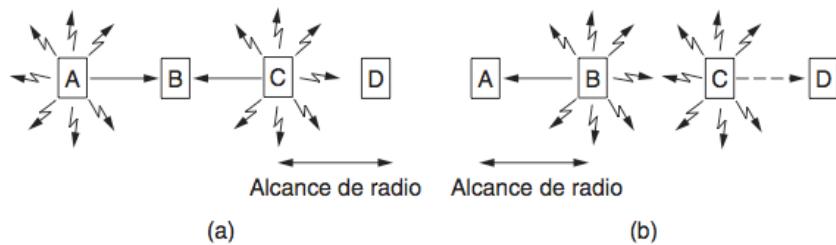
Aunque hay una gama muy amplia de protocolos MAC que han sido propuestos en teoría, modelados por simulación e incluso probados en redes experimentales, en la práctica las posibles opciones se reducen a un número muy pequeño. Además el protocolo MAC va implícito en la tecnología de red local utilizada, que muchas veces se decide en base otros factores, tales como costo, disponibilidad de productos, etc. por lo que el margen de maniobra en cuanto a la elección del protocolo MAC es prácticamente nulo.

### 3.3.11. Protocolos de redes inalámbricas MACA : CSMS/CA

Las ondas electromagnéticas no guiadas son un medio ideal para la creación de redes broadcast; ya se ha visto como algunas de las primeras experiencias (Aloha) se hicieron con este tipo de medios de transmisión. Actualmente, con el auge de los sistemas móviles han aparecido redes locales basadas en ondas radioeléctricas e infrarrojos; los sistemas infrarrojos por sus características tienen un alcance reducido y requieren estricta visión directa entre emisor y receptor. Los de radio solo pueden transmitir a muy baja potencia (0,1 W) por restricciones legales, por lo que su alcance es también reducido, aunque no tanto como los infrarrojos. Normalmente se emplea la banda conocida como Industrial/Científica /Médica (2,4 - 2,484 GHz). Típicamente una LAN inalámbrica está formada por un conjunto de computadores base, unidos entre sí por algún tipo de cable, y una serie de computadores móviles que se comunican con el computador base más próximo. El conjunto de computadores base forma en realidad un sistema celular en miniatura.

El problema de las colisiones debe encararse de modo diferente. En redes inalámbricas, lo que importa es la percepción del receptor. No hay forma de que el emisor determine qué sucede en el receptor. Midiendo el nivel de energía de radio frecuencia, una estación puede determinar si hay otra cuya emisión puede perturbarla. Mucho más simple que full dúplex.

Dada la topología y alcance de la siguiente figura, cuando A transmite, su señal llega hasta B, pero no llega ni a C, ni a D. Una transmisión de A hacia B puede hacer colisión con otra cuyo destino sea B, pero no afecta si los destinos son C o D



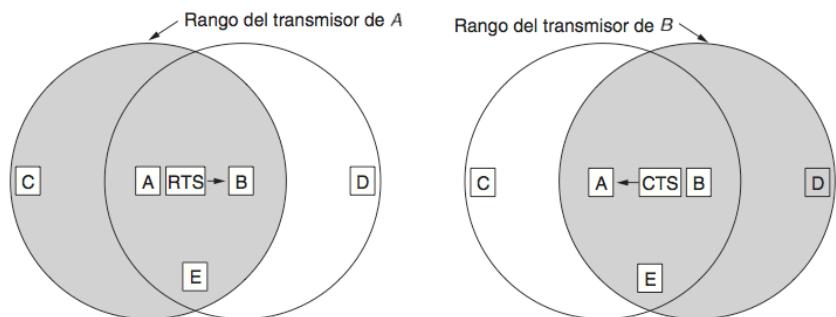
- **El problema de la estación escondida**

En (a), A comienza a transmitir hacia B, C escucha el canal, y no detecta actividad, C envía un mensaje hacia B, provocando una colisión.

- **El problema de la estación expuesta**

En (b), B comienza a transmitir hacia A, C, que pretende enviar hacia D, escucha el canal, y percibe actividad, C no envía su mensaje hacia D, aunque no provocaría colisión con el de B hacia A

MACA (Multiple Access with Collision Avoidance) es el protocolo MAC que ha servido de base para el estándar IEEE 802.11 que es el que especifica el funcionamiento de LANs inalámbricas. MACA resuelve los dos problemas antes mencionados.



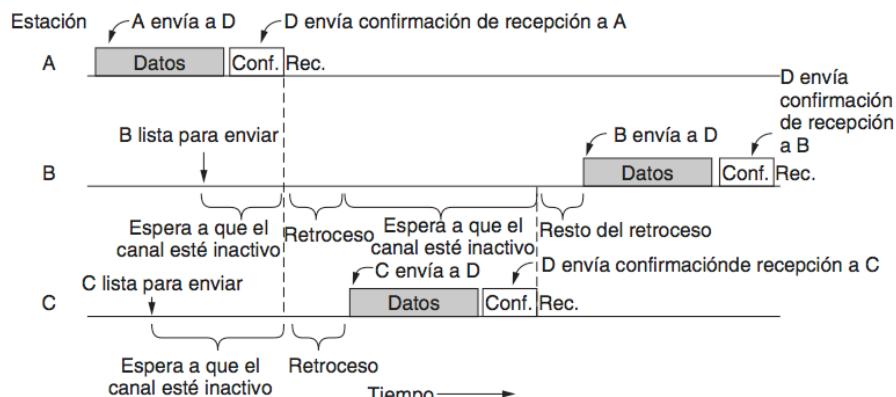
Cuando una estación tiene una trama que transmitir antes de enviarla envía una trama pequeña de aviso (de 30 bytes) denominada RTS (Request To Send). La trama RTS contiene información sobre la longitud de la trama que se pretende transmitir y la estación de destino.

Al recibir la trama RTS la estación de destino, si está en condiciones de recibir la transmisión, responde con otra trama denominada CTS (Clear To Send). La trama CTS también indica la longitud de la trama

que se va a recibir. Cualquier estación que escuche el RTS está bastante cerca de A y debe permanecer en silencio durante el tiempo suficiente para que el CTS se transmita de regreso a A sin conflicto. Es evidente que cualquier estación que escuche el CTS está bastante cerca de B y debe permanecer en silencio durante la siguiente transmisión de datos, cuya longitud puede determinar examinando la trama CTS. En la figura, C está en el alcance de A pero no en el alcance de B. Por lo tanto, escucha el RTS de A pero no el CTS de B. En tanto no interfiera con el CTS, está libre para transmitir mientras se envía la trama de datos. En contraste, D está en el alcance de B pero no de A. No escucha el RTS pero sí el CTS. Al escuchar el CTS sabe que está cerca de una estación que está a punto de recibir una trama, por lo que difiere el envío de cualquier cosa hasta el momento en que se espera la terminación de esa trama. La estación E escucha ambos mensajes de control y, al igual que D, debe permanecer en silencio hasta que se haya completado la trama de datos.

El 802.11 trata de evitar colisiones con un protocolo llamado **CSMA/CA** (CSMA con Evitación de Colisiones), que se basa en MACA. En concepto, este protocolo es similar al CSMA/CD de Ethernet, con detección del canal antes de enviar y retroceso exponencial después de las colisiones. Sin embargo, una estación que desee enviar una trama empieza con un retroceso aleatorio (excepto en el caso en que no haya utilizado el canal recientemente y éste se encuentre inactivo). No espera una colisión. La estación espera hasta que el canal está inactivo, para lo cual detecta que no hay señal durante un periodo corto y realiza un conteo descendente de las ranuras inactivas, haciendo pausa cuando se envían tramas. Envía su trama cuando el contador llega a 0. Si la trama logra pasar, el destino envía de inmediato una confirmación de recepción corta. La falta de una confirmación de recepción se interpreta como si hubiera ocurrido un error, sea una colisión o cualquier otra cosa. En este caso, el emisor duplica el periodo de retroceso e intenta de nuevo, continuando con el retroceso exponencial como en Ethernet, hasta que la trama se transmite con éxito o se llegue al número máximo de retransmisiones.

En la figura siguiente se muestra una línea de tiempo de ejemplo. La estación A es la primera en enviar una trama. Mientras A envía, las estaciones B y C se preparan para enviar. Ven que el canal está ocupado y esperan a que esté inactivo. Poco después de que A recibe una confirmación de recepción, el canal queda inactivo. Sin embargo, en vez de enviar una trama de inmediato y colisionar, B y C realizan un retroceso. C elige un retroceso corto, por lo que envía primero. B detiene su conteo mientras detecta que C está usando el canal y lo reanuda después de que C recibe una confirmación de recepción. Poco después, B completa su retroceso y envía su trama.

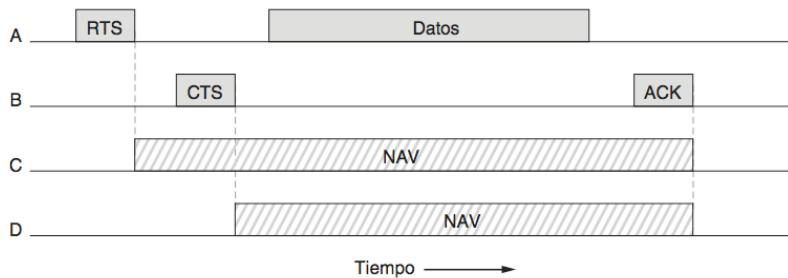


Para decidir qué estación va a transmitir en caso de tener el canal habilitado, el 802.11 define la detección del canal como un proceso que consiste tanto de una detección física como de una detección virtual. En la detección física sólo se verifica el medio para ver si hay una señal válida. En la detección virtual, cada estación mantiene un registro lógico del momento en que se usa el canal rastreando el NAV (Vector de Asignación de Red). Cada trama lleva un campo NAV que indica cuánto tiempo tardará en completarse la secuencia a la que pertenece esta trama. Las estaciones que escuchen por casualidad esta trama saben que el canal estará ocupado durante el periodo indicado por el NAV, sin importar que puedan detectar o no una señal física.

Hay un mecanismo RTS/CTS opcional que usa el NAV para evitar que las terminales envíen tramas al mismo tiempo como terminales ocultas. En la siguiente figura, A desea enviar a B. C es una estación dentro del alcance de A. D es una estación dentro del alcance de B, pero no dentro del alcance de A.

El protocolo empieza cuando A decide que desea enviar datos a B. A empieza por enviar una trama RTS a B para solicitar permiso de enviarle una trama. Si B recibe esta solicitud, responde con una trama CTS para indicar que el canal está libre para enviar. Al recibir la CTS, A envía su trama e inicia un temporizador ACK. Al recibir de forma correcta la trama de datos, B responde con una trama ACK para completar el intercambio. Si el temporizador ACK de A expira antes de que la trama ACK vuelva a ella, se considera como una colisión y se lleva a cabo todo el protocolo de nuevo, después de un retroceso.

C está dentro del alcance de A, por lo que puede recibir la trama RTS. Si pasa esto, se da cuenta de que alguien pronto va a enviar datos. A partir de la información proporcionada en la solicitud RTS, C puede estimar cuánto tardará la secuencia, incluyendo la trama ACK final. Entonces, desiste de transmitir cualquier cosa hasta que el intercambio esté completo. A continuación actualiza su registro del NAV para indicar que el canal está ocupado. D no escucha el RTS pero sí el CTS, por lo que también actualiza su NAV. Es necesario tener en cuenta que las señales NAV no se transmiten; sólo son recordatorios internos para mantenerse en silencio durante cierto periodo.



### 3.4. REDES ETHERNET

Una vez tratados de manera general los protocolos de asignación de canal, es tiempo de ver la forma en que estos principios se aplican a sistemas reales. Muchos de los diseños para las redes personales, locales y de área metropolitana se han estandarizado bajo el nombre de IEEE 802. Algunos han sobrevivido pero muchos no. Los sobrevivientes más importantes son el 802.3 (Ethernet) y el 802.11 (LAN inalámbrica).

Ethernet, probablemente es el tipo más común de red de computadoras en el mundo. Existen dos clases de Ethernet: **Ethernet clásica**, que resuelve el problema de acceso múltiple mediante el uso de las técnicas que se han estudiado en este capítulo; el segundo tipo es la **Ethernet conmutada**, en donde los dispositivos llamados switches se utilizan para conectar distintas computadoras. Es importante mencionar que, aunque se hace referencia a ambas como Ethernet, son muy diferentes. La Ethernet clásica es la forma original que operaba a tasas de transmisión de 3 a 10 Mbps. La Ethernet conmutada es en lo que se convirtió la Ethernet y opera a 100, 1000 y 10000 Mbps, en formas conocidas como Fast Ethernet, Gigabit Ethernet y 10 Gigabit Ethernet. Actualmente, en la práctica sólo se utiliza Ethernet conmutada.

Se analizará estas formas históricas de Ethernet en orden cronológico para mostrar cómo se desarrollaron. Puesto que Ethernet y el IEEE 802.3 son idénticos, excepto por una pequeña diferencia, muchas personas usan los términos "Ethernet" e "IEEE 802.3" sin distinción.

#### 3.4.1. IEEE 802.3 - Ethernet Clásica

La red Ethernet fue inventada por Xerox en 1976. Esta red primitiva permitía conectar más de 100 computadores en un área de aproximadamente 1 km, a una velocidad de 2.94 Mbps. Debido al éxito alcanzado, Xerox, DEC e Intel Corporation desarrollaron una versión mejorada que permitía tasas de transferencia de 10 Mbps. La estandarización de la LAN Ethernet se llama IEEE 802.3.

La red Ethernet utiliza un medio común (cable coaxial o UTP), que puede ser usado en un instante determinado por una sola computadora para transmitir. El cable llega a todas las computadoras e independientemente de su estructura física, se comporta como un bus, donde todo lo que se coloca en el cable llega a todas las computadoras.

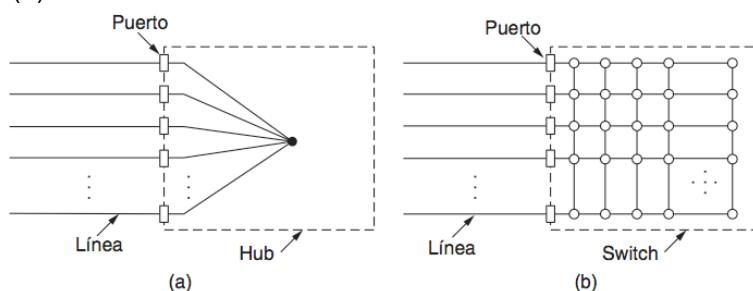
Una red Ethernet está compuesta de uno o más segmentos de cable (unidos por repetidores) y puede construirse con diferentes tecnologías. En la descripción de una red Ethernet, se usa la nomenclatura del tipo XYZ donde X especifica la tasa de transferencia medida en Mbps, Y determina el tipo de transmisión y Z define la longitud máxima de un tramo sin repetidor medido en cientos de metros; T si es par trenzado (10baseT) o F si es fibra optica (10baseF).

**10Base5:** Se conoce también como Ethernet de cable coaxial grueso (Thick Ethernet). Permite una tasa de transferencia de 10 Mbps, la transmisión es en banda base (modulación digital Manchester), los segmentos pueden tener hasta 500 m (distancia máxima entre repetidores que a lo sumo pueden ser 4) y un computador debe separarse de otro al menos 2,5 m. La longitud máxima del enlace no puede superar los 2.500 m. Solo pueden colocarse 200 computadores en cada tramo, lo que da un número máximo de 1000 computadores. Lo más destacable del 10Base5 es que las computadoras no se conectan directamente al cable coaxial, sino a través de unos dispositivos llamados transceivers. Los transceivers tienen una aguja metálica que atraviesa el cable hasta llegar a la línea de conductor interno (por este motivo, a estos dispositivos también se les conoce con el nombre de vampiros). Del transceiver salen 15 cables que pueden tener una longitud máxima de 50 m y que van a parar al conector del interface de red (NIC) del computador. La ventaja de los transceivers es que no es necesario cortar la línea para instalar una nueva computadora.

**10Base2:** Llamado también Ethernet de cable coaxial fino (thin Ethernet), permite tasas de 10 Mbps en banda base (modulación Manchester), pero los segmentos pueden ser a lo sumo de 200 m., (185 m en la práctica). En este caso, las computadoras se conectan directamente a la red colocando un conector T del tipo BNC en el punto de inserción. El hardware que antes se encontraba en el transceiver ahora se encuentra en el interface de red del computador. Para instalar un nuevo computador es necesario cortar la línea e instalar un conector T.

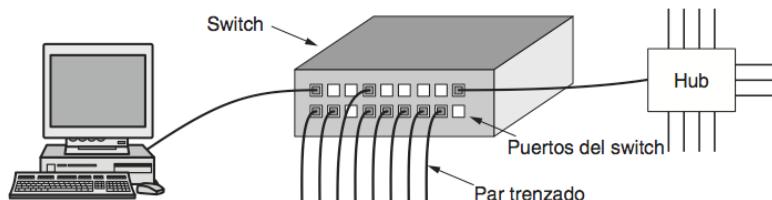
### 3.4.2. Ethernet conmutada

Ethernet empezó a evolucionar y a alejarse de la arquitectura de un solo cable extenso de la Ethernet clásica. Los problemas asociados con el hecho de encontrar interrupciones o conexiones flojas condujeron hacia un distinto tipo de patrón de cableado, en donde cada estación cuenta con un cable dedicado que llega a un hub (concentrador) central. Un hub simplemente conecta de manera eléctrica todos los cables que llegan a él, como si estuvieran soldados en conjunto. Esta configuración se muestra en la siguiente figura (a).

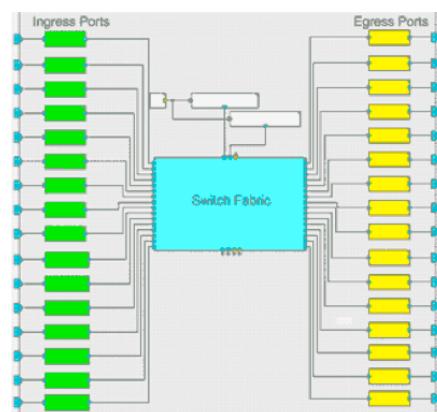


Los cables eran pares trenzados de la compañía telefónica, ya que la mayoría de los edificios de oficinas contaban con este tipo de cableado y por lo general había muchos de sobra. Esta reutilización fue una ventaja, pero a la vez se redujo la distancia máxima de cable del hub hasta 100 metros. En esta configuración es más simple agregar o quitar una estación, además de que los cables rotos se pueden detectar con facilidad. Con las ventajas de usar el cableado existente y la facilidad de mantenimiento, los hubs de par trenzado se convirtieron rápidamente en la forma dominante de Ethernet. Sin embargo, los hubs no incrementan la capacidad debido a que son lógicamente equivalentes al cable extenso individual de la Ethernet clásica. A medida que se agregan más estaciones, cada estación recibe una parte cada vez menor de la capacidad fija. En un momento dado, la LAN se saturará. Una forma de solucionar esto es usar una Ethernet conmutada. El corazón de este sistema es un conmutador (switch) que contiene un plano posterior (backplane) de alta velocidad, el cual conecta a todos los puertos como se muestra en la figura anterior (b). Desde el exterior, un switch se ve igual que un hub. Ambos son cajas que por lo general contienen de 4 a 48 puertos, cada uno con un conector estándar RJ-45 para un cable de par

trenzado. Cada cable conecta al switch o hub con una sola computadora. Un switch tiene también las mismas ventajas que un hub. Es fácil agregar o quitar una nueva estación con sólo conectar o desconectar un cable, y es fácil encontrar la mayoría de las fallas, ya que un cable o puerto defectuoso por lo general afectará a una sola estación. De todas formas hay un componente compartido que puede fallar (el mismo switch).



Sin embargo, dentro del switch ocurre algo muy distinto. Los switches sólo envían tramas a los puertos para los cuales están destinadas. Cuando el puerto de un switch recibe una trama Ethernet de una estación, el switch verifica las direcciones de Ethernet para ver cuál es el puerto de destino de la trama. Este paso requiere que el switch sea capaz de deducir qué puertos corresponden a qué direcciones. A continuación, el switch reenvía la trama a través de su plano posterior (switch fabric) de alta velocidad hacia el puerto de destino. Por lo general, el plano posterior opera a muchos Gbps mediante el uso de un protocolo propietario que no necesita estandarización, ya que está completamente oculto dentro del switch. Después, el puerto de destino transmite la trama sobre el cable, de manera que pueda llegar a la estación de destino. Ninguno de los otros puertos sabe siquiera que existe la trama.

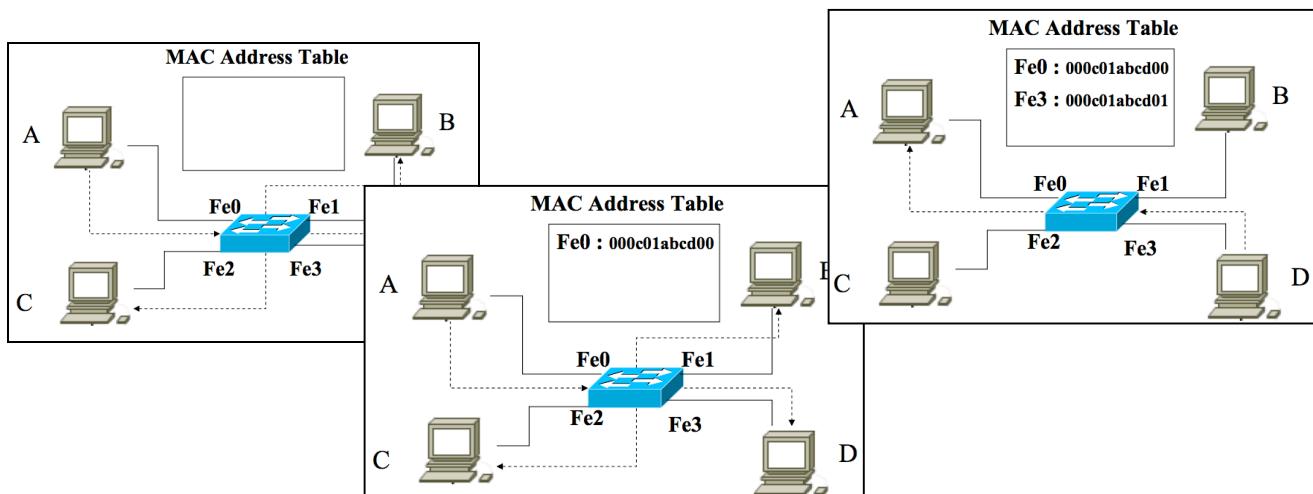


Si más de una estación o puerto desea enviar una trama al mismo tiempo, los switches difieren de los hubs. En un hub, todas las estaciones están en el mismo dominio de colisión. Deben usar el algoritmo CSMA/CD para programar sus transmisiones. En un switch, cada puerto es su propio dominio de colisión independiente.

Un switch mejora el desempeño de la red en comparación con un hub de dos maneras. Primero, como no hay colisiones, la capacidad se utiliza con más eficiencia. Segundo y más importante, con un switch se pueden enviar varias tramas al mismo tiempo (por distintas estaciones). Estas tramas llegarán a los puertos del switch y viajarán hacia el plano posterior de éste para enviarlos por los puertos apropiados. No obstante, como se podrían enviar dos tramas al mismo puerto de salida y al mismo tiempo, el switch debe tener un búfer para que pueda poner temporalmente en cola una trama de entrada hasta que se pueda transmitir al puerto de salida. En general, estas mejoras producen una considerable ganancia en el desempeño que no es posible lograr con un hub. Con frecuencia, la velocidad real de transmisión total del sistema se puede incrementar en un orden de magnitud, dependiendo del número de puertos y patrones de tráfico.

En el funcionamiento de un switch se analiza una trama entrante (dirección MAC) y redirecciona la misma únicamente a la salida correspondiente. Requiere de tablas de conmutación (enrutamiento) para determinar la interfaz de salida correspondiente a cada equipo en la red.

Por ejemplo si A le envía información a D,



**Aprendizaje** se registra la MAC A en la tabla

**Inundación flooding** para la MAC destino

**Filtrado de tramas** se registra la MAC D en la tabla y se hace "Frame filtering" para la MAC A destino, que se encuentra en la tabla.

### 3.4.3. FastEthernet

802.3u, fue aprobado de manera oficial por el IEEE en junio de 1995. Técnicamente, 802.3u no es un nuevo estándar sino un agregado al estándar 802.3 existente, para su compatibilidad con versiones anteriores. Es común llamarlo Fast Ethernet en vez de 802.3u. El tipo de cableado que soporta es:

Nombre	Cable	Segmento máximo	Ventajas
100Base-T4	Par trenzado	100 m	Utiliza UTP categoría 3.
100Base-TX	Par trenzado	100 m	Full-dúplex a 100 Mbps (UTP cat 5).
100Base-FX	Fibra óptica	2000 m	Full-dúplex a 100 Mbps; distancias largas.

El esquema UTP categoría 3, llamado 100Base-T4, utilizaba una velocidad de señalización de 25 MHz, tan sólo un 25% más rápida que los 20 MHz de la Ethernet estándar. Sin embargo, para alcanzar la tasa de bits necesaria, 100Base-T4 requiere cuatro cables de par trenzado. De los cuatro pares, uno siempre va al hub, uno siempre sale del hub y los otros dos se pueden conmutar a la dirección actual de la transmisión. Para obtener 100 Mbps de los tres pares trenzados en la dirección de la transmisión, se utiliza un esquema bastante complejo en cada par trenzado, que implica enviar dígitos ternarios con tres distintos niveles de voltaje.

Ethernet 100Base-TX, llegó a dominar el mercado. Este diseño es más simple puesto que los cables pueden manejar velocidades de reloj de 125 MHz. Sólo se utilizan dos pares trenzados por estación, uno que va al hub y otro que viene de él. No se utiliza la codificación binaria directa (es decir, NRZ) ni la codificación Manchester. Se utiliza la codificación 4B/5B. Se codifican 4 bits de datos como 5 bits de señal y se envían a 125 MHz para proveer 100 Mbps. El sistema 100Base-TX es full-dúplex; las estaciones pueden transmitir a 100 Mbps en un par trenzado y recibir a 100 Mbps en otro par trenzado al mismo tiempo.

La última opción, 100Base-FX, utiliza dos filamentos de fibra multimodo, una para cada dirección, por lo que también es full-dúplex con 100 Mbps en cada dirección. En esta configuración, la distancia entre una estación y el switch puede ser de hasta 2 km.

Casi todos los switches Fast Ethernet pueden manejar una mezcla de estaciones de 10 Mbps y 100 Mbps. Para facilitar la actualización, el estándar provee por sí solo un mecanismo llamado autonegociación, el cual permite que dos estaciones negocien de manera automática la velocidad óptima (10 o 100 Mbps) y la duplicidad (half-dúplex o full-dúplex). La mayoría de los productos Ethernet usan esta característica para configurarse a sí mismos.

#### 3.4.4. Gigabit Ethernet

El IEEE trabajó en 802.3ab en 1999, llamada Gigabit Ethernet. Los objetivos del comité para la Gigabit Ethernet eran en esencia los mismos que los del comité para Fast Ethernet: que tuviera un desempeño 10 veces mayor y que mantuviera la compatibilidad con todos los estándares Ethernet existentes. Gigabit Ethernet ofrece servicio de datagramas sin confirmación de recepción con unidifusión y multidifusión, utiliza el mismo esquema de direccionamiento de 48 bits que ya estaba en uso y mantiene el mismo formato de trama, incluyendo los tamaños mínimo y máximo de trama.

Al igual que Fast Ethernet, Gigabit Ethernet soporta dos modos diferentes de funcionamiento: modo full-dúplex y modo half-dúplex. El modo "normal" es el modo full-dúplex, que permite tráfico en ambas direcciones al mismo tiempo. Este modo se utiliza cuando hay un switch central conectado a computadoras o a otros switches. En esta configuración, todas las líneas se almacenan en el búfer con el fin de que cada computadora y switch pueda enviar tramas siempre que lo deseé. El emisor no tiene que detectar el canal para ver si alguien más lo está utilizando debido a que la contención es imposible. En la línea entre una computadora y un switch, la computadora es la única que puede enviar al switch y la transmisión tendrá éxito aun cuando el switch esté enviado ahora una trama a la computadora. Los switches tienen la libertad de mezclar e igualar velocidades. La autonegociación se soporta al igual que en Fast Ethernet, sólo que ahora la opción está entre 10, 100 y 1000 Mbps.

El otro modo de operación es half-dúplex y se utiliza cuando las computadoras están conectadas a un hub en vez de un switch. Un hub no almacena las tramas entrantes. En su lugar, conecta en forma eléctrica todas las líneas internamente, simulando el cable con múltiples derivaciones que se utiliza en la Ethernet clásica. En este modo puede haber colisiones, por lo que se requiere el protocolo CSMA/CD estándar.

Gigabit Ethernet soporta tanto el cableado de cobre como el de fibra óptica. La señalización cerca de 1 Gbps requiere codificar y enviar un bit cada nanosegundo. En un principio este truco se lograba con cables cortos de cobre blindados (la versión 1000Base-CX) y con fibra óptica. En la fibra óptica se permiten dos longitudes de onda y resultan dos versiones distintas: 0.85 micras (corto, para 1000Base-SX) y 1.3 micras (largo, para 1000Base-LX).

Nombre	Cable	Segmento máximo	Ventajas
1000Base-SX	Fibra óptica	550 m	Fibra multimodo (50, 62.5 micras)
1000Base-LX	Fibra óptica	5000 m	Monomodo (10 μ) o multimodo (50, 62.5μ)
1000Base-CX	2 pares de STP	25 m	Par trenzado blindado
1000Base-T	4 pares de UTP	100 m	UTP estándar categoría 5

La señalización en la longitud de onda corta se puede realizar mediante LEDs económicos. Se utiliza con fibra multimodo y es útil para las conexiones dentro de un edificio, ya que puede funcionar hasta por 500 m para la fibra de 50 micras. La señalización en la longitud de onda larga requiere láser más costosos. Por otro lado, al combinarse con fibra monomodo (10 micras), la longitud de cable puede ser de hasta 5 km. Para enviar bits por estas versiones de Gigabit Ethernet, se utiliza la codificación 8B/10B, que codifica 8 bits de datos en palabras codificadas de 10 bits que se envían a través del cable o la fibra.

### 3.4.5. 10 Gigabit Ethernet

Aparecieron estándares para fibra y cable de cobre blindado por primera vez en 2002 y 2004, seguidos de un estándar para par trenzado de cobre en 2006.

10 Gbps es una velocidad realmente prodigiosa, 1000 veces más rápida que la Ethernet original. ¿En dónde se podría necesitar? La respuesta es que dentro de los centros e intercambios de datos para conectar enruteadores, switches y servidores de gama alta, así como en las troncales de larga distancia con alto ancho de banda entre las oficinas que permiten la operación de redes de área metropolitana completas, basadas en Ethernet y fibra. Las conexiones de larga distancia usan fibra óptica, mientras que las conexiones cortas pueden usar cobre o fibra.

Todas las versiones de Ethernet de 10 gigabits soportan sólo la operación full-dúplex. CSMA/CD ya no forma parte del diseño y los estándares se concentran en los detalles de las capas físicas que pueden operar a muy alta velocidad. Pero la compatibilidad aún sigue siendo importante, por lo que las interfaces Ethernet de 10 gigabits usan la autonegociación y cambian a la velocidad más alta soportada por ambos extremos de la línea.

En la figura siguiente se listan los principales tipos de Ethernet de 10 gigabits. Se utiliza fibra multimodo con la longitud de onda de 0.85 m (corta) para distancias medias, y la fibra monomodo a 1.3 m (larga) y 1.5 m (extendida) para distancias largas. 10GBase-ER puede operar en distancias de 40 km, lo cual la hace adecuada para aplicaciones de área amplia. Todas estas versiones envían un flujo serial de información que se produce mediante el mezclado de los bits de datos, para después codificarlos mediante un código **64B/66B**. Esta codificación tiene menos sobrecarga que un código 8B/10B.

Nombre	Cable	Segmento máximo	Ventajas
10GBase-SR	Fibra óptica	Hasta 300 m	Fibra multimodo (0.85 μ).
10GBase-LR	Fibra óptica	10 km	Fibra monomodo (1.3 μ).
10GBase-ER	Fibra óptica	40 km	Fibra monomodo (1.5 μ).
10GBase-CX4	4 pares de twinax	15 m	Cobre twinaxial.
10GBase-T	4 pares de UTP	100 m	UTP categoría 6a

La primera versión de cobre que se definió (10GBase-CX4) utiliza un cable con cuatro pares de cableado twinaxial de cobre. Cada par usa codificación 8B/10B y opera a 3.125 Gsímbolos/segundo para alcanzar 10 Gbps. Esta versión es más económica que la fibra y fue de las primeras en comercializarse.

10GBase-T es la versión que usa cables UTP. Aunque requiere cableado categoría 6a o 7, en distancias más cortas puede usar categorías más bajas (incluyendo la categoría 5 o 5e) para reutilizar una parte del cableado ya instalado. Cada uno de los cuatro pares trenzados se utiliza para enviar 2500 Mbps en ambas direcciones. Para llegar a esta velocidad se utiliza una tasa de señalización de 800 Msímbolos/seg, con símbolos que usan 16 niveles de voltaje. Para producir los símbolos se mezclan los datos, se protegen con un código LDPC (Verificación de Paridad de Baja Densidad) y se vuelven a codificar para corrección de errores. A finales de 2007, el IEEE creó un grupo para estandarizar la Ethernet que opera a 40 Gbps y 100 Gbps. Esta actualización permitirá a Ethernet competir en ambientes de muy alto rendimiento, incluyendo las conexiones de larga distancia en redes troncales y las conexiones cortas a través de los planos posteriores de los equipos. El estándar todavía no está completo, pero ya hay productos propietarios disponibles.

## Tema IV

### Capa Enlace de Datos

La capa de enlace se enfoca en los algoritmos para lograr una comunicación confiable y eficiente de unidades completas de información llamadas tramas (en vez de bits individuales, como en la capa física) entre dos máquinas conectadas. La capa de enlace solo se ocupa de equipos física y directamente conectados, sin tener conocimiento o “conciencia” de la red en su conjunto. Esto no quiere decir que no pueda haber ningún dispositivo en el cable que conecta los dos equipos.

La capa de enlace de datos utiliza los servicios de la capa física para enviar y recibir bits a través de los canales de comunicación. Una característica importante de la capa de enlace es que los bits han de llegar a su destino en el mismo orden en que han salido; en algunos casos puede haber errores o pérdida de bits, pero nunca debe producirse una reordenación en el camino. Las principales funciones que desarrolla la capa de enlace son las siguientes:

- Agrupar los bits en grupos discretos denominados tramas. Esto permite desarrollar de forma más eficiente el resto de funciones.
- Realizar la comprobación de errores mediante el código elegido, que puede ser corrector o simplemente detector. En el caso de código corrector se procede a corregir los errores, en el de un código detector la trama errónea se descarta y opcionalmente se pide retransmisión al emisor.
- Efectuar control de flujo, es decir, pedir al emisor que baje el ritmo o deje momentáneamente de transmitir porque el receptor no es capaz de asimilar la información enviada.

No todas las funciones se implementan en los protocolos de enlace. La retransmisión de tramas erróneas y el control de flujo a menudo se implementan en las superiores (capa de red, de transporte, o incluso en la de aplicación). La mayoría de las funciones del nivel de enlace se implementan en el hardware de los equipos. Esto hace que los protocolos de nivel de enlace se modifiquen poco con el tiempo.

La función de la capa de enlace de datos es proveer servicios a la capa de red. Los tipos de servicio que la capa de enlace puede suministrar a la capa de red son los siguientes:

- Servicio no orientado a conexión y sin acuse de recibo
- Servicio no orientado a conexión con acuse de recibo
- Servicio orientado a conexión con acuse de recibo

En el primer caso el envío se hace sin esperar ninguna indicación del receptor sobre el éxito o fracaso de la operación. Este tipo de servicio es apropiado cuando la tasa de error es muy baja (redes locales o fibra óptica) y se deja la misión de comprobar la corrección de los datos transmitidos a las capas superiores (normalmente el nivel de transporte); se considera en estos casos que la probabilidad de error es tan baja que se pierde más tiempo haciendo comprobaciones inútiles que dejando esta tarea a las capas superiores. También se usa este tipo servicio cuando se quiere transmitir información en tiempo real (por ejemplo en una videoconferencia) y no se quiere sufrir el retraso que impondría un servicio más sofisticado en la capa de enlace (se supone que en este caso se prefiere la pequeña tasa de error del medio físico a cambio de minimizar el retardo, o dicho de otro modo si se hiciera reenvío en caso de error sería peor el remedio que la enfermedad).

En el segundo tipo de servicio se produce un acuse de recibo para cada trama enviada. De esta manera el emisor puede estar seguro de que ha llegado. Suele utilizarse en redes con mayor tasa de error, por ejemplo redes inalámbricas.

El tercer servicio es el más seguro y sofisticado. El emisor y el receptor establecen una conexión explícita de antemano, las tramas a enviar se numeran y se aseguran ambos de que son recibidas todas correctamente en su destino y transmitidas a la capa de red una vez y sólo una. En el servicio orientado a conexión se pueden distinguir tres fases: establecimiento de la conexión, envío de los datos, y terminación de la conexión. En la primera se establecen los

contadores y buffers necesarios para la transmisión, en la segunda se envían los datos con las retransmisiones que sea preciso, y en la tercera se liberan los buffers y variables utilizadas.

#### 4.1. ENTRAMADO (FRAMING)

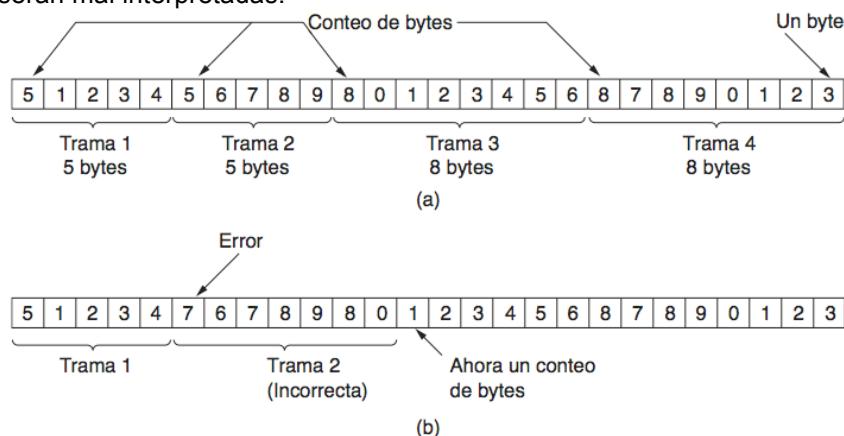
La capa de enlace agrupa los bits en paquetes discretos denominados tramas o marcos (frames) que son los que envía por la línea. Según el tipo de red la trama puede oscilar entre unos pocos y unos miles de bytes. La utilización de tramas simplifica el proceso de detección y eventual corrección de errores. Una buena parte de las tareas de la capa de enlace tiene que ver con la construcción e identificación de las tramas.

Para identificar el principio y final de una trama la capa de enlace puede usar varias técnicas, entre las cuales se tienen:

- Contador de bytes
- Bytes indicadores de inicio y final con bytes de relleno (Framing basado en caracteres)
- Bits indicadores de inicio y final, con bits de relleno (Framing basado en bits)
- Violaciones de codificación de la capa física

##### 4.1.1. Contador de bytes.

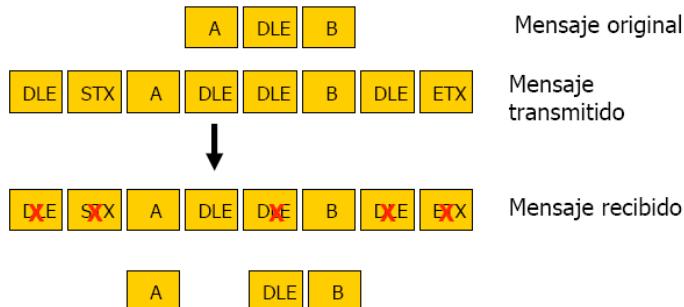
Se utiliza un campo en la cabecera de la trama para indicar el número de bytes (caracteres) de ésta. Tiene un serio problema: si un error afecta precisamente a la parte de la trama que indica la longitud, o si por un error en la línea se envían bits de más o de menos, todas las tramas posteriores serán mal interpretadas.



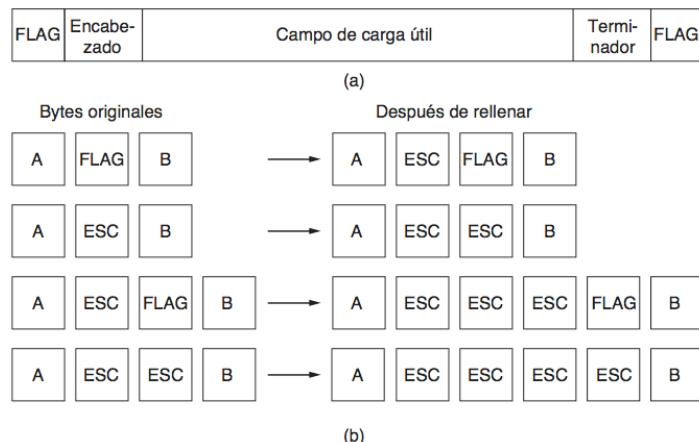
##### 4.1.2. Bytes de inicio y final.

Utiliza una secuencia especial de caracteres para marcar el inicio y final de cada trama (FLAG), normalmente los caracteres ASCII DLE STX para el inicio y DLE ETX para el final (DLE es Data Link Escape, STX es Start of Text y ETX End of Text). De esta forma si ocurre un error o incidente grave el receptor sólo tiene que esperar a la siguiente secuencia DLE STX o DLE ETX para saber en qué punto se encuentra.

Cuando se usa este sistema para transmitir ficheros binarios es posible que por puro azar aparezcan en el fichero secuencias DLE STX o DLE ETX, lo cual provocaría la interpretación incorrecta de un principio o final de trama por parte del receptor. Para evitar esto se utiliza una técnica conocida como *relleno de caracteres* (byte stuffing), el emisor cuando ve que ha de transmitir un carácter DLE proveniente de la capa de red intercala en la trama otro carácter DLE; el receptor, cuando recibe dos DLE seguidos, ya sabe que ha de quitar un DLE y pasar el otro a la capa de red.

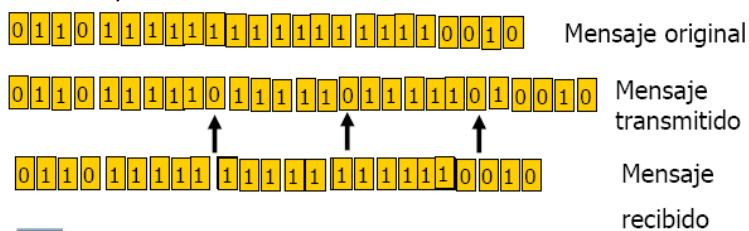


El principal problema que tiene el uso de DLE STX y DLE ETX es su dependencia del código de caracteres ASCII. Este método no resulta adecuado para transmitir otros códigos, especialmente cuando la longitud de carácter no es de 8 bits. Tampoco es posible enviar con este sistema tramas cuyo tamaño no sea múltiplo de ocho bits.



#### 4.1.3. Bits de inicio y final.

Es una técnica que se puede considerar una generalización de la anterior, consistente en utilizar una determinada secuencia de bits para indicar el inicio de una trama. Generalmente se utiliza para este fin la secuencia de bits 01111110 o 0x7E en hexadecimal, que se conoce como byte indicador (flag byte o flag pattern). El receptor está permanentemente analizando la trama que recibe buscando en ella la presencia de un flag byte, y en cuanto lo detecta sabe que ha ocurrido un inicio (o final) de trama. Aunque el flag byte tiene ocho bits el receptor no realiza el análisis byte a byte sino bit a bit, es decir la secuencia 01111110 podría suceder entre dos bytes y el receptor la interpretaría como flag byte; esto permite el envío de tramas cuya longitud no sea múltiplo de ocho.



Queda por resolver el problema de que los datos a transmitir contengan en sí mismos la secuencia 01111110; en este caso se utiliza la técnica conocida como relleno de bits o inserción de bit cero (bit stuffing o zero bit insertion). Consiste en que el emisor, en cuanto detecta que el flujo de bits contiene cinco bits contiguos con valor 1, inserta automáticamente un bit con valor 0. El receptor por su parte realiza la función inversa: analiza el flujo de bits entrante y en cuanto detecta un 0 después de cinco unos contiguos lo suprime en la reconstrucción de la trama recibida. De esta forma la secuencia 01111110 no puede nunca aparecer como parte de los datos transmitidos más que como delimitador de tramas. Si las cosas van mal y el receptor pierde noción de donde se encuentra bastará con que se ponga a la escucha de la secuencia 01111110 que le indicará el inicio o final de una trama.

#### 4.1.4. Violación de código.

Se utiliza en determinados tipos de red local aprovechando el hecho de que determinadas secuencias de símbolos no están permitidas y por tanto no pueden ocurrir en los datos a transmitir. Este método está muy relacionado con las características del medio físico utilizado. Por ejemplo: Si se utiliza el código Manchester en un intervalo de bit se utilizan las secuencias +V -V ó -V +V, pero no +V +V ni -V -V, estas se utilizarán para identificar el principio y final de trama.

### 4.2. CONTROL DE FLUJO

Cuando dos computadores se comunican generalmente han de adoptarse medidas para asegurar que el emisor no sature al receptor. Si la línea entre ellos es de baja capacidad probablemente el factor limitante será la conexión, pero si es un canal rápido (por ejemplo una red local) es posible que el emisor, si es un computador más rápido o está menos cargado que el receptor, envíe datos a un ritmo superior al que sea capaz de asimilar éste. En este caso el nivel de enlace en el receptor utilizará los buffers que tenga disponibles para intentar no perder datos, pero si el ritmo acelerado sigue durante un tiempo suficiente se producirá antes o después una pérdida de tramas por desbordamiento. En estos casos es preciso habilitar mecanismos que permitan al receptor frenar al emisor, es decir ejercer **control de flujo** sobre él. El control de flujo puede implementarse en el nivel de enlace o en niveles superiores (nivel de transporte). Es importante que el control de flujo se ejerza de forma que no produzca ineficiencias en la comunicación; por ejemplo en enlaces de área extensa, donde la capacidad es un bien muy costoso, es importante mantener el nivel de ocupación del enlace tan alto como sea posible sin incurrir por ello en pérdida de tramas.

### 4.3. CONTROL DE ERRORES

El medio de transmisión utilizado en redes de computadores introduce errores. La tasa de errores es función de múltiples factores, pero principalmente del medio de transmisión utilizado. La fibra óptica y las redes locales suelen tener las tasas más bajas<sup>1</sup>, mientras que las transmisiones inalámbricas con equipos móviles (GSM o LANs inalámbricas) o sobre telefonía analógica suelen tener las más altas.

La disciplina que estudia los errores de transmisión desde el punto de vista matemático es la teoría de la codificación. En 1950 R. W. Hamming publicó un artículo donde establecía las bases de los códigos de detección y corrección de errores.

La trama que se transmite de un computador a otro está formada por  $m$  bits de datos y  $r$  bits redundantes, de comprobación. La trama tiene pues una longitud  $n = m + r$ , y forma lo que en teoría de la codificación se denomina una **palabra codificada** o **codeword** de  $n$  bits. Dados dos codewords cualesquiera, por ejemplo 10001001 y 10110001 es fácil determinar en cuantos bits difieren aplicando la operación OR exclusivo entre ambas y contando el número de bits a 1 del resultado; por ejemplo en nuestro caso difieren en 3 bits. Este valor, el número de posiciones de bit en que dos codewords difieren, se denomina **distancia de Hamming**. Si dos codewords están separadas por una distancia  $d$  serán necesarias  $d$  conversiones de un bit (por ejemplo  $d$  errores de un bit) para transformar una en la otra.

En la transmisión de información generalmente los datos de usuario pueden adoptar cualquier valor, por lo que en la parte de datos de la trama hay  $2^m$  valores posibles; pero debido a la manera como se calculan los  $r$  bits de comprobación no están permitidas las  $2^r$  codewords que en principio podrían formar la trama.

La distancia Hamming de un código determina su capacidad de detección y corrección de errores. Para detectar  $d$  errores (es decir,  $d$  bits erróneos en la misma trama) es preciso que la distancia sea como mínimo de  $d + 1$ ; de esa manera la codeword errónea no coincidirá con ninguna otra codeword válida y el receptor puede detectar la anomalía. Si se quiere un código capaz de corregir  $d$  errores es preciso que la distancia Hamming sea como mínimo  $2d + 1$ , ya que entonces la codeword errónea recibida sigue estando más cerca de la codeword original

<sup>1</sup> SONET/SDH requieren en fibra óptica una tasa de error o BER (Bit Error Rate) de  $10^{-12}$ , es decir un bit erróneo cada  $10^{12}$  bits transmitidos.

que de cualquier otra. Así por ejemplo, si la distancia Hamming del código utilizado en la corrección de errores de un protocolo determinado es de 5, entonces el protocolo podrá corregir hasta 2 errores en una trama, y detectar hasta 4.

Los códigos de corrección de errores se denominan también corrección de errores hacia adelante o FEC (Forward Error Control) y los de detección se llaman códigos de corrección de errores hacia atrás o por realimentación (feedback o backward error control).

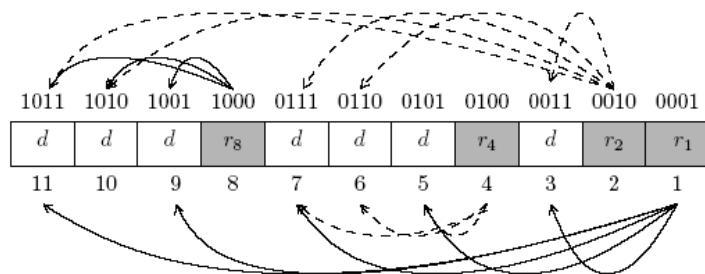
#### 4.4. CÓDIGOS CORRECTORES DE ERRORES

Los códigos de corrección de errores siempre tienen una eficiencia menor que los de detección para el mismo número de bits, y salvo que el medio de transmisión tenga muchos errores no salen rentables.

##### 4.4.1. Código de Hamming

Hamming ideó un sistema de codificación que permite recuperar un número de errores de transmisión por palabra transmitida arbitrariamente grande, aunque por motivos de simplicidad se presenta el código que permite recuperar solo un bit erróneo.

Suponer que el tamaño de los símbolos originales es de 7 bits. El código de Hamming inserta en las posiciones que son potencias de dos un bit de paridad. Por ejemplo, el bit que está en la posición 1 (que tiene un único bit distinto de cero y que es el bit de menos peso) es el bit de paridad de los bits situados en las posiciones impares, porque todas ellas tienen el bit menos significativo igual a 1.

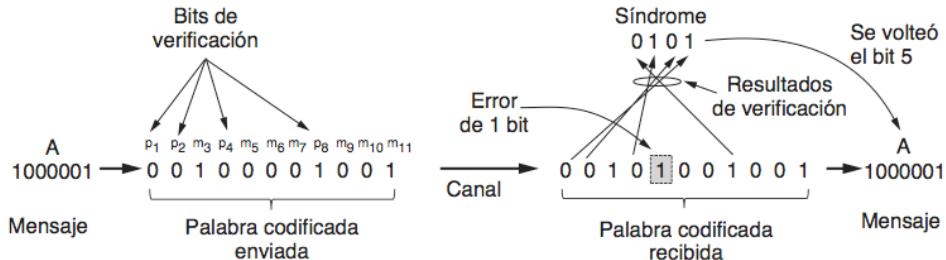


Inserción de los bits de paridad en el código de Hamming. Los puntos seleccionados se corresponden con aquellos cuyo índice sólo tiene un bit a 1 y por tanto son potencias de 2.

Se puede empezar de derecha a izquierda como en el gráfico anterior o de izquierda a derecha, como en el siguiente gráfico y la inserción se realiza de acuerdo al valor de la potencia de 2.

Posición del bit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
bits codificados	p1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7	d8	d9	d10	d11	p16	d12	d13	d14	d15	
bits de paridad	p1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
	p2	X	X		X	X			X	X			X	X			X	X			
	p4			X	X	X	X				X	X	X	X						X	
	p8							X	X	X	X	X	X	X	X						
	p16															X	X	X	X	X	

Para conocer si se ha producido un error de transmisión, el receptor recalcula los bits de paridad, de acuerdo al mismo mecanismo de paridad y dirección. Si ningún error se ha producido, los bits de paridad deben coincidir con los que viajan con los datos. Si ha ocurrido un error, los bits de paridad que cambian indican dónde se ha producido el error.



#### 4.4.2. Código de Reed-Solomon.

A diferencia de los códigos de Hamming, que operan sobre bits individuales, los códigos de Reed-Solomon operan sobre símbolos de  $m$  bits. Naturalmente las matemáticas son más complejas, por lo que se describirá su operación mediante una analogía.

Los códigos de Reed-Solomon se basan en el hecho de que todo polinomio de  $n$  grados se determina de forma única mediante  $n + 1$  puntos. Por ejemplo, una línea con la forma  $ax + b$  se determina mediante dos puntos. Los puntos extra en la misma línea son redundantes, lo cual es útil para la corrección de errores. Si se tiene dos puntos de datos que representan una línea y se envía esos dos puntos de datos junto con dos puntos de verificación seleccionados sobre la misma línea. Si uno de los puntos se recibe con error, de todas formas se puede recuperar los puntos de datos si se ajusta una línea a los puntos recibidos. Tres de los puntos estarán en la línea y el otro punto (el del error) no. Al encontrar la línea se corrige el error.

En realidad los códigos de Reed-Solomon se definen como polinomios que operan sobre campos finitos, pero trabajan de una manera similar. Para símbolos de  $m$  bits, las palabras codificadas son de  $2^m - 1$  símbolos de longitud. Una elección popular es hacer a  $m = 8$ , de modo que los símbolos sean bytes. Así, una palabra codificada tiene una longitud de 256 bytes.

Los códigos de Reed-Solomon se utilizan mucho en la práctica debido a sus poderosas propiedades de corrección de errores, en especial para los errores de ráfagas. Se utilizan para DSL, datos sobre cable, comunicaciones de satélite, en los CD, DVD y discos Blu-ray. Puesto que se basan en símbolos de  $m$  bits, tanto un error de un solo bit como un error de ráfaga de  $m$  bits se tratan simplemente como error de un símbolo. Cuando se agregan  $2t$  símbolos redundantes, un código de Reed-Solomon es capaz de corregir hasta  $t$  errores en cualquiera de los símbolos transmitidos. Esto significa que, por ejemplo, el código (255, 223) que tiene 32 símbolos redundantes puede corregir errores de hasta 16 símbolos. Como los símbolos pueden ser consecutivos y cada uno de ellos es de 8 bits, se puede corregir una ráfaga de errores de hasta 128 bits. La situación es aún mejor si el modelo de error es el de borrado (por ejemplo, una rayadura en un CD que borra algunos símbolos). En este caso se pueden corregir hasta  $2t$  errores. A menudo los códigos de Reed-Solomon se utilizan en combinación con otros códigos, como el convolucional. Los códigos convolucionales son efectivos a la hora de manejar errores de bits aislados, pero es probable que fallen con una ráfaga de errores, si hay demasiados errores en el flujo de bits recibido. Al agregar un código de Reed-Solomon dentro del código convolucional, la decodificación de Reed-Solomon puede limpiar las ráfagas de errores, una tarea que realiza con mucha eficiencia. Así, el código en general provee una buena protección contra los errores individuales y los errores de ráfaga.

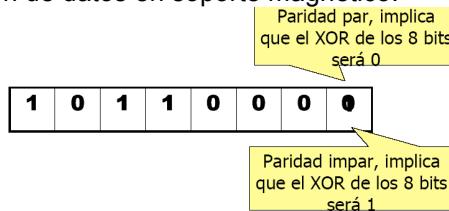
### 4.5. CÓDIGOS DETECTORES DE ERRORES

Un código de detección de errores permite al receptor conocer (con suficiente confiabilidad) si los datos que han sido recibidos han sufrido algún error durante su transmisión, aunque no permite restaurar los datos. La idea es añadir cierta cantidad de información redundante para que el receptor pueda conocer este hecho. En general, cuanto mayor es dicha cantidad de información, mayor es el número de errores que pueden ser detectados.

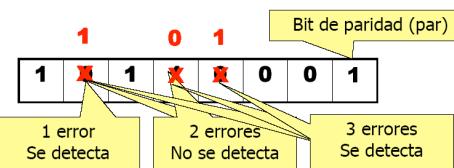
#### 4.5.1. Paridad y paridad Bidimensional

El bit de paridad se elige de forma que mantenga la paridad (par o impar) de la codeword. El código formado con un bit de paridad tiene una distancia de 2, ya que cambiando un bit de cualquier codeword el resultado es ilegal, pero cambiando dos vuelve a serlo. Con una distancia 2 es posible detectar errores de 1 bit, pero no es posible detectar errores múltiples, ni corregir errores de ningún tipo. A cambio tiene un overhead mínimo, ya que supone añadir

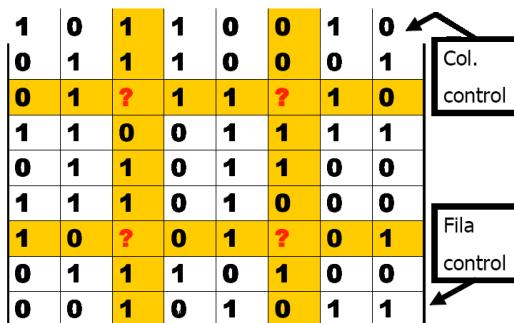
solamente un bit a cada codeword. Por este motivo el bit de paridad se utiliza en situaciones donde la fiabilidad es muy alta y la codeword muy pequeña, como en algunos sistemas de memoria RAM o de grabación de datos en soporte magnético.



La **eficiencia** de un código viene dada por la relación  $m/n$ ; a la diferencia  $1-m/n$  se la denomina **redundancia**. Por ejemplo al utilizar un bit de paridad para acceder a un byte de memoria se tiene una eficiencia de 0,8889 y una redundancia de 0,1111.



En esencia cualquier mecanismo de control de errores se basa en la inclusión de un cierto grado de redundancia, lo cual requiere un compromiso entre eficiencia y fiabilidad. Si por ejemplo se desea transmitir una trama de 64 bits y se utiliza un bit de paridad; la eficiencia se reducirá en un 2% solamente y se podrá detectar errores simples, pero los errores dobles pasarán desapercibidos; en caso de errores múltiples la probabilidad de que pasen desapercibidos es de 0,5, lo cual no es aceptable. Para mejorar la fiabilidad se puede introducir un bit de paridad cada ocho bits de datos; para esto se configura la trama como una matriz de 8 x 8 bits, a la cual se añade una novena columna que son los bits de paridad; la transmisión se haría fila a fila, sin incluir la novena columna (los bits de paridad) que iría al final de la trama; el receptor reconstruiría la matriz 8 x 8 a partir de los bits recibidos y a continuación construiría la columna de bits de paridad, que luego compararía con la recibida; en caso de discrepancia se supondría un error y se pediría retransmisión.



Con este sistema las probabilidades de detectar errores múltiples han aumentado, pero si los errores se producen a ráfagas (cosa muy normal en transmisión inalámbrica, por ejemplo) se tendrán varios bits erróneos cerca, con lo que la probabilidad de que su bit de paridad detecte el error vuelve a ser de 0,5. Ahora bien, si en vez de calcular el bit de paridad para cada fila se lo hace para cada columna se tomaran bits no contiguos de la trama, con lo que la probabilidad de que un error a ráfagas pase desapercibido es mucho menor; se tendría que fallar dos bits de una misma columna y no fallar ningún otro bit, por ejemplo. Si estadísticamente la probabilidad de que un error en una columna pase desapercibido es de 0,5, la de que esta situación se dé en las 8 columnas es de  $0,5^8$ , es decir, 0,0039. En este caso una elección inteligente de los bits de paridad ha permitido aumentar considerablemente la fiabilidad del código sin reducir su eficiencia.

El objetivo esencial de los códigos de detección o corrección de errores consiste en optimizar los algoritmos de cálculo de los bits de control para que sean capaces de detectar el máximo número de errores posible con un número razonable de bits adicionales. Las técnicas matemáticas en que se basan estos algoritmos han sido objeto de exhaustivos estudios por

parte de especialistas en teoría de la codificación, y no son fácilmente mejorables; como consecuencia de esto los algoritmos de detección y corrección de errores son una parte bastante estable dentro de los sistemas de transmisión de datos.

#### 4.5.2. Comprobación de Redundancia Cíclica (CRC)

El algoritmo de detección de errores más utilizado en la práctica se basa en lo que se conoce como *códigos polinómicos* (*comprobación de redundancia cíclica* o *CRC*, *Cyclic Redundancy Check*). La idea básica es la misma que en el caso de los bits de paridad: añadir a los datos a transmitir unos bits adicionales cuyo valor se calcula a partir de los datos; la trama así construida se envía, y el receptor separa los bits de datos de la parte CRC; a partir de los datos recalcula el CRC y compara con el valor recibido; si ambos no coinciden se supone que ha habido un error y se pide retransmisión.

La aritmética polinómica tiene unas propiedades singulares que la hacen especialmente fácil de programar en sistemas digitales, por lo que es posible implementarla directamente en hardware con lo que se consigue una eficiencia elevada, cosa importante para evitar que la comunicación se ralentice por el cálculo del CRC. Algunas de estas propiedades son:

Dada la siguiente suma de polinomios:

$$\begin{array}{r}
 X^7 + X^4 + X^3 + X + 1 \\
 + \quad X^7 + X^6 + X^3 + X \\
 \hline
 X^6 + X^4 + 1
 \end{array}
 \quad \begin{array}{l}
 \text{que equivale a:} \\
 \text{que equivale a:} \\
 \text{que equivale a:}
 \end{array}
 \quad \begin{array}{l}
 10011011 \\
 11001010 \\
 01010001
 \end{array}$$

Se observa como no se arrastra valor a la unidad superior. En la práctica el resultado de la suma es equivalente a haber efectuado un XOR (or exclusivo) bit a bit entre las dos cadenas.

En el caso de la resta la situación es idéntica:

$$\begin{array}{r}
 X^6 + X^4 + X^2 + 1 \\
 - \quad X^7 + X^5 + X^3 + X^2 + X + 1 \\
 \hline
 X^7 + X^6 + X^5 + X^4 + X^3 + X
 \end{array}
 \quad \begin{array}{l}
 01010101 \\
 10101111 \\
 11111010
 \end{array}$$

ya que al utilizar módulo 2 la operación de dos valores iguales siempre da 0 y dos diferentes da 1. En el caso de la división la operación se hace como en binario con la única peculiaridad de que la resta se hace módulo 2, como se acaba de ver. Se ejemplifica paso a paso como se utiliza todo esto en una transmisión de datos:

1. En primer lugar el emisor y el receptor acuerdan un generador polinómico común  $G(x)$ , por ejemplo  $x^4 + x + 1$  (10011); el primero y último bits de un generador polinómico siempre deben ser 1. El CRC siempre tiene una longitud un bit menos que el generador polinómico utilizado, por lo que en nuestro caso será de 4 bits (grado del polinomio)..
2. Si el emisor desea transmitir la cadena  $c_1$ , formada por los bits 1101011011, que sería un polinomio de grado 9 (los datos a transmitir siempre deben tener más bits que el generador polinómico utilizado). El emisor añade cuatro bits (en principio a 0) al final de los datos a transmitir, formando la cadena  $c_2$  110010110110000; esto equivale a multiplicar la cadena por  $2^4$
3. El emisor divide la cadena  $c_2$  por el generador polinómico acordado (10011) usando las reglas de división binaria módulo 2 mencionado, y calcula el resto  $r$ , que es en este caso 1110.
4. El emisor resta el resto  $r$  de la cadena  $c_2$ , formando así la cadena  $c_3$  11010110111110. La resta es una operación XOR sobre los cuatro últimos bits, en la práctica la resta se hace sencillamente sustituyendo los cuatro últimos bits de  $c_2$  por  $r$ . Al restar al dividendo el resto el valor obtenido es divisible por  $g$ .
5. La cadena  $c_3$  es transmitida al receptor.
6. El receptor recibe la cadena  $c_3$  y la divide por  $g$ . Si el resultado no es cero la transmisión se considera errónea y se solicita retransmisión.

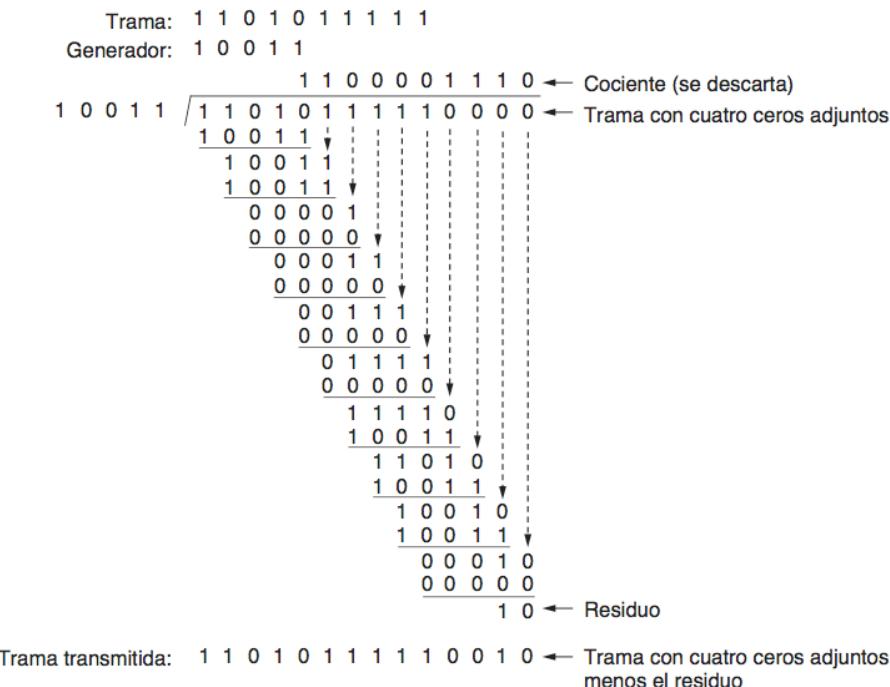
Este algoritmo, que en principio puede parecer extraño y arbitrario, tiene tres características interesantes:

- Da un resultado predecible y reproducible, por lo que aplicado sobre unos mismos datos siempre dará el mismo resultado,
- Las operaciones utilizadas (desplazamiento, XOR, etc.) lo hacen muy fácil de implementar en hardware, y
- Suministra un mecanismo extremadamente flexible y robusto para la detección de errores, a través de la elección del generador polinómico adecuado.

Los generadores polinómicos más utilizados forman parte de estándares internacionales, y son los siguientes:

$$\begin{array}{ll}
 \text{CRC-12:} & x^{12} + x^{11} + x^3 + x^2 + x + 1 \\
 \text{CRC-16:} & x^{16} + x^{15} + x^2 + 1 \\
 \text{CRC-CCITT:} & x^{16} + x^{12} + x^5 + 1 \\
 \text{CRC-32:} & x^{32} + x^{26} + x^{23} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1
 \end{array}$$

CRC-12 se utiliza en los códigos con longitud de carácter de 6 bits; CRC-16 y CRC-CCITT se utilizan en conexiones WAN, mientras que CRC-32 se utiliza en conexiones LAN. En todos los generadores utilizados aparece  $x+1$  como factor, ya que esto asegura la detección de todos los errores con un número impar de bits. Un código polinómico de  $r$  bits detectará todos los errores a ráfagas de longitud  $\leq r$ . Un generador como CRC-16 o CRC-CCITT detectará todos los errores simples y dobles, todos los errores con un número impar de bits, todos los errores a ráfagas de longitud 16 o menor, 99,997% de los errores a ráfagas de 17 bits y 99,998% de los errores a ráfagas de 18 o más bits.



Cabría pensar en la posibilidad de que un error alterara la trama de tal forma que el CRC de la trama errónea coincidiera con el de la trama correcta (después de todo la probabilidad de que una trama distinta tenga el mismo CRC es 1/65536 para CRCs de 16 bits); los algoritmos de cálculo de CRCs intentan conseguir que las otras posibles tramas con igual CRC se encuentren muy alejadas (en términos de distancia Hamming) por lo que tendría que producirse una gran cantidad de errores en la misma trama para que esta posibilidad pudiera darse. En todo caso, cualquier protocolo de nivel de enlace fallará si se produce un error que no pueda ser detectado por el CRC.

## 4.6. PROTOCOLOS DE ENLACE ELEMENTALES

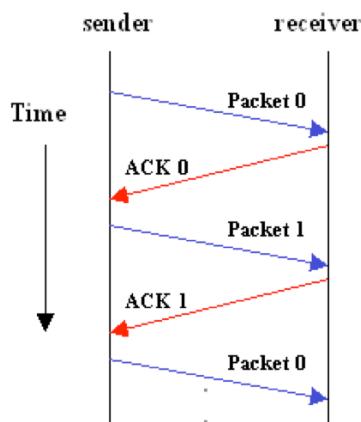
La principal característica que diferencia los protocolos de nivel de enlace es su comportamiento frente a los errores. Cuando el receptor detecta una trama errónea puede hacer una de las dos cosas siguientes:

- Descartar silenciosamente la trama errónea sin notificarlo a nadie.
- Solicitar del emisor la retransmisión de la trama errónea.

En el primer caso, es decir cuando no se realiza retransmisión de tramas erróneas el protocolo de enlace es trivial, por lo que normalmente esta opción casi no se comenta al hablar del nivel de enlace y se le dedica poco o ningún espacio en los libros de texto. En cambio se suelen explicar con todo detalle las diversas variantes de protocolos de enlace con retransmisión. Esto provoca lógicamente que al hablar de protocolos a nivel de enlace casi siempre se piense exclusivamente en los que realizan retransmisión de tramas erróneas. Dada la elevada fiabilidad de la mayoría de los medios físicos actuales normalmente no es rentable solicitar comprobación y retransmisión de las tramas, ya que esto supondría realizar un proceso casi siempre inútil en cada nodo del trayecto. Será normalmente el protocolo de transporte el que se ocupe de solicitar la retransmisión en caso de error. En caso de error la información habrá viajado inútilmente hasta el host de destino, pero esta estrategia es más rentable cuando la tasa de errores es baja. En los casos en que la tasa de errores del medio físico es excesiva se prefiere incorporar en el nivel físico un mecanismo corrector de errores, lo cual se traduce en la práctica en un canal prácticamente libre de errores al nivel de enlace; esto es lo que ocurre por ejemplo en las comunicaciones por red conmutada vía módem gracias al estándar V.42, en las comunicaciones a través de redes GSM o en las transmisiones de televisión digital con el uso del código Reed Solomon.

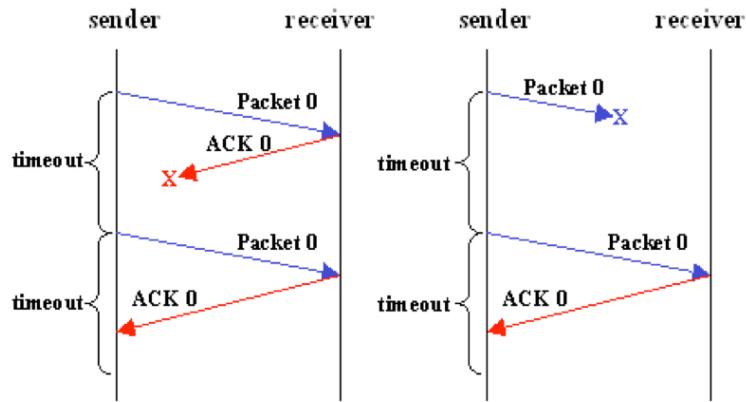
### 4.6.1. Protocolo de parada y espera (Stop & Wait)

Como caso más sencillo de protocolo con retransmisión se tiene el denominado de parada y espera, consistente en que el emisor espera confirmación o acuse de recibo después de cada envío y antes de efectuar el siguiente. El acuse de recibo, también llamado ACK (acknowledgement) sirve tanto para indicar que la trama ha llegado correctamente como para indicar que se está en condiciones de recibir la siguiente, es decir el protocolo incorpora también la función de control de flujo. Este tipo de protocolos donde el emisor espera una confirmación o acuse de recibo para cada dato enviado se denominan protocolos PAR (Positive Acknowledgement with Retransmission) o también ARQ (Automatic Repeat reQuest).



Cuando la trama recibida es errónea (cosa que el receptor podrá verificar gracias al CRC) no se produce ACK. Lo mismo sucede cuando la trama enviada se pierde por completo. En este caso el emisor, pasado un tiempo máximo de espera, reenvía la trama. Una optimización que se puede incorporar en el protocolo es el uso de acuse de recibo negativo o NAK (Negative Acknowledgement) cuando se recibe una trama errónea; de esta forma el emisor puede reenviar la trama sin esperar a agotar el tiempo de espera, con lo que se consigue una mayor utilización de la línea.

Si en una de las veces, lo que se pierde no es la trama enviada sino el mensaje de ACK; pasado el tiempo de espera el emisor concluirá erróneamente que la trama se ha perdido y la reenviará, llegando esta duplicada al receptor; el receptor no tiene ningún mecanismo para detectar que la trama es un duplicado, por lo que pasará el duplicado al nivel de red, lo cual no está permitido en un protocolo de enlace. Una forma de que el receptor distinga los duplicados es numerar las tramas, por ejemplo con un campo de un bit se puede numerar las tramas en base 2 (0, 1, 0, 1, ...) que es suficiente para detectar los duplicados.



Los protocolos de parada y espera son sencillos de implementar, pero son poco eficientes. Por ejemplo si se utiliza una línea de 64 Kb/s para enviar tramas de 640 bits de un computador A a otro B que se encuentra a una distancia de 2.000 Km. A tarda 10 ms en emitir cada trama (640/64000) o, dicho de otro modo, transmite 64 bits cada milisegundo. Por otro lado los bits tardan 10 ms en llegar de A a B (la velocidad de las ondas electromagnéticas en materiales es de unos 200.000 Km/s); justo cuando llega a B el primer bit de la primera trama A termina de emitirla (en ese momento la trama está "en el cable"); diez milisegundos más tarde B recibe la trama en su totalidad, verifica el CRC y devuelve el ACK; suponiendo que el tiempo que tarda B en verificar el CRC y generar el ACK es despreciable la secuencia de acontecimientos es la siguiente:

Instante (ms)	Suceso en A	Suceso en B
0 ms	Emite primer bit de trama 1	Espera
10 ms	Emite último bit de trama 1; espera	Recibe primer bit de trama 1
20 ms	Espera	Recibe último bit de trama 1; envía
ACK		
30 ms	Recibe ACK; emite primer bit de trama 2	Espera
...	...	...

A partir de aquí el ciclo se repite. De cada 30 ms se está transmitiendo 10 ms y esperando 20 ms, es decir se está utilizando la línea con una eficiencia de  $10/30 = 0,33 = 33\%$ .

La eficiencia obtenida depende de tres parámetros: la velocidad de la línea,  $v$ , el tamaño de trama,  $t$ , y el tiempo de ida y vuelta también llamado "round trip time"; en el caso normal de que el tiempo de ida y el de vuelta son iguales se define  $\tau$  como el tiempo de ida, por lo que el tiempo de ida y vuelta es de  $2\tau$ . Se puede derivar una expresión que permite calcular la eficiencia a partir de estos valores:

$$\text{Eficiencia} = \frac{t/v}{(t/v) + 2\tau}$$

Que aplicada al ejemplo anterior resulta:

$$\text{Eficiencia} = (640/64000) / (640/64000 + 2*0,01) = 0,01 / (0,01 + 0,02) = 0,01 / 0,03 = 0,33$$

Es evidente que los protocolos de parada y espera tienen una baja eficiencia en algunos casos. El caso extremo de ineficiencia se da cuando se utilizan enlaces vía satélite, en los que el valor de  $2\tau$  puede llegar a ser de medio segundo.

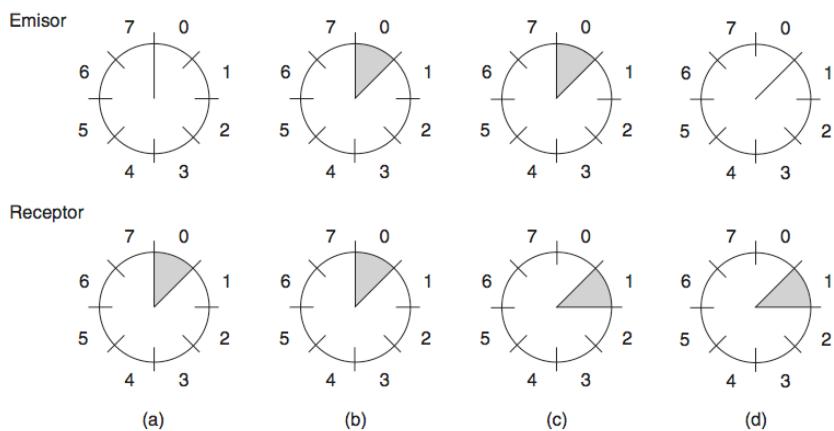
El protocolo de parada y espera visto transmite datos en una sola dirección; el canal de retorno es utilizado únicamente para enviar los mensajes de acuse de recibo (ACK). Si se tiene que transmitir datos en ambas direcciones se puede utilizar dos canales semi-dúplex con los protocolos anteriores, enviando en cada sentido tramas de datos mezcladas con tramas ACK. La trama ACK contiene una cantidad mínima de información útil, pero ha de contener no obstante una serie de campos de control imprescindibles que ocupan más bits que la propia información de ACK. Si se están transmitiendo datos en ambas direcciones resulta más eficiente, en vez de enviar el ACK solo en una trama, enviarlo dentro de una trama de datos; de esta forma el ACK viajará “casi gratis” y se ahorrará el envío de una trama. Esta técnica se conoce con el nombre de *piggybacking* o *piggyback acknowledgement*; (*piggyback* significa llevar a alguien o algo a hombros o a cuestas). Para enviar el ACK en una trama de datos es preciso que esta se envíe en un tiempo razonablemente corto respecto a cuando debería enviarse el ACK; de lo contrario el emisor, al ver que el ACK esperado no llega reenviará la trama, lo cual anularía el pretendido beneficio del piggybacking; como no es posible saber de antemano cuando se va a enviar la siguiente trama de datos generalmente se adopta una solución salomónica: se espera un determinado tiempo y si el nivel de red no genera ningún paquete en ese tiempo se genera una trama ACK; en este caso el tiempo de espera debe ser sensiblemente inferior al timer de reenvío del emisor.

#### 4.6.2. Protocolos de Ventana Deslizante

Para aprovechar mejor los enlaces con valores elevados del tiempo de ida y vuelta hacen falta protocolos que permitan crear un “pipeline”, o dicho de otro modo tener varias tramas “en ruta” por el canal de transmisión.

Al tener varias tramas simultáneamente pendientes de confirmación se necesita un mecanismo que permita referirse a cada una de ellas de manera no ambigua, ya que al recibir los ACK se debe saber a qué trama se refieren. Para ello se utiliza un número de secuencia; sin embargo como el número de secuencia va a aparecer en todas las tramas y los mensajes ACK interesa que sea lo menor posible; por ejemplo con un contador de 3 bits se puede numerar las tramas módulo 8 (0,1,2,...,7) con lo que es posible enviar hasta siete tramas (0....6) antes de recibir el primer ACK; a partir de ese punto se puede enviar una nueva trama por cada ACK recibido. Esto es lo que se denomina un protocolo de *ventana deslizante*.

Se puede imaginar el funcionamiento del protocolo de ventana deslizante antes descrito como un círculo dividido en ocho sectores de  $45^\circ$  cada uno, numerados del 0 al 7; sobre el círculo hay una ventana giratoria que permite ver los sectores correspondientes a las tramas pendientes de confirmación; la ventana puede abrirse como máximo  $315^\circ$ , es decir permite ver hasta siete sectores correspondientes a las tramas enviadas pendientes de confirmación. Cuando se recibe un ACK se envía otra trama y la ventana gira un sector.



Una ventana deslizante de tamaño 1, con un número de secuencia de 3 bits. (a) Al inicio. (b) Despues de enviar la primera trama. (c) Despues de recibir la primera trama. (d) Despues de recibir la primera confirmación de recepción.

Con un número de secuencia de  $n$  bits se puede tener como máximo una ventana de  $2^n - 1$  tramas, no de  $2^n$ . Por ejemplo, con un número de secuencia de tres bits el emisor puede enviar como máximo siete tramas sin esperar contestación. De esta forma se garantiza que el número de trama recibido en dos ACK sucesivos siempre será distinto y no habrá duda posible en la detección de duplicados que pudieran producirse por la expiración prematura de timers; si se utilizara una ventana de  $2^n$  se podrían producir conflictos.

El protocolo de parada y espera se puede considerar como un protocolo de ventana deslizante en el que se utiliza un bit para el número de secuencia; en este caso el círculo estaría formado por dos sectores de  $180^\circ$  cada uno, y la ventana tendría una apertura de  $180^\circ$ .

Suponiendo un retardo nulo en el envío de los bits y en el proceso de las tramas en los respectivos sistemas, así como una longitud nula de las tramas ACK, el tamaño de ventana mínimo necesario  $W$  para poder llenar un canal de comunicación puede calcularse con la fórmula:

$$W = 2\tau^*v/t + 1$$

Debiendo redondearse el valor al entero siguiente por encima. En la fórmula anterior  $2\tau$  es el tiempo (en segundos) que tarda una trama en hacer el viaje de ida y vuelta (round-trip time),  $v$  es la velocidad del canal de transmisión y  $t$  el tamaño de la trama a transmitir. Por ejemplo para el caso del ejemplo anterior donde  $2\tau = 0,02$ ,  $v = 64000$  y  $t = 640$  se obtiene  $W = 3$ , por tanto la ventana mínima es 3.

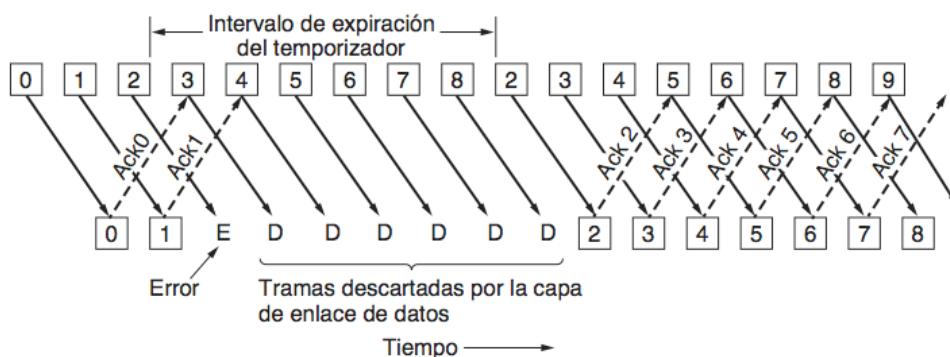
Cuando se utiliza un protocolo de ventana deslizante con ventana mayor que uno el emisor no actúa de forma sincronizada con el receptor; cuando el receptor detecta una trama defectuosa puede haber varias posteriores ya en camino, que llegarán irremediablemente a él, aún cuando reporte el problema inmediatamente. Existen dos posibles estrategias en este caso:

- El receptor ignora las tramas recibidas a partir de la errónea (inclusive) y solicita al emisor retransmisión de todas las tramas a partir de la errónea. Esta técnica se denomina *retroceso N*.
- El receptor descarta la trama errónea y pide retransmisión de ésta, pero acepta las tramas posteriores que hayan llegado correctamente. Esto se conoce como *repetición selectiva*.

#### 4.6.2.1. Protocolo de retroceso N (Go Back N)

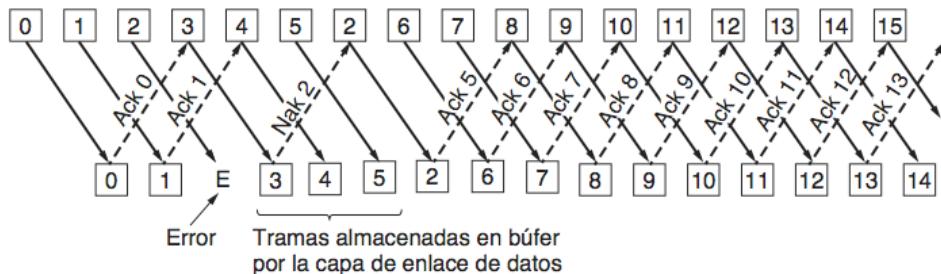
En retroceso N el receptor procesa las tramas en estricta secuencia, por lo que sólo necesita reservar espacio en buffers para una trama. En cambio en repetición selectiva el receptor ha de disponer de espacio en el buffer para almacenar todas las tramas de la ventana, ya que en caso de pedir retransmisión tendrá que intercalar en su sitio la trama retransmitida antes de pasar las siguientes a la capa de red (la capa de red debe recibir los paquetes estrictamente en orden).

En cualquiera de los dos casos el emisor deberá almacenar en su buffer todas las tramas que se encuentren dentro de la ventana, ya que en cualquier momento el receptor puede solicitar la retransmisión de alguna de ellas.



#### 4.6.2.2. Protocolo con repetición selectiva (Selective Repeat)

La repetición selectiva aprovecha las tramas correctas que llegan después de la errónea, y pide al emisor que retransmita únicamente esta trama. Como los paquetes se han de transferir en orden a la capa de red cuando falla una trama el receptor ha de conservar en buffers todos los paquetes posteriores hasta conseguir correctamente la que falta; en la práctica esto requiere tener un buffer lo suficientemente grande para almacenar un número de tramas igual al tamaño de la ventana, ya que se podría perder la primera trama de la ventana y recibirse correctamente el resto, en cuyo caso habría de conservarlas hasta recibir correctamente la primera.



La posibilidad de una recepción no secuencial de tramas plantea algunos problemas nuevos. Por ejemplo, suponer que con un número de secuencia de tres bits el emisor envía las tramas 0 a 6, las cuales son recibidas correctamente. Entonces el receptor realiza las siguientes acciones:

1. Las transmite a la capa de red,
2. Libera los buffers correspondientes
3. Avanza la ventana para poder recibir siete tramas más, cuyos números de secuencia podrán ser 7,0,1,2,3,4,5
4. Envía un ACK para las tramas 0 a 6 recibidas

Imaginar ahora que el ACK no llega al emisor. Éste supondrá que ninguna de las tramas ha llegado, por lo que las reenviará todas de nuevo (tramas 0 a 6). De estas las tramas 0 a 5 se encuentran dentro de la ventana del receptor y son por tanto aceptadas; la trama 6 está fuera de rango y es ignorada. En procesamiento secuencial el receptor no aceptaría estas tramas si no recibiera antes la trama 7 pendiente, pero con retransmisión selectiva las tramas fuera de orden se aceptan y se pide retransmisión de la trama 7; una vez recibida ésta se pasaría a la capa de red seguida de las tramas 0 a 5 antes recibidas, que serían duplicados de las anteriores. Los duplicados no detectados serían pasados al nivel de red, con lo que el protocolo es erróneo.

La solución a este conflicto está en evitar que un mismo número de secuencia pueda aparecer en dos ventanas consecutivas. Por ejemplo con un número de secuencia de 4 bits (0-15) y tamaño de ventana 8 la ventana del receptor sería inicialmente 0-7, después 8-15, 0-7 y así sucesivamente. Al no coincidir ningún número de secuencia entre ventanas contiguas se puede efectuar el proceso no secuencial de tramas sin que ocurra el conflicto anterior.

Como es lógico la técnica de repetición selectiva da lugar a protocolos más complejos que la de retroceso N, y requiere mayor espacio de buffers en el receptor. Sin embargo, cuando las líneas de transmisión tienen una tasa de errores elevada la repetición selectiva da un mejor rendimiento, ya que permite aprovechar todas las tramas correctamente transmitidas. La decisión de cuál utilizar se debería tomar valorando en cada caso la importancia de estos factores: complejidad, espacio en buffers, tasa de errores y eficiencia.

## 4.7. PROTOCOLOS DE NIVEL DE ENLACE REALES

Después de haber visto la teoría se describirá algunos de los protocolos de enlace comúnmente utilizados.

#### 4.7.1. HDLC - High-level Data Link Control

Usando como base la técnica de ventana deslizante descrita, IBM desarrolló en 1972 un protocolo de enlace denominado SDLC (Synchronous Data Link Control Protocol) para las redes SNA. Posteriormente IBM propuso SDLC para su estandarización a ANSI e ISO; cada uno de estos organismos estandarizó el protocolo introduciendo sus propias variantes sobre la propuesta inicial. En particular el protocolo desarrollado por ISO se denominó HDLC (High level Data Link Control) e introducía diversas mejoras sobre el protocolo originalmente desarrollado por IBM. La inmensa mayoría de los protocolos de enlace utilizados actualmente son subconjuntos del HDLC, como el PPP (Point to Point Protocol). La estructura de la trama HDLC es la siguiente:

Campo	Tamaño (bits)	Valor
Delimitador	8	01111110
Dirección	8	Variable
Control	8	Variable
Datos	>=0	Variable
Checksum	16	Variable
Delimitador	8	01111110

La trama se delimita mediante la secuencia 01111110, y para asegurar la transparencia de datos se utiliza relleno de bits (bit stuffing), es decir, se intercala un bit a 0 cuando en la parte de datos aparece una secuencia de cinco bits a 1, procediendo de modo inverso en el lado receptor. En sistemas síncronos cuando la línea no está transmitiendo información útil se envía continuamente la secuencia 0111111011111101111110.... Cada trama puede tener cualquier longitud a partir de 32 bits (sin contar los delimitadores), pudiendo no ser múltiplo de 8, ya que no se presupone una estructura de bytes. Por esto se suele decir que HDLC es un protocolo *orientado al bit*.

El campo *checksum* es un CRC que utiliza el generador polinómico CRC-CCITT.

El campo *datos*, también llamado en ocasiones carga útil (*payload*) puede o no estar presente; puede contener cualquier información y tener cualquier longitud, si bien la eficiencia del checksum disminuye cuando la longitud aumenta.

El campo *dirección* solo se utiliza en líneas multipunto. Cuando se quiere enviar una trama a todas las estaciones (envío broadcast) se utiliza la dirección 11111111.

El campo *control* es el corazón del protocolo. Cuando el primer bit es un cero indica que se trata de una trama de datos, también llamada de *información*. En ese caso la estructura de este campo es la siguiente:

Bits →	1	3	1	3
	0	SEQ	P/F	NEXT

- El subcampo SEQ contiene el número de secuencia de la trama.
- El subcampo P/F (Polling/Final) solo se utiliza en líneas multipunto.
- El subcampo NEXT contiene el ACK “piggybacked”; el convenio en este caso es que el ACK indica la siguiente trama que se espera recibir, no la última recibida (evidentemente se supone que esa trama habrá sido recibida correctamente).

Cuando el primer bit del campo control es un 1 y el segundo un 0 se trata de una trama de supervisión. Estas tramas se utilizan para enviar los ACK cuando no hay tráfico de datos suficiente y también para algunos mensajes de control. La estructura que tienen es la siguiente:

Bits →	2	2	1	3
	1	0	ORDEN	P/F

Según el valor del subcampo ORDEN las tramas de supervisión podrán enviar los siguientes cuatro comandos:

- **00** (Tipo 0): RECEIVE READY. Este es el nombre que recibe en el estándar el acuse de recibo (ACK). Este comando se utiliza cuando no hay tráfico de retorno suficiente para utilizar piggybacking.
- **01** (Tipo 1): REJECT. Corresponde al acuse de recibo negativo (NAK). Solicita retransmisión de una trama, y no acepta ninguna otra entretanto. Se utiliza cuando se

- emplea el mecanismo de retroceso N. La trama solicitada se especifica en el campo NEXT.
- **10** (Tipo 2): RECEIVE NOT READY. Indica un acuse de recibo (ACK) pero además solicita la suspensión del envío para evitar la saturación del receptor (control de flujo); el receptor enviará este mensaje cuando vea que tiene poco espacio para buffers. Para que la retransmisión se reanude el receptor deberá enviar más tarde un RECEIVE READY, REJECT o ciertas tramas de control.
  - **11** (Tipo 3): SELECTIVE REJECT. Se utiliza para solicitar retransmisión de una trama determinada cuando se está empleando retransmisión selectiva.
- En todos los casos el subcampo NEXT indica la siguiente trama que se espera recibir (o la que no se espera recibir en el caso de RECEIVE NOT READY).

Existe en HDLC un tercer tipo de trama que es el que se da cuando los dos primeros bits son 1; este tipo de tramas se denomina no numeradas y se utilizan para dos funciones completamente diferentes:

- Establecer determinados parámetros de inicialización del protocolo.
- Cuando se utiliza el servicio no orientado a conexión, es decir sin ACK. En este caso no es necesario numerar las tramas ya que no se pedirá retransmisión en ningún caso, de ahí la denominación de trama *no numerada*.

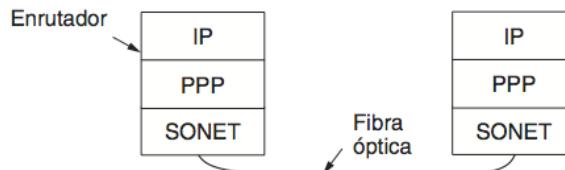
La estructura del campo control en las tramas no numeradas es la siguiente:

Bits →	2	2	1	3	
	1	1	ORDEN1/2	P/F	ORDEN2/2

En conjunto hay 5 bits que sirven para especificar diversos comandos.

#### 4.7.2. Paquetes sobre fibra óptica

SONET, es el protocolo de capa física que se utiliza con más frecuencia sobre los enlaces de fibra óptica de área extensa que constituyen el backbone (espina dorsal) de las redes de comunicaciones. Para transportar paquetes a través de estos enlaces, se necesita cierto mecanismo de entramado para diferenciar los paquetes ocasionales del flujo de bits continuo en el que se transportan. PPP se ejecuta en enrutadores IP para proveer este mecanismo.



PPP constituye una mejora del protocolo más simple conocido como SLIP (Protocolo de Línea Serial de Internet); se utiliza para manejar la configuración de detección de errores en los enlaces, soporta múltiples protocolos, permite la autenticación y tiene muchas otras funciones.

El formato de trama de PPP se escogió de modo que fuera muy parecido al de HDLC (Control de Enlace de Datos de Alto Nivel), una instancia muy utilizada de una familia anterior de protocolos.

La diferencia principal entre PPP y HDLC es que el primero está orientado a bytes, no a bits. En particular, PPP usa el relleno de bytes en las líneas y todas las tramas tienen un número entero de bytes.

El formato de trama de PPP se muestra en la siguiente figura. Todas las tramas PPP comienzan con el byte bandera del estándar de HDLC 0x7E (01111110). Este byte de bandera se rellena con bytes si ocurre dentro del campo de carga útil (Payload), mediante el byte de escape 0x7D. El siguiente byte es el resultado de aplicar un XOR al byte de escape y a 0x20, con lo cual se volteará el quinto bit. Por ejemplo, 0x7D 0x5E es la secuencia de escape para el byte bandera 0x7E. Esto significa que se puede buscar el inicio y el final de las tramas con sólo explorar en busca del byte 0x7E, ya que no ocurrirá en ningún otro lado. La regla para quitar el relleno de bytes al recibir una trama es buscar el byte 0x7D, eliminarlo y aplicar un XOR al siguiente byte junto con 0x20. Además, sólo se necesita un byte bandera entre trama y trama. Se pueden usar varios bytes bandera para llenar el enlace cuando no haya tramas para enviar.

Bytes	1	1	1	1 a 2	Variable	2 a 4	1
	Bandera 01111110	Dirección 11111111	Control 00000011	Protocolo	Carga útil ss	Suma de verificación ss	Bandera 01111110

El campo *dirección* no se utiliza. Siempre vale 11111111.

El campo *control* tiene por defecto el valor 00000011. Por defecto PPP suministra un servicio no orientado a conexión no fiable, es decir sin números de secuencia y acuses de recibo. Aunque no es lo normal, en el momento de establecer la conexión LCP puede negociar una transmisión fiable.

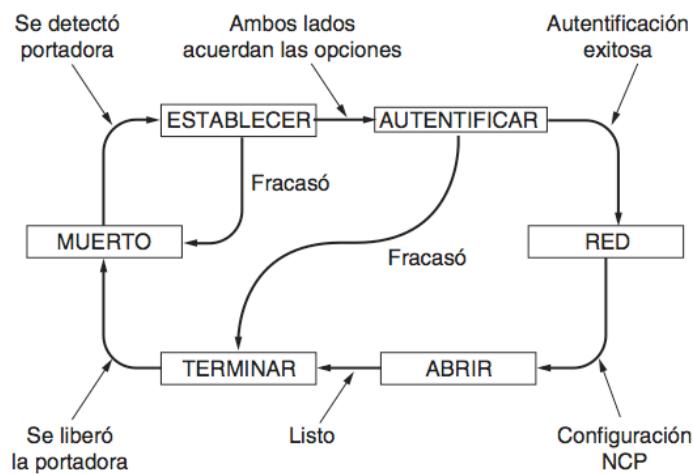
Salvo que se negocie una transmisión fiable los campos dirección y control contienen siempre la secuencia 1111111100000011. Para no transmitir estos dos bytes de información inútil en todas las tramas generalmente LCP negocia la supresión de estos dos bytes de las tramas al inicio de la sesión (salvo que se pida transmisión fiable).

El campo *protocolo* establece a qué tipo de protocolo pertenece el paquete recibido de la capa de red. De esta forma PPP permite establecer una comunicación multiprotocolo, es decir puede utilizarse para transmitir paquetes pertenecientes a diferentes protocolos del nivel de red entre dos computadores simultáneamente. Entre las posibilidades se encuentra IP, IPX(Novell), Appletalk, DECNET, OSI y otros.

El campo *datos o carga util* es de una longitud variable hasta un máximo que negocia LCP al establecer la conexión; por defecto el tamaño máximo de trama es de 1500 bytes.

El campo *checksum o suma de verificación* es normalmente de 2 bytes, pero puede ser de 4 si se negocia.

PPP es un mecanismo de entramado que puede transportar los paquetes de varios protocolos a través de muchos tipos de capas físicas. Para usar PPP sobre SONET se utiliza una suma de verificación de 4 bytes, ya que éste es el medio principal para detectar errores de transmisión a través de las capas física, de enlace y de red. También hay una característica inusual. La carga útil de PPP se mezcla aleatoriamente (scrambled) antes de insertarla en la carga útil de SONET. Antes de transportar las tramas PPP a través de líneas SONET, hay que establecer y configurar el enlace PPP. Las fases por las que pasa el enlace al activarlo, utilizarlo y desactivarlo se muestran en la figura:



El enlace inicia en el estado *MUERTO*, lo que significa que no hay conexión en la capa física. Al establecer una conexión en la capa física, el enlace pasa a *ESTABLECER*. En ese punto, los iguales de PPP intercambian una serie de paquetes LCP (cada uno de los cuales se transporta en el campo *Carga útil* de una trama PPP) para seleccionar de las posibilidades antes mencionadas, las opciones de PPP para el enlace. El igual que inicia propone las opciones y el igual que responde las acepta o las rechaza, todas o una parte de ellas. El que responde también puede hacer propuestas alternativas.

Si la negociación de opciones LCP tiene éxito, el enlace llega al estado *AUTENTIFICAR*. Ahora las dos partes pueden verificar las identidades una de la otra, si lo desean. Si la autenticación tiene éxito, el enlace entra al estado *RED* y se envía una serie de paquetes NCP para

configurar la capa de red. Es difícil generalizar sobre los protocolos NCP, ya que cada uno es específico para cierto protocolo de capa de red y permite hacer peticiones de configuración específicas para ese protocolo.

Una vez que el enlace llega a ABRIR, se puede llevar a cabo el transporte de datos. En este estado es en donde se transportan los paquetes IP en tramas PPP a través de la línea SONET. Al terminar el transporte de los datos, el enlace pasa al estado TERMINAR y de ahí se regresa al estado MUERTO cuando se desactiva la conexión de la capa física.

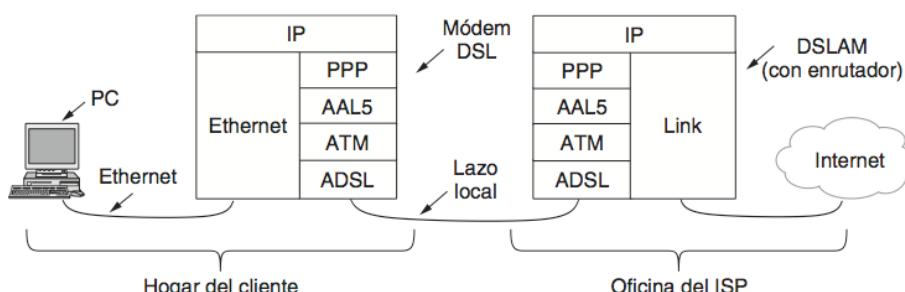
Con un amplio conjunto de opciones, PPP provee tres características principales:

- Un protocolo de control de enlace para activar líneas, probarlas, negociar opciones y desactivarlas en forma ordenada cuando ya no son necesarias. Este protocolo se llama LCP (Protocolo de Control de Enlace)
- NCP (Network Control Protocol). Este se encarga de negociar los parámetros específicos para cada protocolo utilizado. Por ejemplo, en el caso de una conexión IP desde un usuario conectado vía módem le asigna dinámicamente una dirección IP; esto es especialmente útil cuando (como normalmente ocurre) el número de direcciones IP disponibles es menor que el número de usuarios del servicio (aunque por supuesto el número de direcciones IP disponibles debe ser suficiente para poder asignar una diferente a cada usuario simultáneo).
- LCP permite utilizar diversos protocolos de autenticación, es decir que permiten validar al computador que llama (mediante el uso de claves tipo usuario/password). Esto resulta especialmente útil en el caso de conexiones por RTC, por ejemplo para proveedores de servicios Internet que han de facturar a sus usuarios en función del tiempo de conexión. El protocolo de autenticación más utilizado, conocido como CHAP (Challenge Handshake Protocol) utiliza el siguiente mecanismo:
  - El usuario se identifica ante el servidor con su código de usuario correspondiente.
  - El servidor envía al usuario una secuencia de caracteres arbitrariamente generada que el usuario debe transformar mediante un algoritmo de encriptado, usando como clave de encriptado su password.
  - Entretanto el servidor realiza el mismo proceso, es decir encripta la secuencia de caracteres seleccionada utilizando como clave de encriptado la password del usuario que se intenta identificar.
  - El usuario envía al servidor la secuencia encriptada y éste la compara con la suya; si ambas coinciden el servidor concluye que el usuario se ha identificado satisfactoriamente.

El uso de CHAP permite una identificación segura del usuario sin tener que enviar passwords por la red, evitando así los problemas de seguridad que esto supondría.

#### 4.7.3. Paquetes sobre ADSL

ADSL (Línea Asimétrica de Suscriptor Digital) conecta a millones de suscriptores desde su hogar a Internet, a tasas de transmisión de varios megabits/seg sobre el mismo enlace (lazo local telefónico).



El módem DSL envía bits sobre el lazo local a un dispositivo llamado DSLAM (Multiplexor de Acceso a la ADSL), el cual se encuentra en la compañía telefónica. Como se implementan distintos protocolos en diferentes redes, se ha optado por mostrar el escenario más popular. Dentro del hogar, una computadora como una PC envía paquetes IP al módem DSL mediante el uso de una capa de enlace como Ethernet. Despues, el módem DSL envía los paquetes IP sobre el lazo local al DSLAM. En el DSLAM (o en un enrutador conectado a éste, dependiendo de la implementación) se extraen los paquetes IP y se introducen en una red de ISP para llegar a cualquier destino en Internet.

Los protocolos que se muestran sobre el enlace ADSL en la figura empiezan desde la parte inferior, con la capa física de ADSL. Cerca de la parte superior de la pila se encuentra PPP, justo debajo de la capa de red IP. Este protocolo es el mismo PPP que transporta paquetes sobre SONET. Funciona de la misma manera para establecer y configurar el enlace y transportar los paquetes IP.

Los protocolos ATM y AAL5 están entre ADSL y PPP. El protocolo ATM (Modo de Transferencia Asíncrona) se diseñó a principios de 1990 y es una capa de enlace basada en la transmisión de celdas de información de longitud fija. Lo “asíncrono” en su nombre significa que las celdas no siempre se tienen que enviar de la misma manera en que se hace a través de las líneas sincrónicas, como en SONET. Las celdas sólo necesitan enviarse cuando haya información para transportar. ATM es una tecnología orientada a conexión. Cada celda transporta un identificador de circuito virtual en su encabezado y los dispositivos usan este identificador para reenviar celdas a través de las trayectorias de conexiones establecidas.

Cada una de las celdas es de 53 bytes de longitud, en donde 48 bytes son de la carga útil y 5 bytes constituyen el encabezado. Mediante el uso de celdas pequeñas, ATM puede dividir de una manera flexible el ancho de banda de un enlace de capa física entre distintos usuarios, en pequeñas porciones. Esta habilidad es útil cuando, por ejemplo, se envía tanto voz como datos a través de un enlace sin tener paquetes de datos extensos que producirían grandes variaciones en el retardo de las muestras de voz. La elección inusual en cuanto a la longitud de la celda (en comparación con la elección más natural de una potencia de 2) es una indicación de cuánta influencia política tuvo el diseño de ATM. El tamaño de 48 bytes para la carga útil fue un compromiso para resolver un interbloqueo entre Europa, que deseaba celdas de 32 bytes, y Estados Unidos, que quería celdas de 64 bytes.

Para enviar datos a través de una red ATM, es necesario asignarlos a una secuencia de celdas. Esta asignación se realiza mediante una capa de adaptación del ATM en un proceso llamado segmentación y reensamblaje. Se han definido varias capas de adaptación para distintos servicios, que varían desde los muestreos periódicos de voz hasta los datos de paquetes. La capa principal que se utiliza para los datos de paquetes es AAL5 (Capa de Adaptación de ATM 5).

Bytes	1 a 2	Variable	0 a 47	2	2	4
	Protocolo PPP	Carga útil PPP	Relleno	Sin usar	Longitud	CRC
Carga útil AAL5						Terminado AAL5

En la figura se muestra una trama AAL5. En vez de encabezado, tiene un terminador que proporciona la longitud y cuenta con una CRC de 4 bytes para la detección de errores. Naturalmente, la CRC es la misma que se utiliza para las redes PPP y LAN IEEE 802 como Ethernet. Al igual que una carga útil, la trama AAL5 tiene relleno. Esto redondea la longitud total para que sea un múltiplo de 48 bytes, de modo que la trama se pueda dividir equitativamente en celdas. No se necesitan direcciones en la trama, ya que el identificador de circuito virtual incluido en cada celda la llevará al destino correcto.

PPP usa a ATM en el caso de ADSL, con otro estándar llamado PPPoA (PPP sobre ATM). En realidad este estándar no es un protocolo, sino más bien una especificación sobre cómo trabajar con tramas PPP y AAL5.

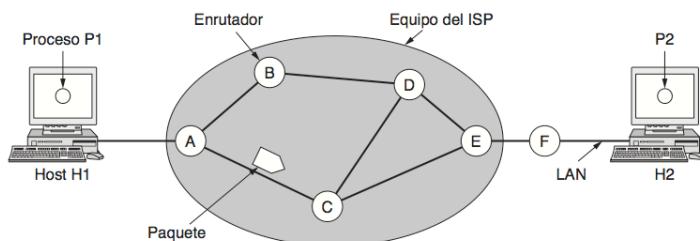
Sólo los campos de protocolo y carga útil de PPP se colocan en la carga útil de AAL5, como se muestra en la anterior figura. El campo de protocolo indica al DSLAM en el extremo lejano si la carga útil es un paquete IP o un paquete de otro protocolo tal como LCP. El extremo lejano sabe que las celdas contienen información PPP, ya que se estableció un circuito virtual ATM para este fin.

Dentro de la trama AAL5 no se necesita entramado PPP, ya que no serviría para ningún propósito puesto que ATM y AAL5 proveen de antemano el entramado. Tampoco se necesita la CRC de PPP, ya que AAL5 incluye la misma CRC. Este mecanismo de detección de errores complementa la codificación de capa física de ADSL que consta de un código de Reed-Solomon para corrección de errores y una CRC de 1 byte para la detección de los errores restantes que no se hayan detectado. Este esquema tiene un mecanismo de recuperación de errores mucho más sofisticado que cuando se envían paquetes a través de una línea SONET, ya que ADSL es un canal mucho más ruidoso.

**Tema V****Capa de Red**

La capa de red se encarga de llevar los paquetes todo el camino, desde el origen hasta el destino. Para llegar al destino tal vez sea necesario realizar muchos saltos en el camino por enrutadores intermedios (router). Para lograr sus objetivos, la capa de red debe conocer la topología de la red (es decir, el conjunto de todos los enrutadores y enlaces) y elegir las rutas apropiadas incluso para redes más grandes. También debe tener cuidado al escoger las rutas para no sobrecargar algunas de las líneas de comunicación y los enrutadores, y dejar inactivos a otros.

Un contexto típico de red se muestra en la siguiente figura, donde los componentes principales de la red son el equipo del Proveedor del Servicio de Internet (ISP) (enrutadores conectados mediante líneas de transmisión), que se muestra dentro del óvalo sombreado, y el equipo de los clientes, que se muestra fuera del óvalo. El host H1 está conectado de manera directa a un enrutador del ISP, A, tal vez en forma de una computadora en el hogar conectada a un módem DSL. En contraste, H2 se encuentra en una LAN (que podría ser una Ethernet de oficina) con un enrutador, F, el cual es propiedad del cliente, quien lo maneja. Este enrutador tiene una línea alquilada que va al equipo del ISP.



Las tareas principales que debe cumplir la capa de red, para hacer que los paquetes de información lleguen de un punto a otro son:

- **Reenvío (Forwarding) de paquetes.** Los paquetes que llegan por una interfaz deben ser reenviados por otra, luego de realizar una búsqueda en la tabla correspondiente.
- **Enrutamiento de la Red (Routing).** Llenado de las tablas correspondientes (Estáticamente o dinámicamente, mediante protocolos de enrutamiento).
- **Calidad de Servicio (QoS).** Como asegurar que distintos tipos de tráfico sean tratados de la manera más justa posible.
- **Control de Congestión.** Como hacer para que una red sobrecargada se comporte al máximo de su rendimiento y no colapse.

## 5.1. SERVICIOS PROPORCIONADOS A LA CAPA DE TRANSPORTE

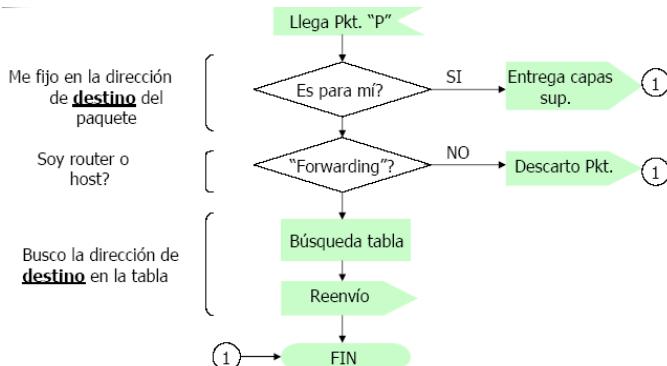
Los servicios que ofrece el nivel de red deberán en lo posible aislar al nivel de transporte de detalles tales como tipo de tecnología física utilizada (LAN, WAN, broadcast, etc.), número y topología de las subredes, etc. Las direcciones de red deberán tener un formato homogéneo, cualquiera que sea el medio físico o subred utilizados. Los **servicios de red** pueden ser orientados a conexión (CONS) o no orientados a conexión (CLNS). Ejemplos de servicios CLNS son el protocolo IP, el ISO CLNS elaborado a imagen y semejanza de IP, y el IPX desarrollado por Novell para su sistema operativo en red Netware. Ejemplos de servicios CONS son X.25, Frame Relay y ATM.

### 5.1.1. Servicio NO Orientado a la conexión (CLNS).

En el servicio sin conexión, los paquetes se transmiten por separado en la red y se enrutan de manera independiente. No se necesita una configuración por adelantado. En este contexto, por lo general los paquetes se conocen como datagramas (en analogía con los telegramas) y la red se conoce como red de datagramas. Las características principales de un servicio CLNS son:

- Se implementan las primitivas SEND PACKET y RECEIVE PACKET.
- No se garantiza el orden de llegada de los datagramas.
- Cada datagrama ha de llevar la dirección de destino. Cada uno ha de averiguar la ruta por sí mismo.

- Cada router ha de mantener una tabla que indique por qué interfaz debe encaminar los paquetes en función de su destino, pero no conserva información de las conexiones (pues no las hay); se dice que el router no tiene estados o que es "stateless".

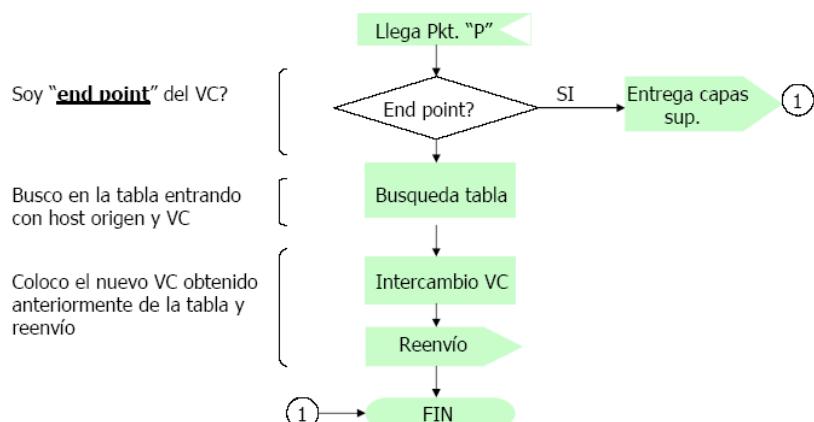


- Si un router de la red queda fuera de servicio la única consecuencia será la pérdida de los datagramas que estuvieran en proceso allí en ese momento, no se pierde ninguna información sobre la conexión, ya que no había conexión; el resto de routers de la red intentará buscar una ruta alternativa para comunicarse.
- Es difícil llevar a cabo un control de congestión, u ofrecer una QoS (Quality of Service, calidad de servicio).

### 5.1.2. Servicio Orientado a la conexión (CONS).

En el servicio orientado a conexión, hay que establecer una ruta del enrutador de origen al enrutador de destino antes de poder enviar cualquier paquete de datos. Esta conexión se conoce como VC (circuito virtual), en analogía con los circuitos físicos establecidos por el sistema telefónico, y la red se denomina red de circuitos virtuales. Las principales características de un servicio CONS son:

- Las dos entidades finales (hosts) establecen primero la conexión de forma explícita (un **circuito virtual** o VC) y le asignan un identificador. Esta conexión se utiliza para enviar todos los datos y luego se libera, también de forma explícita.
- El orden de entrega de los paquetes está garantizado.
- Los paquetes no necesitan llevar la dirección de destino (pero sí el número del VC por el que van a viajar). La ruta está marcada por el VC mientras éste está establecido. Cada nodo intermedio (comunicador) ha de tomar nota de los VCs existentes en cada momento. Cada **comunicador** ha de mantener una tabla que le indique la interfaz de entrada y de salida para cada VC que pasa por él.
- Si un comunicador queda fuera de servicio desaparecerán todos los VCs que pasen por él en ese momento, y los nodos afectados dejarán de comunicar (aunque podrán establecer un nuevo VC si hay un camino alternativo); la información de los VCs no estará presente cuando el comunicador vuelva a entrar en servicio.
- Es fácil efectuar un control de congestión, y también asegurar una QoS, ya que se tiene una información exacta de que conexiones discurren por cada línea física en cada momento.



En una red CLNS el nivel de transporte es normalmente más complejo pues ha de desempeñar más funciones que en una red CONS.

Una red CONS puede ser fiable (X.25) o no fiable (Frame Relay o ATM), mientras que una red CLNS normalmente es no fiable (garantiza la entrega). Generalmente cuando se quiere un servicio fiable en una red CLNS se implementa un servicio CONS a nivel de transporte; este es el caso por ejemplo cuando se utiliza el transporte TCP (CONS) sobre una red IP (CLNS).

## **5.2. ALGORITMOS DE ENRUTAMIENTO (ENCAMINAMIENTO)**

La función fundamental de la capa de red es el enrutamiento o encaminamiento, es decir averiguar por qué interfaz se han de mandar los paquetes recibidos para que lleguen a su destino. Con redes basadas en datagramas esta decisión se toma para cada paquete y el nodo que la realiza se denomina router o encaminador. Con redes orientadas a conexión (basadas en circuitos virtuales) la decisión se toma en el momento de establecer el circuito virtual, y a partir de entonces solo se comutan paquetes, de ahí la denominación de conmutador. En lo sucesivo, los paquetes de datos simplemente siguen la ruta ya establecida. Este último caso a veces se llama enrutamiento de sesión, dado que una ruta permanece vigente durante toda una sesión (por ejemplo, durante una sesión a través de una VPN).

Para poder encaminar los paquetes es preciso primero conocer cuál es la ruta a seguir hacia el destino especificado. El mecanismo que nos permite elegir la ruta a utilizar para posible destino es lo que denominamos un algoritmo de encaminamiento o de routing. Además de otras importantes virtudes un algoritmo de routing debe ser óptimo y justo. Estos conceptos a veces se contraponen, ya que el algoritmo que permite un aprovechamiento óptimo de los recursos no siempre es el que ofrece el reparto más equitativo.

Los algoritmos de enrutamiento se agrupan en dos clases principales: no adaptativos y adaptativos. Los algoritmos no adaptativos no basan sus decisiones de enrutamiento en mediciones o estimaciones del tráfico y la topología actuales. En cambio, la decisión de qué ruta se usará para llegar al destino se calcula por adelantado, fuera de línea, y se descarga en los enrutadores al arrancar la red, mediante tablas de rutas, por lo que no se necesita intercambiar información y por tanto no se requiere un protocolo de routing. Este procedimiento se denomina enrutamiento estático. Como no responde a las fallas, el enrutamiento estático es más útil para las situaciones en las que la elección de enrutamiento es clara.

En contraste, los algoritmos adaptativos cambian sus decisiones de enrutamiento para reflejar los cambios de topología y algunas veces también los cambios en el tráfico. Estos algoritmos denominados de enrutamiento dinámico difieren en cuanto al lugar de donde obtienen su información (por ejemplo, localmente, de los enrutadores adyacentes o de todos los enrutadores), el momento en que cambian sus rutas y la métrica que se usa para la optimización (por ejemplo, distancia, número de saltos o tiempo estimado de tránsito). Es preciso utilizar un protocolo de routing que permita a los routers intercambiar continuamente esa información, pero a cambio se consigue un mecanismo autoadaptativo que puede responder a situaciones cambiantes intentando resolver los problemas que se produzcan. Los algoritmos no pueden ser demasiado complejos, pues han de implementarse en los routers y ejecutarse en tiempo real con los recursos de CPU y memoria de que el router dispone.

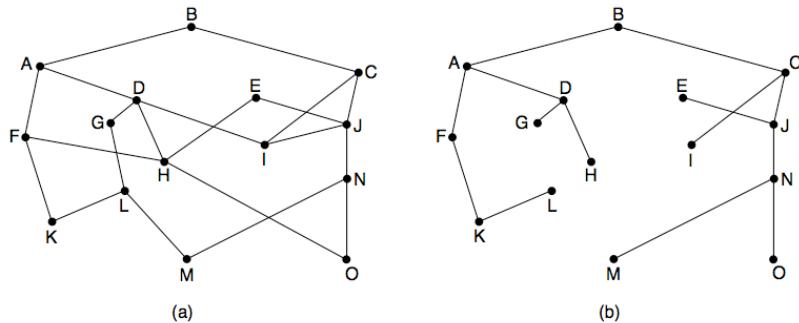
Haciendo una analogía con un viaje en coche se podría decir que el encaminamiento estático equivale a planificar la ruta para un viaje en coche antes de salir de casa utilizando los mapas y nuestro conocimiento a priori sobre el estado de las carreteras y la densidad de tráfico habitual en éstas; en cambio con encaminamiento dinámico se iría fijando la ruta sobre la marcha en base a la información que se obtuviese por radio, o de los puestos de información, sobre el estado de las carreteras, pudiendo así reaccionar a situaciones cambiantes tales como atascos, accidentes, caminos cerrados, fenómenos atmosféricos, etc. En el primer caso no se intercambia ninguna información durante el viaje, mientras que en el segundo sí.

### **5.2.1. El principio de optimalidad**

Es un principio fundamental para los algoritmos de encaminamiento que podemos enunciar así:

*Si B está en la ruta óptima de A a C, entonces el camino óptimo de B a C está incluido en dicha ruta.*

Una consecuencia importante de este principio es que todas las rutas óptimas para llegar a un punto determinado en una red forman un árbol con raíz en el punto de destino. El árbol no contiene bucles, decimos que es un spanning tree y siempre es posible llegar al punto de destino en un número finito de saltos (hops). Dicho árbol se conoce como árbol sumidero (o árbol divergente) y se muestra en la siguiente figura, donde en (a) se tiene una red y en (b) el árbol sumidero para B.



Según este principio, si el camino ABCO es óptimo para ir de A a O, entonces el camino CO es óptimo para ir de C a O.

### 5.2.2. Algoritmo de la ruta más corta

Una técnica simple para calcular las rutas óptimas con base en una imagen completa de la red, es construir un grafo de la red, en donde cada nodo del grafo representa un enrutador y cada arco del grafo representa una línea o enlace de comunicaciones. Para elegir una ruta entre un par específico de enrutadores, el algoritmo simplemente encuentra la ruta más corta entre ellos en el grafo. El concepto de la ruta más corta es una manera de medir la longitud de una ruta (distancia). Es evidente que en redes telemáticas no tiene mucho sentido emplear la distancia física. En los casos más simples la distancia se mide como el número de saltos (hops); a mayor número de saltos mayor distancia. Evidentemente esto es satisfactorio únicamente en casos muy simples en que todos los enlaces tienen la misma capacidad. Normalmente la distancia se calcula a partir de uno o varios de los siguientes parámetros:

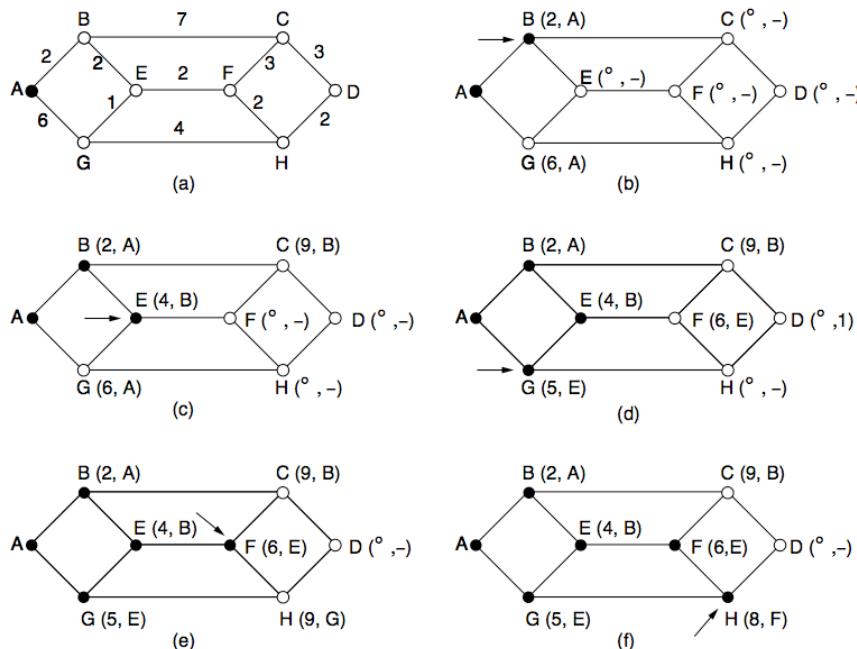
- La capacidad del enlace (información estática)
- El tráfico medio (puede ser información estática o dinámica)
- El retardo (información dinámica medida a partir de los paquetes enviados)
- La fiabilidad (información dinámica medida a partir de los paquetes enviados)

Combinando de determinada forma estos parámetros se calcula una cantidad que será la distancia (métrica) utilizada en el cálculo de las rutas óptimas. La forma de calcular la métrica se elige al configurar el router, pero es importante que sea consistente en todos los routers que participan en el protocolo de routing ya que de lo contrario se pueden dar situaciones asimétricas generalmente absurdas.

Existen varios algoritmos para calcular la ruta más corta entre dos nodos de un grafo. Uno de éstos es el de Dijkstra (1959), el cual encuentra las rutas más cortas entre un origen y todos los destinos en una red. Cada nodo se etiqueta (entre paréntesis) con su distancia desde el nodo de origen a través de la mejor ruta conocida. Las distancias no deben ser negativas, como lo serán si se basan en cantidades reales como ancho de banda y retardo. Al principio no se conocen rutas, por lo que todos los nodos tienen la etiqueta infinito. A medida que avanza el algoritmo y se encuentran rutas, las etiquetas pueden cambiar para reflejar mejores rutas. Una etiqueta puede ser tentativa o permanente. En un principio todas las etiquetas son tentativas. Una vez que se descubre que una etiqueta representa la ruta más corta posible del origen a ese nodo, se vuelve permanente y no cambia más.

En el siguiente gráfico, se muestra un grafo ponderado no dirigido, donde las ponderaciones representan algún tipo de métrica. Se quiere encontrar la ruta más corta posible de A a D. Se comienza por marcar el nodo A como permanente, lo cual se indica mediante un círculo relleno. Despues se examina, por turno, cada uno de los nodos adyacentes a A (el nodo de trabajo) y se reetiqueta cada uno de ellos con la distancia a A. Cada vez que se reetiqueta un nodo, también se lo etiqueta con el nodo desde el que se hizo la prueba, para poder reconstruir más tarde la ruta final. Si la red tuviera más de una ruta más corta de A a D y se quisiera encontrarlas todas, se tendría que

recordar todos los nodos de prueba que podrían llegar a un nodo con la misma distancia.



Una vez examinados cada uno de los nodos adyacentes a  $A$ , se revisan todos los nodos etiquetados tentativamente en el grafo completo y se hace permanente el de la etiqueta más pequeña (b). Éste se convierte en el nuevo nodo de trabajo. Ahora se comienza por  $B$  y se examina todos los nodos adyacentes a él. Si la suma de la etiqueta en  $B$  y la distancia desde  $B$  hasta el nodo en consideración es menor que la etiqueta de ese nodo, se tiene una ruta más corta, por lo que se reetiqueta ese nodo. Después de inspeccionar todos los nodos adyacentes al nodo de trabajo y cambiar las etiquetas tentativas si es posible, se busca en el grafo completo el nodo etiquetado tentativamente con el menor valor. Este nodo se hace permanente y se convierte en el nodo de trabajo para la siguiente ronda. En la figura se muestran los primeros seis pasos del algoritmo.

### 5.2.3. Encaminamiento por inundación (flooding)

La inundación consiste en enviar cada paquete por todas las interfaces, excepto por la que se ha recibido. La inundación genera grandes cantidades de paquetes duplicados; de hecho, puede ser una cantidad infinita a menos que se tomen algunas medidas para limitar el proceso. Una de estas medidas es integrar un contador de saltos al encabezado de cada paquete, que disminuya con cada salto, y que el paquete se descarte cuando el contador llegue a cero. Lo ideal es inicializar el contador de saltos con la longitud de la ruta entre el origen y el destino. Si el emisor desconoce el tamaño de la ruta, puede inicializar el contador para el peor caso; a saber, el diámetro total de la red.

La inundación no es práctica para enviar la mayoría de los paquetes, pero tiene algunos usos importantes. En primer lugar, asegura que un paquete se entregue en todos los nodos de la red. Esto podría ser un desperdicio si sólo hay un destino que necesite el paquete, pero es efectivo para difundir información. En las redes inalámbricas, todos los mensajes transmitidos por una estación los pueden recibir todas las demás estaciones dentro de su alcance de radio, lo cual de hecho se puede considerar como inundación; algunos algoritmos usan esta propiedad.

### 5.2.4. Encaminamiento por vector distancia

El algoritmo de enrutamiento por vector de distancia se lo conoce también como algoritmo de enrutamiento Bellman-Ford distribuido, en honor a los investigadores que lo desarrollaron.

En el encaminamiento por vector distancia cada router mantiene una tabla o vector que le indica la distancia mínima conocida hacia cada posible destino y que línea o interfaz debe utilizar para llegar a él. La tabla se actualiza regularmente con información obtenida de los routers vecinos. Cada router manda la tabla completa de distancias a todos sus vecinos, y solo a ellos. A partir de la información que tiene y la recibida de sus vecinos cada router recalculará continuamente su tabla de distancias. Por ejemplo si se quiere construir la tabla de enrutamiento del nodo  $i$ , dada la siguiente red:

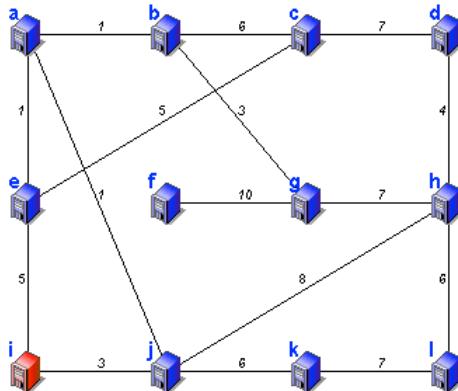
**Primera actualización**

Tabla de encaminamiento											
a	b	c	d	e	f	g	h	i	j	k	l
-	-	-	-	5	-	-	-	0	3	-	-
-	-	-	-	e	-	-	-	j	-	-	-

**Segunda actualización**

Tabla de encaminamiento											
a	b	c	d	e	f	g	h	i	j	k	l
4	-	10	-	5	-	-	11	0	3	9	-
j	-	e	-	e	-	-	j	-	j	j	-
+5 e	1	-	5	-	0	-	-	-	5	-	-
+3 j	1	-	-	-	-	-	8	3	0	6	-

**Tercera actualización**

Tabla de encaminamiento												
a	b	c	d	e	f	g	h	i	j	k	l	
4	5	10	15	5	-	18	11	0	3	9	16	
j	j	e	j	e	-	j	j	-	j	j	j	
+5 e	1	2	5	12	0	-	-	-	5	2	-	
+3 j	1	2	-	12	2	-	15	8	3	0	6	13

**Cuarta actualización**

Tabla de encaminamiento												
a	b	c	d	e	f	g	h	i	j	k	l	
4	5	10	15	5	28	8	11	0	3	9	16	
j	j	e	j	e	j	j	j	-	j	j	j	
+5 e	1	2	5	12	0	-	5	10	5	2	8	
+3 j	1	2	7	12	2	25	5	8	3	0	6	13

**Quinta actualización**

Tabla de encaminamiento												
a	b	c	d	e	f	g	h	i	j	k	l	
4	5	10	15	5	18	8	11	0	3	9	16	
j	j	e	j	e	j	j	j	-	j	j	j	
+5 e	1	2	5	12	0	15	5	10	5	2	8	
+3 j	1	2	7	12	2	15	5	8	3	0	6	13

La métrica (el valor utilizado para elegir la ruta óptima) puede ser número de saltos, retardo, paquetes encolados, etc., o una combinación de estos u otros parámetros. Para medir el retardo el router puede enviar paquetes de prueba que deben ser respondidos por el router remoto, aunque también es frecuente que los retardos se asignen de acuerdo con un convenio preestablecido en función del tipo de interfaz y de la capacidad del enlace, sin hacer ninguna medida real sobre el terreno. Cada router sólo conoce el valor de los parámetros para los enlaces con sus vecinos, los valores correspondientes a enlaces más lejanos los conoce de manera indirecta en base a la información sumarizada que sus vecinos le facilitan.

En el routing por vector distancia las noticias buenas se propagan rápidamente, pero se reacciona lentamente a las malas. Esto se conoce como el problema de la cuenta a infinito. Se han ideado multitud de trucos para resolver este problema, pero para cada nueva propuesta se ha encontrado una situación patológica en la que falla. No existe al parecer una solución definitiva a este problema, si bien la combinación de varios de esos trucos parece dar un resultado más que aceptable en la práctica. A pesar de sus inconvenientes el algoritmo del vector distancia se utiliza aun bastante en la

actualidad, y tiene fervientes partidarios sobre todo debido a su sencillez y consiguiente economía de recursos. El algoritmo del vector distancia fue utilizado en la ARPANET original. También se utilizó en DECNET e IPX, y se usa en Appletalk. Se usa actualmente en el protocolo RIP (Routing Information Protocol), que hasta 1988 era el único protocolo de routing utilizado en Internet. También se utiliza en los protocolos propietarios IGRP y EIGRP de Cisco, ampliamente extendidos.

### 5.2.5. Encaminamiento por estado del enlace

El algoritmo de encaminamiento basado en el estado del enlace se conoce también como algoritmo de Dijkstra o algoritmo SPF (Shortest Path First).

Este algoritmo apareció como un intento de resolver los problemas que planteaba el encaminamiento por vector distancia, fundamentalmente el de la cuenta a infinito. Se trata de un algoritmo más sofisticado y robusto, pero también más complejo. Su funcionamiento se describe mejor dividiéndolo en cinco fases:

1. Descubrir los routers vecinos y averiguar sus direcciones.
2. Medir el retardo o costo de llegar a cada vecino
3. Construir un paquete que resuma toda esta información,
4. Enviar este paquete a todos los routers de la red y recibir paquetes de ellos
5. Calcular el camino mas corto a cada router

#### Aprender sobre los vecinos

Para darse a conocer cada router cuando arranca envía paquetes de presentación (HELLO) por todas sus interfaces; los paquetes HELLO son respondidos con mensajes identificativos por los routers que los reciben.

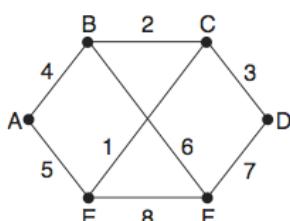
#### Establecimiento de los costos de los enlaces

El algoritmo de enrutamiento por estado del enlace requiere que cada enlace tenga una métrica de distancia o costo para encontrar las rutas más cortas. El costo para llegar a los vecinos se puede establecer de modo automático, o el operador de red lo puede configurar. Una elección común es hacer el costo inversamente proporcional al ancho de banda del enlace. Por ejemplo, una red Ethernet de 1 Gbps puede tener un costo de 1 y una red Ethernet de 100 Mbps un costo de 10. Esto hace que las rutas de mayor capacidad sean mejores opciones. Si la red está geográficamente dispersa, el retardo de los enlaces se puede considerar en el costo, de modo que las rutas a través de enlaces más cortos sean mejores opciones. La manera más directa de determinar este retardo es enviar un paquete especial ECO a través de la línea, que el otro extremo tendrá que regresar de inmediato. Si se mide el tiempo de ida y vuelta, y se divide entre dos, el enrutador emisor puede obtener una estimación razonable del retardo.

#### Construcción de los paquetes de estado del enlace

Una vez que se ha recabado la información necesaria para el intercambio, el siguiente paso es que cada enrutador construya un paquete que contenga todos los datos llamado LSP (Link State Packet). El paquete comienza con la identidad del emisor, seguida de un número de secuencia, una edad y una lista de vecinos. También se proporciona el costo para cada vecino.

Es fácil construir los paquetes de estado del enlace. La parte difícil es determinar cuándo construirlos. Una posibilidad es construirlos de manera periódica; es decir, a intervalos regulares. Otra posibilidad es construirlos cuando ocurra un evento significativo, como la caída o la reactivación de una línea o de un vecino, o cuando sus propiedades cambien en forma considerable.



	Enlace	Estado	Paquetes
A	B Sec.	C Sec.	E Sec.
Sec.	Sec.	Sec.	Sec.
Edad	Edad	Edad	Edad
B 4	A 4	B 2	A 5
E 5	C 2	C 3	B 6
	D 3	F 7	C 1
	F 6	E 1	D 7
			F 8
			E 8

### Distribución de los paquetes de estado del enlace

La parte más complicada del algoritmo es la distribución de los paquetes de estado del enlace. Todos los enrutadores deben recibir todos los paquetes de estado del enlace con rapidez y confiabilidad. Si se utilizan distintas versiones de la topología, las rutas que se calculen podrían tener inconsistencias como ciclos, máquinas inalcanzables y otros problemas.

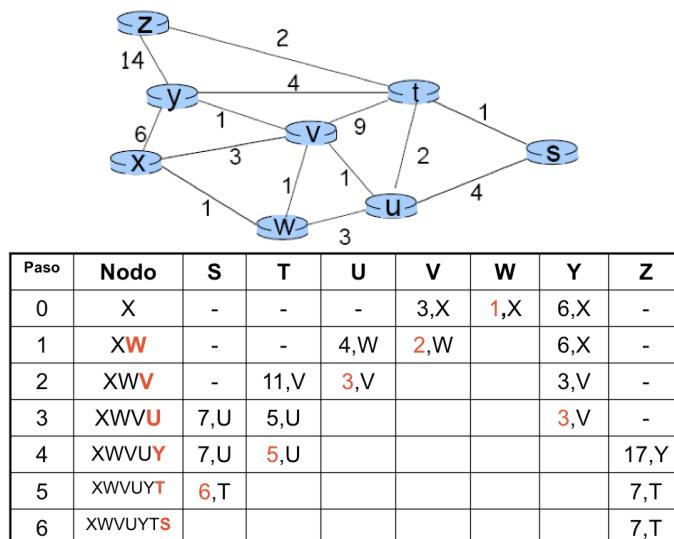
La idea fundamental es utilizar inundación para distribuir los paquetes de estado del enlace a todos los enrutadores. Con el fin de mantener controlada la inundación, cada paquete contiene un número de secuencia que se incrementa con cada nuevo paquete enviado. Los enrutadores llevan el registro de todos los pares (enrutador de origen, secuencia) que ven. Cuando llega un nuevo paquete de estado del enlace, se verifica y compara con la lista de paquetes ya vistos. Si es nuevo, se reenvía a través de todas las líneas, excepto aquella por la que llegó. Si es un duplicado, se descarta. Si llega un paquete con número de secuencia menor que el mayor visto hasta el momento, se rechaza como obsoleto debido a que el enrutador tiene datos más recientes.

Un campo importante es la edad de cada paquete que se disminuye una vez cada segundo. Cuando la edad llega a cero, se descarta la información de ese enrutador. Por lo general, un paquete nuevo entra, por ejemplo, cada 10 segundos, por lo que la información de los enrutadores sólo expira cuando un enrutador está caído (o cuando se pierden seis paquetes consecutivos, un evento poco probable). Los enrutadores también decrementan el campo Edad durante el proceso inicial de inundación para asegurar que no pueda perderse ningún paquete y sobrevivir durante un periodo de tiempo indefinido (se descarta el paquete cuya edad sea cero).

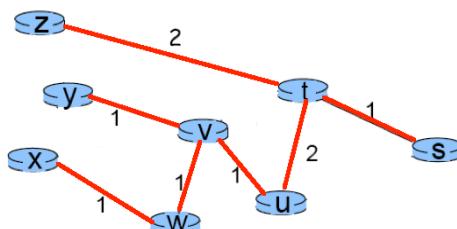
### Cálculo de las nuevas rutas

Una vez que un enrutador ha acumulado un conjunto completo de paquetes de estado del enlace, puede construir el grafo de toda la red debido a que todos los enlaces están simbolizados. De hecho, cada enlace se representa dos veces, una para cada dirección. Las distintas direcciones pueden tener incluso costos diferentes. Así, los cálculos de la ruta más corta pueden encontrar rutas del enrutador A a B que sean distintas a las del enrutador de B a A.

Ahora se puede ejecutar localmente el algoritmo de Dijkstra para construir las rutas más cortas a todos los destinos posibles. Los resultados de este algoritmo indican al enrutador qué enlace debe usar para llegar a cada destino. Esta información se instala en las tablas de enrutamiento y se puede reanudar la operación normal. Por ejemplo si se quiere construir la tabla de enrutamiento del nodo X, dada la siguiente red:



En el algoritmo basado en el estado del enlace cada router puede, a partir de la información obtenida, conocer su árbol de expansión o spanning tree completo, mientras que esto no es posible con routing por el vector distancia.



Entre los protocolos de routing que utilizan algoritmos basados en el estado del enlace se encuentra OSPF (Open Shortest Path First) que es el protocolo de routing estándar de Internet (aunque también se utilizan otros). Otro protocolo basado en el algoritmo del estado del enlace también utilizado en Internet y que proviene del mundo OSI es IS-IS (Intermediate System-Intermediate System). IS-IS es multiprotocolo, es decir, soporta múltiples protocolos de red por encima. OSPF está basado en IS-IS, pero no es multiprotocolo.

### 5.2.6. Encaminamiento jerárquico

A medida que crece el tamaño de las redes, también lo hacen en forma proporcional las tablas de enruteamiento del enruteador. Las tablas que están en crecimiento constante no sólo consumen memoria del enruteador, sino que también se necesita más tiempo de CPU para examinarlas y más ancho de banda para enviar informes de estado entre enruteadores. En cierto momento, la red puede crecer hasta el punto en que ya no sea viable que cada enruteador tenga una entrada para cada uno de los demás enruteadores, por lo que el enruteamiento tendrá que hacerse de manera jerárquica, como ocurre en la red telefónica.

Cuando se utiliza el enruteamiento jerárquico, los enruteadores se dividen en **regiones**. Cada enruteador conoce todos los detalles para enrutar paquetes a destinos dentro de su propia región, pero no sabe nada de la estructura interna de las otras regiones. Cuando se interconectan diferentes redes, es natural considerar cada una como región independiente con el fin de liberar a los enruteadores de una red de la necesidad de conocer la estructura topológica de las demás redes.

En las redes enormes, tal vez no sea suficiente una jerarquía de dos niveles; puede ser necesario agrupar las regiones en clústeres, los clústeres en zonas, las zonas en grupos, etc., etc.

### 5.2.7. Encaminamiento por difusión (broadcast)

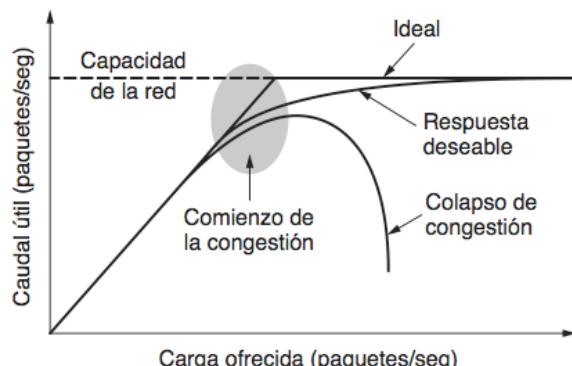
En algunos casos se necesita enviar un paquete a todos los destinos posibles en una red, es decir se quiere hacer un envío broadcast. La forma más sencilla de conseguirlo es **inundación**, técnica especialmente apropiada en este caso que ya ha sido descrita anteriormente.

Otro método es el **routing multidestino**, que consiste en mandar un único paquete con todas las direcciones de destino; el paquete es replicado en cada router por las interfaces por donde debe enviarse, es decir, las que son parte de la mejor ruta para alguno de los destinos indicados.

Otro algoritmo es construir el árbol de expansión o **spanning tree** con raíz en el origen y seguirlo, replicando el paquete allí donde haya una bifurcación. El spanning tree no tiene bucles. Este sistema es óptimo, ya que se asegura que la distribución se hará generando el número mínimo de paquetes y sin envíos duplicados. Pero esto requiere que cada router conozca cuales de sus interfaces forman parte del spanning tree para el router origen y cuales no, es decir los routers han de conocer en detalle la topología de la red. Con routing del estado del enlace los routers poseen esta información.

## 5.3. ALGORITMOS DE CONTROL DE CONGESTIÓN

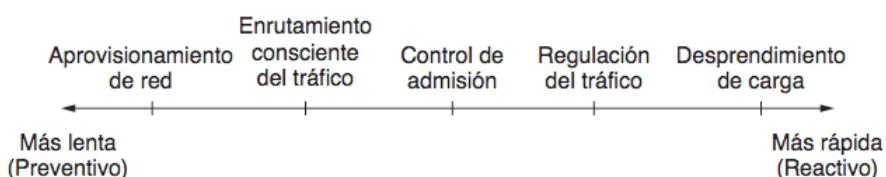
Congestión es cuando hay demasiados paquetes presentes en una red (o en una parte de ella), por lo que hay retardo o pérdida en los paquetes y se degrada el desempeño. Las capas de red y de transporte comparten la responsabilidad de manejar la congestión. Como ésta ocurre dentro de la red, la capa de red es quien la experimenta en forma directa y en última instancia debe determinar qué hacer con los paquetes sobrantes. Sin embargo, la manera más efectiva de controlar la congestión es reducir la carga que la capa de transporte coloca en la red. Para ello se requiere que las capas de red y de transporte trabajen en conjunto.



En la figura se muestra el comienzo de la congestión. Cuando la cantidad de paquetes que el host envía a la red está muy por debajo de su capacidad de transporte, la cantidad entregada es proporcional a la cantidad enviada. Si se envía el doble de paquetes, se entrega el doble de ellos. Sin embargo, a medida que la carga ofrecida se acerca a la capacidad de transporte, las ráfagas de tráfico llenan ocasionalmente los búferes dentro de los enrutadores y se pierden algunos paquetes. Estos paquetes perdidos consumen una parte de la capacidad, por lo que la cantidad de paquetes entregados cae por debajo de la curva ideal. Entonces la red está congestionada. A menos que la red esté bien diseñada, puede experimentar un colapso por congestión, en donde el desempeño se desploma a medida que aumenta la carga ofrecida más allá de la capacidad. Esto puede ocurrir debido a que los paquetes se pueden retrasar el tiempo suficiente dentro de la red como para que ya no sean útiles cuando salgan de ella.

### 5.3.1. Métodos para el control de la congestión

La presencia de congestión significa que la carga es (temporalmente) mayor de la que los recursos (en una parte de la red) pueden manejar. Dos soluciones son obvias: aumentar los recursos o reducir la carga. Estas soluciones por lo general se aplican en distintas escalas de tiempo, para prevenir la congestión o reaccionar ante ella una vez que se presenta.



#### 5.3.1.1 Aprovisionamiento de red

La manera más básica de evitar la congestión es construir una red que coincida bien con el tráfico que transmite. Si hay un enlace con poco ancho de banda en la ruta a través de la cual se dirige la mayor parte del tráfico, es probable que haya congestión. Algunas veces se pueden agregar recursos en forma dinámica cuando hay un problema grave de congestión. Lo más frecuente es que los enlaces y enrutadores que se utilizan mucho en forma regular se actualicen a la primera oportunidad. A esto se le conoce como aprovisionamiento y ocurre en una escala de tiempo de meses, con base en las tendencias de tráfico de largo plazo.

#### 5.3.1.2. Enrutamiento consciente del tráfico

Los esquemas de enrutamiento vimos con anterioridad utilizaban ponderaciones de enlaces fijas. Estos esquemas se adaptaban a los cambios en la topología, pero no a los cambios en la carga. El objetivo al tener en cuenta la carga al calcular las rutas es desviar el tráfico de los puntos más activos que serán los primeros lugares en la red en experimentar congestión.

La manera más directa de hacer esto es establecer la ponderación de enlaces de manera que sea una función del ancho de banda del enlace (fijo) y el retardo de propagación más la carga medida (variable) o el retardo de encolamiento promedio. Así, las rutas de menor ponderación favorecerán a las rutas que tengan cargas más ligeras, siendo todo lo demás igual.

En Internet los protocolos de enrutamiento por lo general no ajustan sus rutas dependiendo de la carga, sino que los ajustes se realizan fuera del protocolo de enrutamiento, al cambiar lentamente sus entradas. A esto se le denomina ingeniería de tráfico.

#### 5.3.1.3. Control de admisión

Una técnica que se utiliza mucho en las redes de circuitos virtuales para la congestión es el control de admisión. La idea es simple, no se debe establecer un nuevo circuito virtual a menos que la red pueda transportar el tráfico adicional sin congestionarse. Por lo tanto, pueden fallar los intentos por establecer un circuito virtual; ya que dejar entrar más personas cuando la red está ocupada sólo empeora las cosas.

El truco con este método es averiguar cuándo puede un nuevo circuito virtual provocar una congestión. A menudo el tráfico se describe en términos de su tasa de transmisión y forma. El problema de cómo describirlo en una forma simple pero significativa es difícil, ya que por lo general el tráfico es de ráfagas; la tasa promedio es sólo la mitad de la historia. Por ejemplo, el tráfico que varía mientras se navega por la web es más difícil de manejar que una película de flujo continuo con la misma velocidad real de transporte a largo plazo, pues es más probable que las ráfagas del tráfico web congestionen los enrutadores en la red. Un descriptor de uso común que captura este efecto es

la cubeta agujereada o con goteo (*leaky bucket*) o cubeta con token (*token bucket*). Una cubeta con goteo tiene dos parámetros que vinculan la tasa promedio y el tamaño de la ráfaga instantánea de tráfico. Armada con las descripciones del tráfico, la red puede decidir si admite o no el nuevo circuito virtual. Una posibilidad es que la red reserve suficiente capacidad a lo largo de las rutas de cada uno de sus circuitos virtuales, de modo que no ocurra una congestión. En este caso, la descripción del tráfico es un acuerdo de servicio en cuanto a lo que la red garantizará a sus usuarios. La red puede usar las descripciones del tráfico para el control de admisión. De esta forma, la tarea es estimar cuántos circuitos caben dentro de la capacidad de transporte de la red sin que haya congestión.

#### 5.3.1.4. Regulación de tráfico

En Internet y en muchas otras redes de computadoras, los emisores ajustan sus transmisiones para enviar tanto tráfico como la red pueda distribuir. En este escenario, la red aspira a operar justo antes de que comience la congestión. Cuando la congestión es inminente, debe pedir a los emisores que reduzcan sus transmisiones y su velocidad. Esta retroalimentación es algo común, en vez de una situación excepcional.

Los métodos para regular el tráfico se pueden usar en las redes de datagramas y en las de circuitos virtuales. Cada método debe resolver dos problemas:

- En primer lugar, los enrutadores deben determinar cuando se acerca la congestión, siendo lo ideal antes de que haya llegado. Para ello, cada enrutador puede monitorear en forma continua los recursos que utiliza. Tres posibilidades son:
  - Usar los enlaces de salida,
  - el búfer para los paquetes puestos en cola dentro del enrutador y
  - el número de paquetes que se pierden debido a una capacidad insuficiente.
 De estas posibilidades, la segunda es la más útil. Los promedios de uso no justifican directamente las ráfagas de la mayoría del tráfico; un uso del 50% puede ser bajo para un tráfico uniforme y demasiado alto para un tráfico muy variable. Las cuentas de los paquetes perdidos llegan demasiado tarde. La congestión ya ha comenzado para cuando se empiezan a perder los paquetes.
- El segundo problema es que los enrutadores deben entregar una retroalimentación oportuna a los emisores que provocan la congestión. Ésta se experimenta en la red, pero para aliviarla se requiere una acción de parte de los emisores que están usando la red. Para entregar la retroalimentación, el enrutador debe identificar a los emisores apropiados. Después, debe advertirles con cuidado, sin enviar más paquetes a la red que ya está congestionada.

#### Paquetes reguladores

La manera más directa de notificar a un emisor sobre la congestión es decírselo directamente. En este método, el enrutador selecciona un paquete congestionado y envía un paquete regulador de vuelta al host de origen, proporcionándole el destino encontrado en el paquete. El paquete original se puede etiquetar (se activa un bit de encabezado) de modo que no genere más paquetes reguladores más adelante en la ruta y después se reenvíe de la manera usual. Para evitar aumentar la carga en la red durante un momento de congestión, el enrutador tal vez sólo envíe paquetes reguladores a una tasa de transmisión baja. Cuando el host de origen obtiene el paquete regulador, se le pide que reduzca el tráfico enviado al destino especificado; por ejemplo, un 50%.

#### Notificación explícita de congestión

En vez de generar paquetes adicionales para advertir sobre la congestión, un enrutador puede etiquetar cualquier paquete que reenvíe (para lo cual establece un bit en el encabezado de éste) para indicar que está experimentando una congestión. Cuando la red entrega el paquete, el destino puede observar que hay congestión e informa al emisor sobre ello cuando envíe un paquete de respuesta. En consecuencia, el emisor puede regular sus transmisiones como antes. A este diseño se le conoce como ECN (Notificación Explícita de Congestión) y se utiliza en Internet.

#### Contrapresión de salto por salto

A largas distancias o altas velocidades, muchos paquetes nuevos se pueden transmitir una vez que se haya señalado la congestión debido al retardo antes de que la señal haga efecto.

Un método alternativo es hacer que el paquete regulador ejerza su efecto en cada salto por el que pase. El efecto neto de este esquema de salto por salto es proporcionar un alivio rápido en el punto de congestión, a expensas de usar más búferes ascendentes. De esta manera se puede cortar la congestión de raíz sin que se pierdan paquetes.

### 5.3.1.5. Desprendimiento de carga

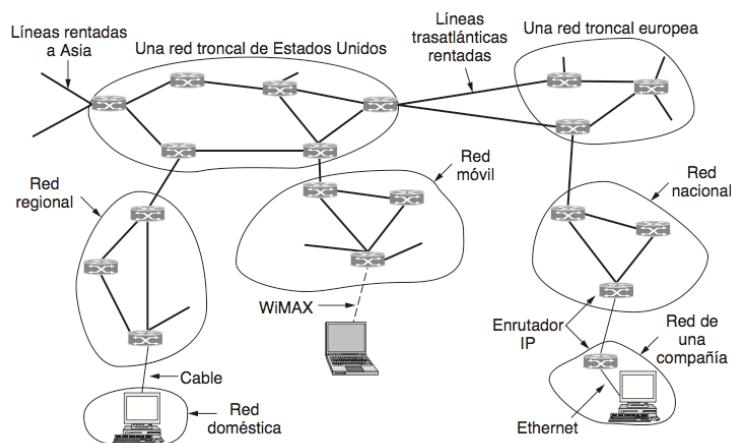
Cuando ninguno de los métodos anteriores elimina la congestión, los enrutadores pueden desprendérse de carga, es decir, que cuando se inunda a los enrutadores con paquetes que no pueden manejar, simplemente se desechan. La pregunta clave para un enrutador abrumado por paquetes es cuáles paquetes desechar. La opción preferida puede depender del tipo de aplicaciones que utiliza la red. Para implementar una política inteligente de descarte, las aplicaciones deben marcar sus paquetes para indicar a la red qué tan importantes son. Así, al tener que descartar paquetes, los enrutadores pueden eliminar primero los paquetes de la clase menos importante, luego los de la siguiente clase más importante, y así en lo sucesivo.

### Detección temprana aleatoria

Es más efectivo lidiar con la congestión cuando apenas empieza que dejar que dañe la red y luego tratar de solucionarlo; descartar paquetes antes de que se agote realmente el espacio en el búfer. Al hacer que los enrutadores descarten los paquetes antes de que la situación se vuelva imposible, hay tiempo para que la fuente tome acción antes de que sea demasiado tarde. Un algoritmo popular para realizar esto se conoce como RED (Detección Temprana Aleatoria). Para determinar cuándo hay que empezar a descartar paquetes, los enrutadores mantienen un promedio acumulado de sus longitudes de cola. Cuando la longitud de cola promedio en algún enlace sobrepasa un umbral, se dice que el enlace está congestionado y se descarta una pequeña fracción de los paquetes al azar. Al elegir los paquetes al azar es más probable que los emisores más rápidos vean un desprendimiento de paquetes; ésta es la mejor opción, ya que el enrutador no puede saber cuál fuente está causando más problemas en una red de datagramas. El emisor afectado observará la pérdida cuando no haya confirmación de recepción, y entonces el protocolo de transporte reducirá su velocidad. Así, el paquete perdido entrega el mismo mensaje que un paquete regulador, pero de manera implícita sin que el enrutador envíe ninguna señal explícita.

## 5.4. La capa de Red en Internet

En la capa de red, Internet puede verse como un conjunto de redes, o **Sistemas Autónomos** (AS) interconectados. No hay una estructura real, pero existen varias redes troncales (backbones) principales. Éstas se construyen a partir de líneas de alto ancho de banda y enrutadores rápidos. Las más grandes de estas redes troncales, a la que se conectan todos los demás para llegar al resto de Internet, se llaman **redes de Nivel 1**. Conectadas a las redes troncales hay ISP (Proveedores de Servicio de Internet) que proporcionan acceso a Internet y están llenas de servidores y redes regionales (de nivel medio). A las redes regionales están conectados más ISP, LAN de muchas universidades y empresas, y otras redes de punta.

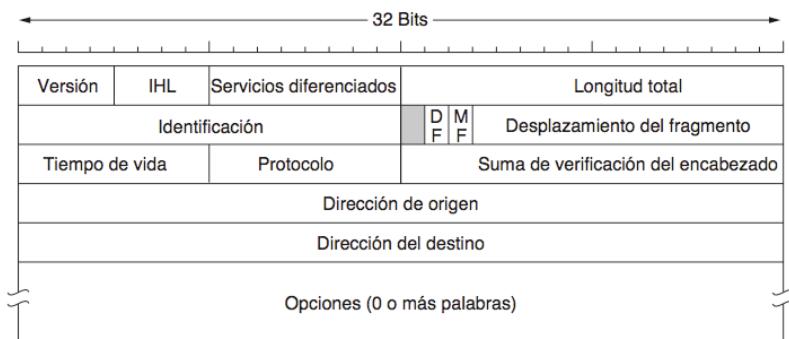


El pegamento que mantiene unida a Internet es el protocolo de capa de red, IP (Protocolo de Internet). IP se diseñó desde el principio con la interconexión de redes en mente. Una buena manera de visualizar la capa de red es que su trabajo es proporcionar un medio de mejor esfuerzo (sin garantía) para transportar paquetes de la fuente al destino, sin importar si estas máquinas están en la misma red o si hay otras redes entre ellas. La comunicación en Internet funciona de la siguiente manera. La capa de transporte toma flujos de datos y los divide para poder enviarlos como paquetes IP. En teoría, los paquetes pueden ser de hasta 64 Kbytes cada uno, pero en la práctica por lo general no sobrepasan los 1500 bytes (ya que son colocados en una trama de Ethernet). Los

enrutadores IP reenvían cada paquete a través de Internet, a lo largo de una ruta de un enrutador a otro, hasta llegar al destino. En el destino, la capa de red entrega los datos a la capa de transporte, que a su vez los entrega al proceso receptor. Cuando todas las piezas llegan finalmente a la máquina de destino, la capa de red las vuelve a ensamblar para formar el datagrama original y se entrega a la capa de transporte.

#### 5.4.1. EL PROTOCOLO IP VERSION 4

En una red IP absolutamente toda la información viaja en datagramas IP. El datagrama IP tiene dos partes: cabecera y texto; la cabecera tiene una parte fija de 20 bytes y una opcional de entre 0 y 40 bytes. La longitud total de la cabecera siempre es múltiplo de 4; esto garantiza un proceso eficiente por parte de equipos (hosts o routers) cuya arquitectura optimiza el acceso a direcciones de memoria que estén en frontera de 32 bits.



El campo **versión** permite que coexistan en la misma red sin ambigüedad paquetes de distintas versiones de IP; la versión actualmente utilizada es la 4.

El campo **IHL** especifica la longitud de la cabecera, ya que ésta puede variar debido a la presencia de campos opcionales. Se especifica en palabras de 32 bits. La longitud mínima es 5 y la máxima 15, que equivale a 40 bytes de información opcional. La longitud de la cabecera siempre ha de ser un número entero de palabras de 32 bits, por lo que si la longitud de los campos opcionales no es un múltiplo exacto de 32 bits se añade un relleno al final de la cabecera.

El campo **Servicios Diferenciados** ha sustituido recientemente al antes denominado tipo de servicio. Su finalidad es implementar Calidad de Servicio en redes IP mediante la arquitectura denominada Servicios Diferenciados o Diffserv.

El campo **longitud total** especifica la longitud del datagrama completo (cabecera incluida) en bytes. El valor máximo es 65535 bytes. Este campo sirve para saber donde termina el datagrama información que es realmente necesaria sólo en muy pocos casos, ya que en la mayoría de situaciones puede deducirse a partir de la información que posee el nivel de enlace. Los cuatro campos siguientes (**Identificación**, **DF**, **MF** y **Fragment Offset**) están relacionados con la fragmentación de datagramas que se explicará más adelante.

El campo **TTL** (Tiempo de Vida) sirve para descartar un datagrama cuando ha dado un número excesivo de saltos o ha pasado un tiempo excesivo viajando por la red y es presumiblemente inútil. Se trata de un contador regresivo que indica el tiempo de vida restante del datagrama medido en segundos, de forma que si llega a valer cero el datagrama debe ser descartado. Además cada router por el que pasa dicho datagrama está obligado a restar uno del TTL, independientemente del tiempo que tarde en reenviarlo. Esto evita que por algún problema en las rutas se produzcan bucles y un datagrama permanezca flotando indefinidamente en la red. Es habitual utilizar para el TTL los valores 64 o 255. El comando de prueba ping permite fijar el valor inicial del TTL en los datagramas de prueba enviados; por ejemplo en los sistemas operativos Windows se utiliza para este fin la opción *-i* del comando ping.

El campo **protocolo** especifica a qué protocolo del nivel de transporte corresponde el datagrama. La tabla de protocolos válidos y sus correspondientes números son controlados por el IANA (Internet Assigned Number Authority) [www.iana.org/numbers.html](http://www.iana.org/numbers.html).

El campo **suma de verificación (checksum)** sirve para detectar errores producidos en la cabecera del datagrama; no es un CRC sino el complemento a uno en 16 bits de la suma complemento a uno de toda la cabecera tomada en campos de 16 bits (incluidos los campos opcionales si los hay). Para el cálculo el campo checksum se pone a sí mismo a ceros. El checksum permite salvaguardar al datagrama de una alteración en alguno de los campos de la cabecera que pudiera producirse por ejemplo por un problema hardware en un router.

Los campos **dirección de origen** y **dirección de destino** corresponden a direcciones IP de cuatro bytes según el formato que se verá más adelante.

Los campos opcionales de la cabecera no siempre están soportados en los routers y se utilizan muy raramente. Cada campo opcional está compuesto por una etiqueta, seguida opcionalmente de información adicional. Los más importantes son los siguientes:

- *Record route (registrar la ruta)*: Esta opción solicita a los routers por los que pasa que anoten su dirección en la cabecera del datagrama, con lo que al final del trayecto se dispone de una traza de la ruta seguida para fines de prueba o diagnóstico de problemas. Esto es como si cada router estampara su sello en el datagrama antes de reenviarlo. El máximo de direcciones que puede registrarse es de 9, ya que si se registraran 10 se ocuparían los 40 bytes de opciones y no quedaría sitio para la propia opción record route. El comando de prueba ping permite solicitar el uso de este campo para registrar la ruta; por ejemplo en los sistemas operativos Windows se utiliza para este fin la opción *-r* del comando ping.
- *Timestamp (marca de tiempos)*: esta opción actúa de manera similar a record route, pero el router en lugar de anotar su dirección anota el instante (tiempo desde la medianoche en milisegundos) en el que el datagrama pasa por él; es un campo entero de 32 bits. También puede acompañarse el timestamp de la dirección IP del router; en este caso cada anotación ocupa ocho bytes y solo pueden producirse cuatro ya que la quinta superaría el tamaño máximo del campo opciones (la propia indicación de que se trata de un campo timestamp ya ocupa cuatro bytes). El comando de prueba ping permite solicitar el uso de este campo para registrar la ruta y hora; por ejemplo en los sistemas operativos Windows se utiliza para este fin la opción *-s* del comando ping.

El uso de los campos opcionales de la cabecera IP tiene generalmente problemas de rendimiento, ya que las implementaciones de los routers optimizan el código para las situaciones normales, es decir para datagramas sin campos opcionales. Aun en el caso de que las opciones estén implementadas lo harán generalmente de forma poco eficiente, ya que en el diseño del equipo no se ha hecho énfasis en su optimización. Por tanto solo deben utilizarse en situaciones de prueba o diagnóstico de errores, nunca en entornos en producción.

#### 5.4.2. Fragmentación

El tamaño de un datagrama IP se especifica en un campo de dos bytes en la cabecera, por lo que su valor máximo es de 65535 bytes, pero muy pocos protocolos o tecnologías a nivel de enlace admiten enviar tramas de semejante tamaño. Normalmente el nivel de enlace no fragmenta, por lo que tendrá que ser IP el que adapte el tamaño de los datagramas para que quepan en las tramas del nivel de enlace; por tanto en la práctica el tamaño máximo del datagrama viene determinado por el tamaño máximo de trama característico de la red utilizada. Este tamaño máximo de datagrama se conoce como MTU (Maximum Transfer Unit). La tabla siguiente muestra algunos ejemplos de valores de MTU característicos de las redes más habituales.

Protocolo a nivel de enlace	MTU (bytes)
PPP (valor por defecto)	1500
Frame relay	1600 normalmente
Ethernet DIX	1500
Ethernet LLC-SNAP	1492
FDDI	4352
Hyperchannel	65535
Classical IP over ATM	9180

Se puede distinguir dos situaciones en las que se produce fragmentación. La primera, denominada *fragmentación en ruta*, se produce cuando un datagrama es creado por un host en una red con un valor determinado de MTU y en su camino hacia el host de destino ha de pasar por otra red con una MTU menor. En estos casos el router que hace la transición a la red de MTU menor ha de fragmentar los datagramas para que no excedan el tamaño de la nueva red.

La segunda, llamada *fragmentación en origen*, se produce como consecuencia del diseño de la aplicación. Existe una fuerte polémica sobre la conveniencia de que realice la fragmentación en ruta, ya que esto perjudica seriamente la eficiencia de los routers por la carga de CPU que supone y la de la red ya que la probabilidad de perder algún fragmento es mayor y en ese caso es preciso reenviar todos los fragmentos. En general se considera preferible realizar la fragmentación en origen siempre que esto sea posible. Una forma fácil de conseguirlo es utilizar un MTU de 1500, valor que es soportado por la mayoría de las redes existentes, con lo que se minimizan los casos en que los routers han de recurrir a la fragmentación.

La fragmentación se hace cortando la parte de datos del datagrama en trozos tales que cada fragmento con su cabecera quepa en la MTU de la nueva red (redondeado por abajo para que la cantidad de datos sea múltiplo de ocho bytes). Todos los campos de la cabecera del datagrama original se replican en los fragmentos excepto aquellos que se emplean para distinguir los fragmentos. Una vez fragmentado un datagrama no se reensambla hasta que llegue al host de destino, aun cuando en el trayecto pase a redes que admitan una MTU mayor.

El campo **Identificación** de la cabecera IP lo usa el emisor para marcar en origen cada datagrama emitido; de esta forma en caso de que se produzca fragmentación el receptor podrá reconocer las partes que corresponden al mismo datagrama original, ya que todas irán acompañadas de la misma identificación.

El bit **DF** (Don't Fragment) cuando está a 1 indica a los routers que este datagrama no debe fragmentarse. Normalmente esto se hace por uno de los dos motivos siguientes:

- El receptor no está capacitado para reensamblar los fragmentos. Evidentemente para que esto sea posible será necesario que el trayecto completo por el que ha de transitar ese datagrama soporte el tamaño de MTU correspondiente.
- Cuando se aplica la técnica de descubrimiento de la MTU del trayecto o path MTU discovery para averiguar la MTU de una ruta. Esta técnica consiste en que el host de origen envía un datagrama del tamaño máximo al host de destino con el bit DF puesto; si el datagrama no puede pasar en algún punto del trayecto el router correspondiente genera un mensaje de error que es devuelto al host emisor; entonces este envía otro datagrama más pequeño, también con el bit DF a 1 y así sucesivamente, hasta que consigue que algún datagrama pase sin fragmentar al host de destino; tanteando de esta forma consigue averiguar cuál es la máxima MTU de la ruta y a partir de ahí utiliza este como valor máximo para todos los datagramas.

El comando de prueba ping permite especificar la no fragmentación. Por ejemplo en los sistemas operativos Windows se utiliza para este fin la opción –f. Esto combinado con la especificación de un determinado tamaño de paquete (opción –l en Windows) permite hacer sondeos manuales de la MTU de un trayecto, lo cual resulta muy útil en la resolución de problemas o en pruebas de rendimiento.

El bit **MF** (More Fragments) puesto a 1 especifica que este datagrama es realmente un fragmento de un datagrama mayor, y que no es el último. Si está a 0 indica que este es el último fragmento o bien que el datagrama no ha sido fragmentado.

El campo **Fragment offset** sirve para indicar, en el caso de que el datagrama sea un fragmento, en qué posición del datagrama original se sitúan los datos que contiene. Los cortes siempre se realizan en múltiplo de 8 bytes, que es la unidad elemental de fragmentación, por lo que el Fragment offset cuenta los bytes en grupos de 8. Gracias a eso este contador requiere únicamente 13 bits en vez de los 16 que harían falta si contara bytes ( $2^{13}=8192$ ,  $8192 \times 8 = 65536$ ). De los tres bits que se ganan dos se utilizan en los flags DF y MF y el tercero no se utiliza.

Los fragmentos de un datagrama pueden llegar desordenados a su destino; el receptor podrá identificarlos gracias al campo Identificación. La longitud total del datagrama puede calcularla cuando recibe el último fragmento (identificado por el bit MF a 0) a partir de los campos Longitud y Fragment offset; la longitud será  $\text{fragment\_offset} * 8 + \text{longitud}$ .

Cuando se fragmenta un datagrama el host receptor retiene en su buffer los fragmentos y los reensambla cuando los ha recibido todos. Mientras mantiene retenido un fragmento el host va restando cada segundo una unidad al campo TTL. Cuando el valor de TTL es igual a cero descarta el fragmento. Si alguno de los fragmentos de un datagrama se pierde el resto terminarán desapareciendo a medida que agoten su TTL. No existe ningún mecanismo en IP que contemple el reenvío de datagramas o de fragmentos.

Suponer que se genera en una red Token Ring, un datagrama de 4000 bytes de datos (es decir 4000 bytes más la cabecera IP) que ha de pasar por una red Ethernet DIX (MTU de 1500 bytes); el resultado de la fragmentación será el siguiente:

	Campos de cabecera en datagrama IP						Datos
Datagrama Original	Id = X	L = 4020	DF = 0	MF = 0	Offset = 0	ABCDEF GHIJKL MNOP	
Fragmento 1	Id = X	L = 1500	DF = 0	MF = 1	Offset = 0	ABCDEF	
Fragmento 2	Id = X	L = 1500	DF = 0	MF = 1	Offset = 185	GHIJKL	
Fragmento 3	Id = X	L = 1060	DF = 0	MF = 0	Offset = 370	MNOP	

El primer y segundo fragmentos contendrán 1480 bytes de datos y 20 de cabecera; el tercero tendrá 1040 de datos y 20 de cabecera.

Puede suceder que un datagrama que ha sido fragmentado en un punto determinado de la ruta tenga que ser fragmentado de nuevo más tarde porque pase a otra red de MTU aun menor. Suponer, entonces, que los fragmentos 2 y 3 anteriores pasan después por una red PPP con bajo retardo (MTU de 296 bytes); el resultado será el siguiente:

	Campos de cabecera en datagrama IP						Datos
Fragmento 2	Id = X	L = 1500	DF = 0	MF = 1	Offset = 185	GHIJKL	
Fragmento 2a	Id = X	L = 292	DF = 0	MF = 1	Offset = 185	G	
Fragmento 2b	Id = X	L = 292	DF = 0	MF = 1	Offset = 219	H	
Fragmento 2c	Id = X	L = 292	DF = 0	MF = 1	Offset = 253	I	
Fragmento 2d	Id = X	L = 292	DF = 0	MF = 1	Offset = 287	J	
Fragmento 2e	Id = X	L = 292	DF = 0	MF = 1	Offset = 321	K	
Fragmento 2f	Id = X	L = 140	DF = 0	MF = 1	Offset = 355	L	

	Campos de cabecera en datagrama IP						Datos
Fragmento 3	Id = X	L = 1060	DF = 0	MF = 0	Offset = 370	MNOP	
Fragmento 3a	Id = X	L = 292	DF = 0	MF = 1	Offset = 370	M	
Fragmento 3b	Id = X	L = 292	DF = 0	MF = 1	Offset = 404	N	
Fragmento 3c	Id = X	L = 292	DF = 0	MF = 1	Offset = 438	O	
Fragmento 3d	Id = X	L = 244	DF = 0	MF = 0	Offset = 472	P	

Aquí cada fragmento (excepto el último de cada grupo) lleva 272 bytes de datos y 20 de cabecera. Aunque la MTU posible es de 296 bytes los datagramas generados son de 292 bytes porque la parte de datos de los fragmentos siempre debe ser múltiplo de 8 bytes. Después de una fragmentación múltiple solo el último fragmento del último grupo (el 3d en el ejemplo) tiene puesto a 0 el bit MF.

#### 5.4.3. DIRECCIONES IP

Una característica que define a IPv4 consiste en sus direcciones de 32 bits. Cada host y enrutador de Internet tiene una dirección IP que se puede usar en los campos *Dirección de origen* y *Dirección de destino* de los paquetes IP. Es importante tener en cuenta que una dirección IP en realidad no se refiere a un host, sino a una interfaz de red, por lo que si un host está en dos redes, debe tener dos direcciones IP.

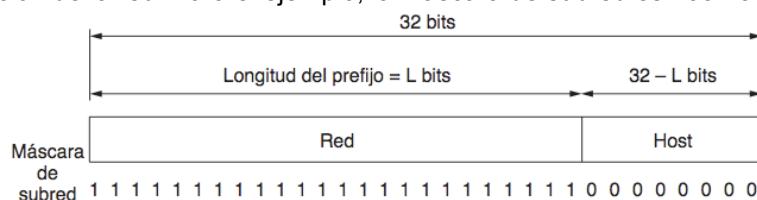
#### Prefijos

A diferencia de las direcciones Ethernet, las direcciones IP son jerárquicas. Cada dirección de 32 bits está compuesta de una porción de red de longitud variable en los bits superiores, y de una porción de host en los bits inferiores. La porción de red tiene el mismo valor para todos los hosts en una sola red, como una LAN Ethernet. Esto significa que una red corresponde a un bloque contiguo de espacio de direcciones IP. A este bloque se le llama prefijo.

Las direcciones IP se escriben en notación decimal con puntos. En este formato, cada uno de los 4 bytes se escribe en decimal, de 0 a 255. Para escribir los prefijos, se proporciona la dirección IP menor en el bloque y el tamaño del mismo. El tamaño se determina mediante el número de bits en la porción de red; el resto de los bits en la porción del host pueden variar. Esto significa que el tamaño debe ser una potencia de dos.

Por convención, el prefijo se escribe después de la dirección IP como una barra diagonal seguida de la longitud en bits de la porción de red. Como ejemplo, si el prefijo contiene 28 direcciones y, por lo tanto, deja 24 bits para la porción de red, se escribe como 128.208.0.0/24.

Como la longitud del prefijo no se puede inferir sólo a partir de la dirección IP, los protocolos de enrutamiento deben transportar los prefijos hasta los enruteadores. Algunas veces los prefijos se describen simplemente mediante su longitud, como en un “/16”. La longitud del prefijo corresponde a una máscara binaria de 1s en la porción de red. Cuando se escribe de esta forma, se denomina **máscara de subred**. Se puede aplicar un AND a la máscara de subred con la dirección IP para extraer sólo la porción de la red. Para el ejemplo, la máscara de subred es 255.255.255.0.



De los valores de los primeros bits de cada una de las clases siguientes se puede deducir el rango de direcciones que corresponde a cada una de ellas. Así pues, en la práctica es fácil saber a qué clase pertenece una dirección determinada sin más que observar el primer byte de su dirección.

Clase	Primeros bits	Bits red/host	Núm. Redes	Núm. Direcc.	Máscara	Rango de direcciones
A	0--	7/24	128	16777216	255.0.0.0	0.0.0.0 - 127.255.255.255
B	10--	14/16	16384	65536	255.255.0.0	128.0.0.0 - 191.255.255.255
C	110-	21/8	2097152	256	255.255.255.0	192.0.0.0 - 223.255.255.255
D	1110			268435456		224.0.0.0 - 239.255.255.255
E	1111					240.0.0.0 - 255.255.255.255

Para evitar conflictos, los números de red se administran a través de una corporación sin fines de lucro llamada ICANN (Corporación de Internet para la Asignación de Nombres y Números). Esta corporación ha delegado partes de este espacio de direcciones a varias autoridades regionales NICs (NIC = Network Information Center), las cuales reparten las direcciones IP a los ISP y otras compañías. Al principio había un NIC para toda la Internet pero luego se crearon NICs regionales (NorteAmérica [www.internic.net](http://www.internic.net), SudAmerica y el caribe [www.lacnic.net](http://www.lacnic.net), Europa [www.ripe.net](http://www.ripe.net), Asia y Pacífico [www.apnic.net](http://www.apnic.net)). Estos NICs asignan direcciones IP a los proveedores de Internet y a las grandes organizaciones. Los proveedores a su vez asignan direcciones a las organizaciones de menor tamaño y a sus usuarios.

Existen reglas y convenios que asignan significados especiales a determinadas direcciones IP que es importante conocer:

- La dirección **255.255.255.255** se utiliza para indicar broadcast en la propia red. Solo se puede utilizar como dirección de destino, nunca como dirección de origen.
- La dirección **0.0.0.0** identifica al host actual. Solo se puede utilizar como dirección de origen, no de destino.
- Las direcciones con el **campo host todo a ceros** identifican redes y por tanto no se utilizan para ningún host. Se emplean para especificar rutas y nunca deberían aparecer como direcciones de origen o destino de un datagrama (167.157.28.0).
- La dirección con el **campo host todo a unos** se utiliza como dirección broadcast dentro de la red y por tanto no se utiliza para ningún host. Solo puede ser una dirección de destino. (167.157.28.255)
- La dirección con el **campo red todo a ceros** identifica a un host en la propia red, cualquiera que sea (0.0.0.21)

- La dirección **127.0.0.1** se utiliza para pruebas loopback; todas las implementaciones de IP devuelven a la dirección de origen los datagramas enviados a esta dirección sin intentar enviarlos a ninguna parte.
- Las redes **127.0.0.0**, **128.0.0.0**, **191.255.0.0**, **192.0.0.0** y el rango de **240.0.0.0** en adelante (clase E) están reservados y no deben utilizarse.
- Las redes **10.0.0.0** (clase A), **172.16.0.0 a 172.31.0.0** (clase B) y **192.168.0.0 a 192.168.255.0** (clase C) están reservadas para redes privadas (intranets); estos números no se asignan a ninguna dirección válida en Internet y por tanto pueden utilizarse para construir redes locales sin conexión a internet.

#### 5.4.4. Subredes (Subnetting)

Suponiendo que una empresa dispone de varias oficinas en diferentes ciudades, cada una con una red local, todas ellas interconectadas entre sí, y que desea unirlas mediante el protocolo TCP/IP; una de dichas oficinas (la principal) dispone además de una conexión a Internet. Suponer también que la suma de todos los requerimientos de direcciones en las oficinas no supera las 254 direcciones de hosts. En principio sería posible asignar una red clase C diferente para cada oficina, pero esto supone solicitar al NIC una red para cada oficina que se conecte, y al ser cada una independiente de las demás la gestión se complica; por ejemplo sería preciso anunciar en Internet la ruta para cada nueva red para que la oficina correspondiente fuera accesible. Dado que cada red sería en principio independiente de las demás no habría una forma sencilla de agrupar las redes de la organización.

Hay un mecanismo que permite dividir una red IP en trozos o subredes, de forma que la empresa del ejemplo podría solicitar una clase C y asignar fragmentos de dicha red a cada oficina a medida que se fueran incorporando a la red. Esto equivale a crear un nivel jerárquico intermedio entre la red y el host, permitiendo así grados variables de agrupación según el nivel en el que se encuentre. Las subredes se añadieron a la Internet en 1982; con ello se consiguió una mayor flexibilidad en el reparto de direcciones dentro de una red. Para dividir la red en subredes se define una nueva *máscara*. Como siempre los bits a 1 de la máscara identifican la parte de red (en este caso la parte de red-subred) y los bits a cero corresponden al host. Por ejemplo, la máscara 255.255.255.0 aplicada sobre una red clase B la divide en 256 subredes de 256 direcciones cada una, pues tiene puestos a 1 los primeros 24 bits; en cierto modo se podría decir que esta máscara convierte una red clase B en 256 subredes clase C. Se pueden hacer divisiones que no correspondan a bytes enteros, por ejemplo la máscara 255.255.252.0 hace subredes más grandes, reserva los primeros 6 bits para la subred y deja 10 para el host, con lo que podría haber hasta 64 subredes con 1024 direcciones cada una.

Cuando se crean subredes hay dos direcciones en cada subred que quedan automáticamente reservadas: las que corresponden al campo host todo a ceros y todo a unos; estas se emplean para designar la subred y para broadcast dentro de la subred, respectivamente, siguiente la ecuación  $2^N - 2$ . Por tanto el número de hosts de una subred es siempre dos menos que el rango de direcciones que abarca. Una consecuencia de lo anterior es que resulta absurdo crear subredes con la máscara 255.255.255.254, ya que el campo host tendría un bit solamente y no quedaría ninguna dirección aprovechable para hosts. Del mismo modo que los valores todo ceros o todo unos del campo host están reservados con un significado especial, los valores todo ceros y todo unos del campo subred (es decir la primera y la última subredes) también son especiales. El valor todo ceros se utiliza para representar la subred misma. Por consiguiente en el campo subred también se pierden siempre dos direcciones, y tampoco tendría sentido crear máscaras con el campo subred de un bit, como sería el caso de una máscara 255.255.128.0 en el caso de una red clase B.

Mientras que la restricción de las direcciones todo ceros o todo unos en el campo host se ha de respetar siempre, existen muchas situaciones en las que interesa aprovechar la subred toda a ceros o toda a unos, violando la segunda norma antes mencionada. Esta violación, permitida por muchas implementaciones, se conoce como **subnet-zero** y se adopta para aprovechar mejor el espacio de direcciones disponible; con subnet-zero es posible por ejemplo dividir una red clase B por la mitad en dos subredes mediante la máscara 255.255.128.0, cosa que no sería posible si no se permitiera esta pequeña infracción. En el caso de una red clase C las posibles subredes y máscaras son las que se recogen en la tabla siguiente

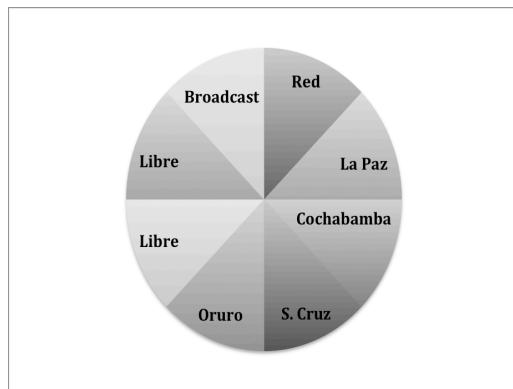
Bits de subred	Número de subredes	Nº subredes (subred cero)	Bits de host	Número de hosts	Máscara
0	0	0	8	254	255.255.255.0
1	0	2	7	126	255.255.255.128
2	2	4	6	62	255.255.255.192
3	6	8	5	30	255.255.255.224
4	14	16	4	14	255.255.255.240
5	30	32	3	6	255.255.255.248
6	62	64	2	2	255.255.255.252
7	126	128	1	0	255.255.255.254
8	254	256	0	0	255.255.255.255

Como ejemplo ilustrativo se requiere satisfacer los requerimientos de direcciones de una Empresa que tiene 4 sucursales en diferentes ciudades, y sus requerimientos de conexión en La Paz son 55, en Santa Cruz son 20, en Cochabamba son 35, en Oruro 15 y en Sucre 10 computadoras respectivamente. El ISP local le asigno la dirección clase C 200.87.52.0 con mascara 255.255.255.0. La división homogénea en subredes seria:

Bits	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Hosts	128	64	32	16	8	4	2	1

Considerando  $2^5 = 32 - 2$  hosts, se tendrían  $2^3 = 8 - 2$  subredes, que satisface la mayoría de los requerimientos de las 4 ciudades.

Ciudad	Subred	Direcciones validas	Broadcast	Submascara	Perdidas
Anulado	200.87.52.0				
La Paz	200.87.52.32	200.87.52.33 – 200.87.52.62	200.87.52.63	255.255.255.224	25
Cochabamba	200.87.52.64	200.87.52.65 – 200.87.52.94	200.87.52.95	255.255.255.224	5
Santa Cruz	200.87.52.96	200.87.52.97 – 200.87.52.126	200.87.52.127	255.255.255.224	0
Oruro	200.87.52.128	200.87.52.129 – 200.87.52.158	200.87.52.159	255.255.255.224	0
Libre	200.87.52.160				
Libre	200.87.52.192				
Anulado	200.87.52.224				



La división en subredes no ha de hacerse necesariamente de forma homogénea en todo el espacio de direcciones, como se ha visto hasta ahora. La empresa del ejemplo anterior tiene una serie de oficinas pequeñas que tienen bastante con subredes de 32 direcciones, pero otras mas grandes requieren un número mayor y según la división anterior se advierte una perdida de direcciones en algunas sucursales. La solución más adecuada en este caso sería dividir la red en subredes de diferentes tamaños y asignar a cada oficina una subred adecuada a sus necesidades. La técnica de dividir una red en subredes de diferentes

tamaños se conoce comúnmente como **máscaras de tamaño variable (VLSM)**. Para aplicar esta técnica al ejemplo anterior se debe ordenar los requerimientos de mayor a menor y hacer el calculo ciudad por ciudad, cuidando de tener direcciones libres de disponer en todo momento.

Para La Paz con necesidad de 55 direcciones, el calculo seria:

Bits	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Hosts	128	64	32	16	8	4	2	1

Considerando  $2^6 = 64 - 2$  hosts, se tendrían  $2^2 = 4 - 2$  subredes utilizables

Ciudad	Subred	Direcciones validas				Broadcast	Submascara	Perdidas
Anulado	200.87.52.0							
La Paz	200.87.52.64	200.87.52.65 – 200.87.52.126				200.87.52.127	255.255.255.192	0
Libre	200.87.52.128							
Anulado	200.87.52.192							

Para Cochabamba con necesidad de 35 direcciones, el calculo seria:

Bits	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Hosts	128	64	32	16	8	4	2	1

Considerando  $2^5 = 32 - 2$  hosts, se tendrían  $2^3 = 4 - 2$  subredes utilizables

Ciudad	Subred	Direcciones validas				Broadcast	Submascara	Perdidas
Anulado	200.87.52.0							
Anulado	200.87.52.32							
Utilizado	200.87.52.64							
Utilizado	200.87.52.96							
Cochabamba	200.87.52.128	200.87.52.129 – 200.87.52.158				200.87.52.159	255.255.255.224	5
Libre	200.87.52.160							
Anulado	200.87.52.192							
Anulado	200.87.52.224							

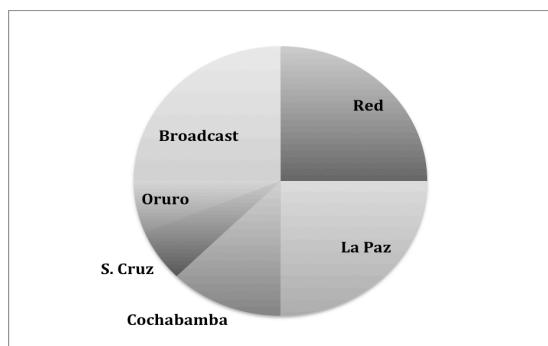
Para Santa Cruz con necesidad de 20 direcciones y Oruro con 15 direcciones, el calculo seria:

Bits	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Hosts	128	64	32	16	8	4	2	1

Considerando  $2^4 = 16 - 2$  hosts, se tendrían  $2^4 = 16 - 2$  subredes utilizables

Ciudad	Subred	Direcciones validas				Broadcast	Submascara	Perdidas
Anulado	200.87.52.0							
Anulado	200.87.52.16							
Anulado	200.87.52.32							
Anulado	200.87.52.48							
Utilizado	200.87.52.64							
Utilizado	200.87.52.80							
Utilizado	200.87.52.96							

Utilizado	200.87.52.112					
Utilizado	200.87.52.128					
Utilizado	200.87.52.144					
Santa Cruz	200.87.52.160	200.87.52.161 – 200.87.52.174	200.87.52.175	255.255.255.240	6	
Oruro	200.87.52.176	200.87.52.177 – 200.87.52.190	200.87.52.191	255.255.255.240	1	
Anulado	200.87.52.192					
Anulado	200.87.52.208					
Anulado	200.87.52.224					
Anulado	200.87.52.240					



Si se comparan los resultados de las distribuciones homogéneas y heterogéneas se verifica que las perdidas son menores en el segundo caso (12) frente a las 30 del primer caso; por lo que para este ejemplo la distribución que hace un mejor uso de las direcciones es la heterogénea.

#### 5.4.5. Superredes. Enrutamiento Interdominio sin Clases (CIDR)

El rápido crecimiento de Internet ha creado varios problemas, el más importante de los cuales es el agotamiento del espacio de direcciones IP. La causa de esto ha sido en parte la excesiva disparidad de tamaños entre las diferentes clases de redes. Hace ya mucho tiempo que han dejado de asignarse redes clase A, debido a su escaso número y tamaño excesivo. Las organizaciones tenían pues que elegir entre solicitar una clase B o una clase C. En muchos casos una clase B era excesiva, pero una C resultaba insuficiente, por lo que la mayoría de las organizaciones optaban por solicitar una clase B, aunque a menudo no necesitaban tantas direcciones. A la vista del rápido agotamiento de redes clase B debido a este motivo se pensó en crear grupos de clases C, de forma que las organizaciones pudieran optar por niveles intermedios entre las redes B y C, más adecuados a sus necesidades; por ejemplo una organización que necesite 2048 direcciones puede hoy en día solicitar un grupo de ocho redes clase C. De esta forma se reduce el problema de escasez de direcciones, pero se crea un problema nuevo: el crecimiento de las tablas de rutas. Antes, cuando a una organización que se conectaba a Internet se le asignaba una red esto suponía una nueva entrada en las tablas de rutas de Internet, pero al dar grupos de clases C se requiere una entrada diferente para cada red asignada. Esto habría provocado un crecimiento exponencial en las tablas de rutas de los routers que forman el backbone, cuyas capacidades se encuentran ya bastante cerca del límite de la tecnología.

Los dos problemas antes descritos, desperdicio del espacio de direcciones debido a la rigidez en la asignación de rangos (redes clase B o C) y crecimiento de las tablas de rutas, se resolvieron conjuntamente en 1993 con la adopción de un sistema denominado CIDR (Classless InterDomain Routing) que consiste en dos medidas complementarias. La primera medida consiste en establecer una jerarquía en la asignación de direcciones. En vez de utilizar un criterio puramente cronológico, que desde le punto de vista geográfico o de topología de la red equivale a una asignación aleatoria, los rangos se asignan por continentes. Inicialmente se realizó la asignación de una parte del espacio de clase C de la siguiente manera:

Multi regional:	192.0.0.0 - 193.255.255.255
Europa:	194.0.0.0 - 195.255.255.255
Otros:	196.0.0.0 - 197.255.255.255
Norteamérica:	198.0.0.0 - 199.255.255.255

Centro y Sudamérica:	200.0.0.0 - 201.255.255.255
Anillo Pacífico:	202.0.0.0 - 203.255.255.255
Otros:	204.0.0.0 - 205.255.255.255
Otros:	206.0.0.0 - 207.255.255.255

Algunos de estos grupos se han ampliado posteriormente con nuevos rangos. A su vez cada proveedor Internet solicita rangos propios al NIC que le corresponde según el continente donde se encuentra. Con esta distribución regional de los números es en principio posible agrupar las entradas en las tablas de rutas; por ejemplo un router en Japón podría tener una sola entrada en sus tablas indicando que todos los paquetes dirigidos a las redes 194.0.0.0 hasta 195.255.255.0 se envíen a la interfaz por la cual accede a Europa, evitando así las 131.072 entradas que normalmente harían falta para este rango de direcciones.

Para que la sumarización de rutas (o agrupamiento de redes clase C) sea posible es preciso introducir una ligera modificación en el software de los routers, ya que en principio el software no considera el rango 194.0.0.0–195.255.255.255 como una sola red sino como 131.072 redes clase C. Para resolver este problema se ha extendido el concepto de subred en sentido contrario, es decir la máscara no solo puede crecer hacia la derecha para dividir una red en subredes sino que puede menguar hacia la izquierda para agrupar varias redes en una mayor (denominada superredes). Dicho de otra forma, la parte red de la dirección vendrá especificada por la longitud de la máscara únicamente, no teniendo ya ningún significado la clasificación tradicional en clases A, B y C de acuerdo con el valor de los primeros bits; solo se respeta dicho significado en el caso de las clases D (multicast) y E (reservado). Dicha supresión de las clases tradicionales es lo que da nombre a esta técnica conocida como enrutamiento entre dominios **sin clases** o CIDR (Classless InterDomain Routing).

La segunda medida adoptada por CIDR es en realidad un complemento de la anterior. Consiste sencillamente en dar a cada organización (bien directamente o a través de su proveedor correspondiente) la posibilidad de solicitar rangos de direcciones ajustado a sus necesidades previstas, dándole siempre un rango contiguo y que tenga una máscara de red común; por ejemplo un rango de 2048 direcciones se daría asignando los primeros 11 bits de la dirección y podría estar formado por ejemplo por el rango que va del 194.0.8.0 al 194.0.15.255. La aplicación de CIDR ha permitido extender considerablemente la vida prevista inicialmente del espacio de direcciones de 32 bits de IPv4.

Por ejemplo, si la UMSS requiere 22000, la UMSA 15000, la UAGRM 4000 y la UTO 1000 direcciones respectivamente, y un grupo de direcciones que empieza en 166.114.0.0/16 el calculo de distribución CIDR sería el siguiente:

Para la UMSS que requiere 22000:

Bits Red	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Redes clase C	128	64	32	16	8	4	2	1
Bits Hosts	$2^{15}$	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$
Número Hosts	32768	16382	8192	4096	2048	1024	512	256

Se precisa  $32 - 15 = 17$  bits de máscara, lo que supone asignarle 32768 hosts en 128 redes clase C.

Universidad	Superred	Direcciones válidas	Broadcast
UMSS	166.114.0.0/17	166.114.0.1 – 166.114.127.254	166.114.127.255

Comprobando que la máscara (bit 17) cubre el rango de direcciones asignado:

0 = 0 | 0000000 ✓

127 = 0 | 1111111

Para la UMSA que requiere 15000:

Bits Red	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Redes clase C	128	64	32	16	8	4	2	1
Bits Hosts	$2^{15}$	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$
Numero Hosts	32768	16382	8192	4096	2048	1024	512	256

Se precisa  $32 - 14 = 18$  bits de mascara, lo que supone asignarle 16382 hosts en 64 redes clase C.

Universidad	Superred	Direcciones validas	Broadcast
UMSA	166.114.128.0/18	166.114.128.1 – 166.114.191.254	166.114.192.255

Comprobando que la mascara (bit 18) cubre el rango de direcciones asignado:

128 = 10 | 000000 ✓

191 = 10 | 111111

Para la UAGRM que requiere 4000:

Bits Red	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Redes clase C	128	64	32	16	8	4	2	1
Bits Hosts	$2^{15}$	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$
Numero Hosts	32768	16382	8192	4096	2048	1024	512	256

Se precisa  $32 - 12 = 20$  bits de mascara, lo que supone asignarle 4096 hosts en 16 redes clase C.

Universidad	Superred	Direcciones validas	Broadcast
UAGRM	166.114.192.0/20	166.114.192.1 – 166.114.207.254	166.114.207.255

Comprobando que la mascara (bit 20) cubre el rango de direcciones asignado:

192 = 1100 | 0000 ✓

207 = 1100 | 1111

Para la UTO que requiere 1000:

Bits Red	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Redes clase C	128	64	32	16	8	4	2	1
Bits Hosts	$2^{15}$	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$
Numero Hosts	32768	16382	8192	4096	2048	1024	512	256

Se precisa  $32 - 10 = 22$  bits de mascara, lo que supone asignarle 1024 hosts en 4 redes clase C.

Universidad	Superred	Direcciones validas	Broadcast
UTO	166.114.208.0/20	166.114.208.1 – 166.114.211.254	166.114.211.255

Comprobando que la mascara (bit 22) cubre el rango de direcciones asignado:

208 = 110100 | 00 ✓

211 = 110100 | 11

#### 5.4.6. NAT: Traducción de Dirección de Red

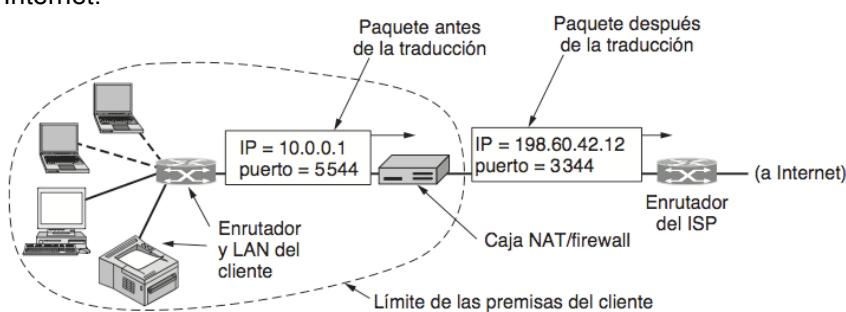
Las direcciones IP son escasas. Un ISP podría tener una dirección con prefijo de /16, lo cual le da 65534 direcciones de host. Si tiene más clientes que esos, tiene un problema. Esta escasez ha conducido a técnicas para usar las direcciones IP con moderación. Un método es asignar dinámicamente una dirección IP a una computadora cuando ésta se encuentra encendida y usa la red, y tomar de vuelta la dirección IP cuando el host se vuelve inactivo. Así, la dirección IP se puede asignar a otra computadora que se active en ese momento. De esta manera, una sola dirección de /16 puede manejar hasta 65 534 usuarios activos.

Esta estrategia funciona bien en algunos casos, por ejemplo, para las redes de marcación telefónica, las computadoras móviles y otras computadoras que pueden estar temporalmente ausentes o apagadas. Sin embargo, no funciona muy bien para los clientes comerciales. Estos negocios tienen una línea de acceso que siempre proporciona conectividad al resto de Internet.

Esta situación se aplica cada vez más a los usuarios domésticos que se suscriben a ADSL o Internet por cable, ya que no hay un cargo por conexión (sólo una tarifa mensual fija). Muchos de estos usuarios tienen dos o más computadoras en su hogar, a menudo una para cada miembro de la familia, y todos quieren estar en línea todo el tiempo. La solución es conectar todas las computadoras en una red doméstica a través de una LAN y colocar un enrutador (inalámbrico) en ella. Así, el enrutador se conecta con el ISP. Desde el punto de vista del ISP, la familia es ahora lo mismo que un pequeño negocio con un puñado de computadoras. Con las técnicas que hemos visto hasta ahora, cada computadora debe tener su propia dirección IP todo el tiempo. Para un ISP con muchos miles de clientes, en especial clientes comerciales y familias que son casi como pequeños negocios, la demanda de direcciones IP puede exceder rápidamente el bloque disponible.

La solución, que se utiliza ampliamente en la actualidad, se conoce como NAT (Traducción de Dirección de Red).

La idea básica detrás de NAT es que el ISP asigne a cada hogar o negocio una sola dirección IP (o a lo más, una pequeña cantidad de éstas) para el tráfico de Internet. Dentro de la red del cliente, cada computadora obtiene una dirección IP única, la cual se utiliza para enrutar el tráfico interno. Sin embargo, justo antes de que un paquete salga de la red del cliente y vaya al ISP, la dirección IP única interna se traduce a la dirección IP pública compartida. Esta traducción hace uso de los tres rangos de direcciones IP que se han declarado como privados (10.0.0.0-10.255.255.255/8, 172.16.0.0-172.31.255.255/12, 192.168.0.0-192.168.255.255/16). Las redes pueden utilizarlos de manera interna como deseen. La única regla es que no pueden aparecer paquetes que contengan estas mismas direcciones en Internet.



#### 5.4.7. PROTOCOLOS DE CONTROL DE INTERNET

Todo el tráfico de Internet está formado por datagramas IP. Normalmente los datagramas transportan TPDUs (Transport Protocol Data Unit) de TCP o UDP, que son los dos protocolos de transporte utilizados en Internet. Todas las aplicaciones de interés para el usuario de Internet (correo electrónico, transferencia de archivos, videoconferencia, etc.) generan tráfico TCP o UDP. Sin embargo en la estructura de la cabecera de un datagrama el valor del campo protocolo se observa

que existen muchos posibles significados del contenido de un datagrama IP, aparte de los normales que serían TCP y UDP. Algunos de los datos que pueden transportarse en datagramas IP son mensajes de protocolos de control de Internet, que son una parte necesaria para el correcto funcionamiento de la red (ICMP, ARP, RARP, BOOTP y DHCP).

#### 5.4.7.1. ICMP (Protocolo de mensajes de Control en Internet)

En el funcionamiento normal de una red se dan a veces situaciones extraordinarias que requieren enviar avisos especiales. El mecanismo para reportar todos estos incidentes en Internet es el protocolo conocido como ICMP. Los mensajes ICMP viajan por la red como datagramas IP (con el valor 1 en el campo protocolo), sujetos en los routers a las mismas reglas que cualquier otro datagrama. Los mensajes ICMP son generados por el host o router que detecta el problema o situación extraordinaria y dirigidos al host o router que aparece en el campo dirección origen del datagrama que causó el problema. Para facilitar la identificación del datagrama por parte del host emisor la mayoría de los mensajes ICMP incluyen, además del código de error correspondiente, la cabecera y los primeros ocho bytes de datos del datagrama original. La tabla siguiente resume los mensajes ICMP más importantes:

Tipo de mensaje	Descripción
<i>Destination unreachable</i> (Destino inaccesible).	No se pudo entregar el paquete.
<i>Time exceeded</i> (Tiempo excedido).	El tiempo de vida llegó a cero.
<i>Parameter problem</i> (Problema de parámetros).	Campo de encabezado inválido.
<i>Source quench</i> (Fuente disminuida).	Paquete regulador.
<i>Redirect</i> (Redireccionar).	Enseña a un enrutador la geografía.
<i>Echo and echo reply</i> (Eco y respuesta de eco).	Verifica si una máquina está viva.
<i>Timestamp request/reply</i> (Estampa de tiempo, Petición/respuesta).	Igual que solicitud de eco, pero con marca de tiempo.
<i>Router advertisement/solicitation</i> (Enrutamiento anuncio/solicitud).	Busca un enrutador cercano.

#### 5.4.7.2. ARP (Protocolo de Resolución de direcciones)

Aunque en Internet cada máquina tiene una o más direcciones IP, en realidad éstas no son suficientes para enviar paquetes. Las NIC (Tarjetas de Interfaz de Red) de la capa de enlace de datos no entienden las direcciones de Internet. En el caso de Ethernet, cada NIC de las que se hayan fabricado viene equipada con una dirección Ethernet única de 48 bits. Los fabricantes de NIC Ethernet solicitan un bloque de direcciones Ethernet al IEEE para asegurar que no haya dos NIC con la misma dirección. Las NIC envían y reciben tramas basadas en direcciones Ethernet de 48 bits. No saben nada sobre direcciones IP de 32 bits.

Suponiendo que una PC conectada a una red local Ethernet mediante una tarjeta configurada con la dirección IP 147.156.1.2 y la máscara 255.255.0.0; quiere iniciar una sesión de terminal remoto telnet con una computadora cuya dirección IP es 147.156.1.3. En el momento en que se teclea *telnet 147.156.1.3* el software del PC compara la dirección de destino con la suya propia y con su máscara y deduce que el computador de destino se encuentra en la misma red local. El PC genera entonces un mensaje ARP con la pregunta ¿quién tiene la dirección 147.156.1.3? y lo envía en una trama Ethernet que tiene como dirección MAC de destino la dirección broadcast (todo a unos); la trama es recibida y procesada por todas las máquinas de la red que en ese momento estén activas, siendo retransmitida a través de los commutadores o puentes transparentes locales o remotos que haya. Eventualmente una máquina (y solo una) se reconoce propietaria de la dirección IP solicitada y responde al mensaje; la respuesta puede ser y normalmente será una trama unicast puesto que el computador de destino ya conoce la dirección MAC del PC que lanzó la pregunta; la respuesta incluye la dirección MAC solicitada, por lo que a partir de ese momento ambos computadores pueden comunicarse mediante tramas unicast, de forma que la generación de tráfico broadcast queda estrictamente limitada al primer mensaje enviado por el PC inicial.

Cada computador de la red local mantiene en memoria una tabla denominada *ARP cache* con las parejas de direcciones MAC-IP utilizadas recientemente. Generalmente cuando un computador envía un mensaje ARP buscando a otro *todas* las máquinas de la red, no sólo la destinataria del mensaje, aprovechan para registrar al emisor, anotándolo en su ARP cache. De esta forma si más tarde necesitan contactar con dicha máquina podrán hacerlo directamente, sin necesidad de enviar un mensaje ARP broadcast. Las entradas en la ARP cache expiran pasados unos minutos sin que haya

tráfico con la máquina correspondiente, para permitir que se produzcan cambios en la tabla por ejemplo por avería de una tarjeta de red o cambio de la dirección IP de un host. El comando `arp -a`, disponible en UNIX y en Windows, permite conocer en cualquier momento la tabla ARP cache disponible en un host.

A nivel Ethernet, ARP es un protocolo diferente de IP, con un valor de Ethertype de 806 (hexadecimal). Esto permite que los routers no confundan los paquetes ARP con los paquetes IP y no los propaguen; de este modo los paquetes ARP quedan siempre confinados en red local donde se producen, evitando así que el tráfico broadcast que generan se propague a otras redes.

Para poder adaptarse a cualquier tipo de red broadcast y a cualquier protocolo a nivel de red el paquete ARP prevé el uso de longitudes arbitrarias tanto de la dirección de red como de la dirección de enlace. A título de ejemplo la tabla siguiente muestra el formato del paquete ARP para el caso más habitual, que corresponde a una red con direcciones MAC de 6 bytes y direcciones IP de 4 bytes. Los campos se describen a continuación:

Campo	Longitud (Bytes)
Tipo de hardware	2
Tipo de protocolo	2
Long. Dirección hardware	1
Long. Dirección red	1
Código operación	2
Dirección MAC emisor	6
Dirección IP emisor	4
Dirección MAC destino	6
Dirección IP destino	4

**Tipo de hardware:** especifica el tipo de red local, por ejemplo el código 1 identifica Ethernet.

**Tipo de protocolo:** especifica el protocolo de red utilizado. Se emplean los mismos códigos que en el Ethertype (por ejemplo x0800 en el caso de IP).

**Long. Dirección hardware:** se especifica en bytes. Por ejemplo en el caso de direcciones MAC la longitud es de seis bytes.

**Long. Dirección red:** también en bytes, por ejemplo cuatro en el caso de IP.

**Código operación:** vale 1 en el caso de una pregunta ARP (¿quién tiene la dirección IP tal?) y 2 en el de una respuesta.

#### 5.4.7.3. DHCP (Protocolo de Configuración Dinámica de Host)

ARP (así como los demás protocolos de Internet) asume que los hosts están configurados con cierta información básica, como sus propias direcciones IP. ¿Cómo obtienen los hosts esta información? Es posible configurar en forma manual cada computadora, pero es un proceso tedioso y propenso a errores. Existe una mejor manera de hacerlo, conocida como **DHCP** (Protocolo de Configuración Dinámica de Host).

Con DHCP, cada red debe tener un servidor DHCP responsable de la configuración. Al iniciar una computadora, ésta tiene integrada una dirección Ethernet u otro tipo de dirección de capa de enlace de datos en la NIC, pero no cuenta con una dirección IP. En forma muy parecida al ARP, la computadora difunde una solicitud de una dirección IP en su red. Para ello usa un paquete llamado DHCP DISCOVER. Este paquete debe llegar al servidor DHCP. Si el servidor no está conectado directamente a la red, el enrutador se configurará para recibir difusiones DHCP y transmitirlas al servidor DHCP en donde quiera que se encuentre.

Cuando el servidor recibe la solicitud, asigna una dirección IP libre y la envía al host en un paquete DHCP OFFER (que también se puede transmitir por medio del enrutador). Para que esto pueda funcionar incluso cuando los hosts no tienen direcciones IP, el servidor identifica a un host mediante su dirección Ethernet (la cual se transporta en el paquete DHCP DISCOVER).

Un problema que surge con la asignación automática de direcciones IP de una reserva es determinar qué tanto tiempo se debe asignar una dirección IP. Si un host sale de la red y no devuelve su dirección IP al servidor DHCP, esa dirección se perderá en forma permanente. Después de un tiempo, tal vez se pierdan muchas direcciones. Para evitar que eso ocurra, la asignación de direcciones IP puede ser por un periodo fijo de tiempo, una técnica conocida como **arrendamiento**

**(leasing).** Justo antes de que expire el arrendamiento, el host debe pedir una renovación al DHCP. Si no puede hacer una solicitud o si ésta se rechaza, tal vez el host ya no pueda usar la dirección IP que recibió antes.

Se utiliza ampliamente en Internet para configurar todo tipo de parámetros, además de proporcionar a los hosts direcciones IP, como la máscara de red, la dirección IP de la puerta de enlace predeterminada y las direcciones IP de los servidores DNS y de tiempo. DHCP ha reemplazado en gran parte los protocolos anteriores (conocidos como RARP y BOOTP), con una funcionalidad más limitada.

#### 5.4.8. PROTOCOLOS DE ROUTING

La Internet está formada por multitud de redes interconectadas, pertenecientes a diversas empresas y organizaciones. Todas estas redes interconectadas comparten a nivel de red el protocolo IP. Al margen de esta interoperabilidad existen dos aspectos fundamentales en los que las redes pueden diferir entre sí:

- **El protocolo de routing utilizado:** existen como veremos multitud de protocolos de routing diferentes, unos basados en el algoritmo del vector distancia y otros en el del estado del enlace; incluso utilizando el mismo algoritmo se pueden emplean protocolos diferentes. Aun utilizando el mismo algoritmo y protocolo de routing dos proveedores diferentes normalmente no querrán que sus routers intercambien entre sí la misma información de routing que intercambian internamente.
- **La política de intercambio de tráfico:** un proveedor puede tener acuerdos bilaterales o multilaterales para intercambiar tráfico con otros, pero normalmente no estará dispuesto a ser utilizado como vía de tránsito para el tráfico entre dos proveedores si esto no está expresamente recogido en los acuerdos, aun cuando desde el punto de vista de topología de la red pueda ser ese el camino más corto entre ambas.

##### 5.4.8.1. Sistema Autónomo

Sistema Autónomo (AS, Autonomous System) es la subred que es administrada o gestionada por una autoridad común, que tiene un protocolo de routing homogéneo mediante el cual intercambia información en toda la subred y que posee una política común para el intercambio de tráfico con otras redes o sistemas autónomos. Normalmente cada ISP (Internet Service Provider) constituye su propio sistema autónomo. Los sistemas autónomos reciben números de dos bytes que se registran en el IANA de forma análoga a las direcciones IP. De la misma forma que existen unos rangos de direcciones IP reservados para redes privadas existe un rango de números de sistemas autónomos reservados para sistemas autónomos privados, que son los que van del 64512 al 65535.

En Internet se dan dos niveles jerárquicos de routing, el que se realiza dentro de un sistema autónomo (AS) y el que se efectúa entre sistemas autónomos. El primero se denomina routing interno y al routing entre sistemas autónomos se denomina routing externo. Dado que los requerimientos en uno y otro caso son muy diferentes, se utilizan protocolos de routing distintos. Los protocolos de routing dentro del sistema autónomo se denominan IGP (Interior Gateway Protocol), mientras que los utilizados entre sistemas autónomos se llaman EGP (Exterior Gateway Protocol).

##### 5.4.8.2. Protocolos de routing interno (IGP)

En Internet se usan actualmente diversos protocolos de routing interno. Estos pueden agruparse en protocolos de vector distancia entre los que se destaca RIP, RIPv2, IGRP y EIGRP, y protocolos del estado del enlace de los que los más importantes son IS-IS y OSPF.

##### RIP y RIPv2

RIP (Routing Information Protocol) es uno de los protocolos de routing más antiguos y deriva del protocolo de routing de XNS (Xerox Network Systems); RIP sufre los problemas típicos de los algoritmos basados en el vector distancia, tales como la cuenta a infinito, etc. Además RIP arrastra otros problemas que son consecuencia de ser un protocolo de routing muy antiguo, como son:

- Métricas basadas exclusivamente en número de saltos
- No soporta subredes ni máscaras de tamaño variable (si en RIPv2).
- No permite usar múltiples rutas simultáneamente.
- Se genera una gran cantidad de información de routing que se intercambia cada 30 segundos. Con el paso del tiempo los routers tienden a sincronizarse de forma que todos acaban enviando los paquetes a la vez; esto provoca congestión y bloqueos en la red durante el momento en que se intercambia la información de routing.

Algunos de estos problemas aumentan a medida que crece el tamaño de los sistemas autónomos, por lo que en la práctica no es aconsejable usar RIP en ninguna red que tenga más de 5 a 10 routers. A pesar de todos sus inconvenientes RIP aún se utiliza en algunas partes de Internet. Existen dos versiones de RIP: la versión 1, que se definió en el RFC 1058 y se publicó en 1983 y su uso está desaconsejado. En vista de la popularidad de RIP y de los muchos problemas que presentaba en 1993 se publicó RIP versión 2, que intentaba resolver al menos algunos de ellos (RFC 1388).

## **IGRP y EIGRP**

A pesar de sus inconvenientes, el routing por vector distancia tiene algunos serios partidarios. Quizá el más importante sea la empresa Cisco, actualmente el principal fabricante de routers en el mundo. En 1988, cuando el único protocolo de routing estandarizado y ampliamente utilizado era RIP, Cisco optó por crear un protocolo de routing propio denominado IGRP (Interior Gateway Routing Protocol) para resolver algunos de los problemas que presentaba RIP. IGRP está basado también en el vector distancia. Cisco siguió (y sigue) apostando por los protocolos de routing basados en el vector distancia ya que en 1993 produjo un nuevo protocolo denominado EIGRP (Enhanced IGRP) que introducía mejoras importantes respecto a IGRP, pero basado también en el vector distancia. Hay que resaltar que tanto IGRP como EIGRP son protocolos propietarios, y no hay implementaciones de ellos para equipos de otros fabricantes, por lo que el uso de estos protocolos requiere que todos los routers del sistema autónomo correspondiente sean de Cisco. Los routers Cisco también pueden funcionar con protocolos estándar, tales como RIP OSPF e IS-IS.

## **OSPF**

La respuesta del IETF a los problemas de RIP fue OSPF (Open Shortest Path First), protocolo de routing basado en el estado del enlace. OSPF fue desarrollado entre 1988 y 1990, y en 1991 ya se había producido OSPF versión 2. OSPF está basado en IS-IS y muchos de los conceptos que maneja son comunes a ambos protocolos. Es un estándar Internet y es el protocolo actualmente recomendado por el IAB para sustituir a RIP. Su complejidad es notablemente superior, mientras que la descripción de RIP ocupa menos de 20 páginas la especificación de OSPF emplea más de 200. La especificación vigente de OSPF está en el RFC 2328.

Entre las características más notables de OSPF se pueden destacar las siguientes:

- Es un algoritmo dinámico autoadaptativo, que reacciona a los cambios de manera automática y rápida.
- Soporta una diversidad de parámetros para el cálculo de la métrica, tales como capacidad (ancho de banda), retardo, etc.
- Realiza balance de carga si existe más de una ruta hacia un destino dado.
- Establece mecanismos de validación de los mensajes de routing, para evitar que un usuario malintencionado envíe mensajes engañosos a un router.
- Soporta rutas de red, de subred y de host.
- Se contempla la circunstancia en la que dos routers se comuniquen directamente entre sí sin que haya una línea directa entre ellos, por ejemplo cuando están conectados a través de un túnel.

OSPF permite dos niveles de jerarquía creando lo que se denominan áreas dentro de un sistema autónomo. De esta forma un router sólo necesita conocer la topología e información de routing correspondiente a su área, con lo que la cantidad de información de routing se reduce. En redes complejas esta es una característica muy valiosa.

Los algoritmos de routing por el estado del enlace se aplican dentro de cada área. En todo Sistema Autónomo (AS) hay al menos un área, el área 0 denominada backbone. Un router puede pertenecer simultáneamente a dos o más áreas, en cuyo caso debe disponer de la información de routing y ejecutar los cálculos correspondientes a todas ellas. Al menos un router de cada área debe estar además en el backbone, para conectar dicha área con el resto del Sistema Autónomo. Dos áreas sólo pueden hablar entre sí a través del backbone.

En OSPF se contemplan cuatro clases de routers:

- Routers *backbone*; son los que se encuentran en el área 0 ó backbone.
- Routers *internos*; los que pertenecen únicamente a un área.
- Routers *frontera de área*; son los que están en más de un área, y por tanto las interconectan (una de las áreas interconectadas siempre es necesariamente el backbone).
- Routers *frontera de Sistema Autónomo*; son los que intercambian tráfico con routers de otros Sistemas Autónomos. Estos routers pueden estar en el backbone o en cualquier otra área.

Existen tres tipos de rutas: intra-área, inter-área e inter-AS. Las rutas intra-área son determinadas directamente por cualquier router, pues dispone de toda la información; las rutas inter-área se resuelven en tres fases: primero ruta hacia el backbone, después ruta hacia el área de destino dentro del backbone, y por último ruta hacia el router deseado en el área de destino.

Para el intercambio de información cada router envía cuando arranca unos mensajes de salutación, denominados HELLO, por todas sus interfaces. Este y otros mensajes los reenvía periódicamente para asegurarse de que las líneas permanecen operativas. Con la información que posee y la recabada en respuesta a sus mensajes el router calcula las rutas óptimas para cada destino en cada momento.

Cuando en una red local hay varios routers resulta poco eficiente que cada uno intercambie toda la información de routing que posee con todos los demás, ya que mucha de la información sería redundante. En una red local con  $n$  routers esto produciría  $(n^2-n)/2$  intercambios de información diferentes. En estos casos OSPF prevé que uno de los routers se convierta en el *router designado*, siendo éste el único que intercambiará información con todos los demás. De esta forma el número de intercambios de información se reduce a  $n-1$ .

## IS-IS

El protocolo de routing IS-IS (Intermediate System-Intermediate System) está basado en el algoritmo del estado del enlace; además IS-IS permite hacer routing integrado, es decir calcular las rutas una vez y aplicarlas para todos los protocolos utilizados, permitiendo así auténtico routing multiprotocolo. Soporta hasta ocho niveles jerárquicos, para reducir así la cantidad de información de routing intercambiada. IS-IS fue diseñado para el protocolo DECNET (de Digital) y adoptado después por ISO como protocolo de routing para el protocolo de red CLNP. Una variante de IS-IS se utiliza en Netware de Novell. IS-IS también se utiliza en algunas zonas de la Internet. El protocolo IS-IS se especifica en el RFC 1142.

IS-IS no es un estándar Internet. Actualmente es el protocolo utilizado en las redes grandes, por ejemplo la gran mayoría de las redes de los ISPs utilizan IS-IS en lugar de OSPF.

### 5.4.8.3. Protocolos de routing externo

Todos los protocolos de routing hasta ahora descritos se emplean dentro de sistemas autónomos. Normalmente un sistema autónomo corresponde a una subred que tiene una entidad común desde el punto de vista administrativo y de gestión, puede ser por ejemplo la red de una gran empresa, de un proveedor de servicios Internet o la red académica de un país como RedIRIS. En estos casos se supone que el protocolo de routing ha de buscar la ruta óptima atendiendo únicamente al criterio de minimizar la “distancia” medida en términos de la métrica elegida para la red.

La selección de rutas para el tráfico entre sistemas autónomos plantea un problema diferente, ya que la cuestión no se reduce a la selección de la ruta óptima sino que se debe atender a criterios externos que obedezcan a razones de tipo político, económico, administrativo, etc. (recordemos que se trata de decidir el routing entre redes que pertenecen a organizaciones diferentes). Un ejemplo típico de

este tipo de restricciones es el caso en que la ruta óptima entre dos sistemas autónomos, X e Y, pasa por un tercero Z que no desea que su red sea utilizada como vía de tránsito. Para dar cabida a la utilización de criterios externos en el cálculo de las rutas entre sistemas autónomos se utilizan en este caso otro tipo de protocolos de routing, denominados protocolos de routing externo.

Hasta 1990 se utilizaba como protocolo de routing externo en la Internet el denominado EGP (Exterior Gateway Protocol), diseñado entre 1982 y 1984. Como era de esperar EGP no fue capaz de soportar la enorme evolución que sufrió Internet y como ya era habitual el IETF desarrolló un nuevo protocolo de routing externo, denominado BGP (Border Gateway Protocol). La primera especificación de BGP apareció en 1989; desde entonces el IETF ha producido cuatro versiones de BGP; las especificaciones actualmente vigentes de BGP-4 se encuentran en el RFC 1771.

Los routers que utilizan BGP (pertenecientes a diferentes ASes) forman entre ellos una red e intercambian información de routing para calcular las rutas óptimas; se utiliza el vector distancia, pero para evitar el problema de la cuenta a infinito la información intercambiada incluye, además de los routers accesibles y el costo, la ruta completa utilizada para llegar a cada posible destino; de esta forma el router que recibe la información puede descartar las rutas que pasan por él mismo que son las que podrían dar lugar al problema de la cuenta a infinito. La especificación de la ruta completa permite también a los routers revisar si dicha ruta es conforme con las políticas que se hayan especificado en cuanto a tránsito por otros sistemas autónomos.

BGP permite introducir manualmente restricciones o reglas de tipo “político”; éstas se traducen en que cualquier ruta que viola la regla recibe automáticamente una distancia de infinito.

Para simplificar la gestión de los Sistemas Autónomos se crean Confederaciones de Sistemas Autónomos; una confederación es vista como un único Sistema Autónomo desde el exterior. Esto equivale a introducir en el protocolo de routing externo dos niveles jerárquicos, con lo que se reduce la información de routing de forma análoga a lo que ocurría con las áreas de OSPF dentro de un Sistema Autónomo.

#### 5.4.8.4. Puntos neutros de interconexión

Cuando dos ISPs están conectados a la Internet en principio siempre es posible el intercambio de información entre ellos. Sin embargo este intercambio no siempre ocurre por el camino óptimo. Por ejemplo, si en Bolivia dos ISPs contratan conectividad Internet, uno a Entel y el otro a AXS, su intercambio de tráfico se realizará normalmente a través de algún punto de interconexión fuera de Bolivia, lo cual no es óptimo ya que consume costosos recursos de líneas internacionales. La solución a este problema es la realización de un acuerdo bilateral entre los dos proveedores (ENTEL y AXS), de forma que se establezca un enlace directo entre sus sistemas autónomos en Bolivia para que puedan intercambiar tráfico sin necesidad de salir fuera. Cuando el número de proveedores crece la cantidad de enlaces que hay que establecer aumenta de forma proporcional a  $(n^2-n)/2$  donde  $n$  es el número de proveedores que intercambian tráfico<sup>1</sup>. Para reducir el número de enlaces se suelen crear los denominados puntos neutros de interconexión, consistentes en un nodo normalmente gestionado por una entidad independiente (para garantizar su “neutralidad”) al cual se conectan los proveedores que desean participar. En principio en el ámbito del punto neutro el intercambio de tráfico debería ocurrir sin restricciones entre todos los proveedores; sin embargo en la práctica los proveedores han de establecer acuerdos bilaterales, por lo que no siempre dos proveedores conectados a un mismo punto neutro intercambian tráfico.

La implementación de un punto neutro es muy sencilla. Se trata básicamente de un conmutador LAN Ethernet/Fast Ethernet en el que a cada proveedor se le asigna una LAN a la que puede conectar sus routers. Los routers de diferentes proveedores intercambian información de routing mediante BGP. Cada proveedor ha de conseguir los medios necesarios para conectar su red al punto neutro (líneas dedicadas, etc.). Físicamente las instalaciones de un punto neutro han de cumplir unos requisitos de fiabilidad y seguridad muy altos, ya que se trata de un elemento crítico en el funcionamiento de la red.

<sup>1</sup> El problema que se plantea es en cierto modo parecido al que se da cuando varios routers de una misma red local participan en un protocolo de routing, problema que OSPF resolvía mediante el mecanismo del router designado.

#### 5.4.8.5. Routing Multicast

Las direcciones IP clase D (entre 224.0.0.0 y 239.255.255.255) están reservadas para tráfico multicast. Se pueden crear dos tipos de grupos multicast, temporales y permanentes. Algunos grupos permanentes que están ya predefinidos son los siguientes:

224.0.0.1	Todos los hosts en una LAN
224.0.0.2	Todos los routers en una LAN
224.0.0.5	Todos los routers OSPF en una LAN
224.0.0.6	Todos los routers OSPF designados en una LAN

Los grupos temporales se han de crear cada vez que se van a utilizar. Un host (más exactamente un proceso en un host) puede solicitar unirse a un grupo multicast (join), o puede decidir abandonarlo (leave). Cuando el último proceso de un host abandona un grupo multicast el host mismo abandona el grupo. En Internet el protocolo que gestiona todas las operaciones relacionadas con los grupos multicast es el IGMP (Internet Group Management Protocol). Para captar una emisión multicast es preciso formar parte del grupo correspondiente, pero no es necesario estar en dicho grupo para realizar una emisión multicast hacia él.

En una LAN, al ser el medio físico intrínsecamente broadcast, no es necesaria ninguna acción especial para permitir el tráfico multicast; los paquetes viajan por la red y los hosts capturan los que corresponden a los grupos multicast a los que pertenecen. El router multicast de la LAN pregunta aproximadamente una vez por minuto a los hosts de su LAN (dirección 224.0.0.1) en qué grupos multicast están interesados. Los hosts devuelven la relación de direcciones clase D en las que están interesados. De acuerdo con las respuestas que recibe el router van actualizando el árbol de distribución, añadiendo o suprimiendo (podando) ramas de éste. Si el detecta que alguna dirección multicast ha dejado de interesar (es decir, ya no hay miembros de ese grupo en la LAN) envía un mensaje al router siguiente en el árbol solicitándole le “pode”, es decir le suprime de la distribución. Inversamente puede también solicitar su adhesión a un grupo en el que antes no estuviera.

En enlaces punto a punto los routers han de revisar regularmente el árbol de distribución multicast, a fin de “podar” de éste las ramas innecesarias e incluir las que presenten nuevos miembros del grupo multicast; de esta forma se evita cargar las líneas con tráfico innecesario.

La resolución de direcciones multicast en redes locales no se realiza mediante el protocolo ARP, ya que no tendría sentido que el emisor multicast tuviera que preguntar a los miembros del grupo cual es la dirección MAC multicast correspondiente a una dirección IP determinada. En este caso se aplica la técnica de emplear un algoritmo que permita deducir la dirección MAC a partir de la dirección IP (RFC 1112): la dirección IP multicast está representada por 28 bits (ya que los cuatro primeros siempre son 1110) por lo que se podría trasladar por ejemplo a los 28 bits menos significativos de la dirección MAC, fijando los 20 primeros. Dado que el OUI (Organizationally Unique Identifier) ocupa los 24 primeros bits de la dirección MAC para poder utilizar cuatro de estos bits en las direcciones multicast habría sido necesario reservar para este fin 16 valores consecutivos de OUI, cosa que llegó a ser propuesta por el IETF al IEEE, pero no fue considerada aceptable por éste. En su lugar se decidió utilizar la mitad del OUI que había reservado para el IETF (01.00.5E), dejando así 23 bits libres en los que se mete la parte menos significativa de la dirección IP multicast de 28 bits. La correspondencia por tanto no es biunívoca, ya que existen 32 direcciones IP multicast diferentes por cada dirección MAC multicast; por ejemplo las direcciones IP multicast 224.0.0.1 y 224.128.0.1 que tienen iguales los 23 últimos bits se mapean ambas en la dirección MAC multicast 01.00.5E.00.00.80<sup>2</sup>. Podrá por tanto ocurrir que en una misma LAN dos grupos IP multicast diferentes utilicen la misma dirección MAC multicast; en este caso algunos hosts capturarán tramas multicast que luego, al examinar la cabecera IP, descubrirán que no iban dirigidas a ellos; esto produce una pequeña pérdida de eficiencia por el tiempo que el nivel de red emplea en analizar una trama que no iba dirigida a él, pero la probabilidad de que esto ocurra en la práctica es tan pequeña que la pérdida de eficiencia es despreciable<sup>3</sup>. Aun cuando se produzca coincidencia de direcciones MAC entre dos grupos diferentes nunca debe producirse por este motivo la entrega al nivel de transporte de datagramas que no vayan dirigidos al grupo multicast al que pertenece el host.

<sup>2</sup> Es importante recordar en este punto que las direcciones MAC en Ethernet se representan empezando por el bit menos significativo de cada byte. En cambio en Token Ring y FDDI la representación es al contrario.

<sup>3</sup> Suponiendo un reparto aleatorio la probabilidad de coincidencia entre dos grupos multicast sería de una en diez millones ( $2^5 / 2^{28} = 0,0000001$ )

El soporte de tráfico multicast en routers es relativamente reciente, y existen aún pocas experiencias de redes con tráfico multicast en redes de área extensa en producción, casi todas ellas limitadas a redes académicas.

La mayor experiencia de routing multicast que ha habido en la Internet es la conocida como MBone (Multicast Backbone). La red MBone existe desde 1992 y se emplea principalmente con un conjunto de aplicaciones de videoconferencia. Las reuniones del IETF, congresos y todo tipo de eventos se transmiten regularmente por MBone. Hay retransmisiones de ámbito mundial, continental, nacional y regional.

A veces se quiere interconectar dos redes con tráfico multicast que están conectadas entre sí mediante una red que solo soporta tráfico unicast. En estos casos es posible construir un túnel entre dos routers de las redes multicast y enviar los datagramas multicast encapsulados en paquetes unicast; dado que estos paquetes son vistos como tráfico normal por la red unicast los routers multicast pueden manejarlos sin ningún problema. En estos casos se habla de "islas" multicast unidas a través de túneles unicast. En parte la red MBone está construida mediante túneles de este tipo usando hosts UNÍX como routers multicast (existe software de routing multicast de libre distribución disponible para los principales sistemas operativos UNIX, como Linux, Solaris, Iris, etc.).

Un problema que se plantea con el tráfico multicast en MBone es la forma de limitar el ámbito o alcance de los paquetes. En principio un usuario puede estar interesado en realizar una emisión multicast en un ámbito restringido (por ejemplo a España únicamente). En MBone esto se puede resolver de dos formas: la antigua, aun bastante utilizada el campo TTL de la cabecera IP para limitar el alcance del datagrama. Los routers multicast que se encuentran en la frontera del ámbito correspondiente (en nuestro ejemplo los routers internacionales) restarán varias unidades al TTL de forma que esos paquetes no puedan salir al exterior. La solución más reciente a este problema consiste en asignar el ámbito de acuerdo con el rango de direcciones multicast de que se trate, y elegir la dirección de acuerdo con este criterio. Para este fin se ha reservado el rango de 239.0.0.0 a 239.255.255.255 según se especifica en el RFC 2365.

## 5.9. IPV6

Aunque la adopción de medidas paliativas como CIDR ha dado un respiro momentáneo en el problema del agotamiento del espacio de direcciones y tablas de routing, es evidente que hay que ampliar el campo dirección en la cabecera de los datagramas IP.

En un intento por resolver este problema el IETF empezó a trabajar ya en 1990 en una nueva versión de IP. Aunque fuera el más importante, la escasez de direcciones no era el único problema que presentaba el protocolo IP, por lo que ya puesto a diseñar un nuevo protocolo el IETF decidió abordar también otras deficiencias detectadas. Los objetivos planteados fueron los siguientes:

- **Direcciones:** Establecer un espacio de direcciones que no se agote en el futuro previsible.
- **Eficiencia:** Reducir el tamaño de las tablas de routing. Simplificar el protocolo (la cabecera IP) para permitir a los routers procesar los paquetes más rápidamente.
- **Seguridad:** Ofrecer mecanismos que permitan incorporar fácilmente en el protocolo medidas de seguridad (privacidad y validación) mediante técnicas criptográficas.
- **Tipo de servicio:** Manejar mejor los diferentes tipos de servicio, en especial para poder ofrecer garantías de Calidad de Servicio y para permitir el uso de aplicaciones en tiempo real.
- **Multicasting:** Facilitar el uso de aplicaciones multicast.
- **Movilidad:** Permitir la movilidad de un host sin necesidad de cambiar su dirección.
- **Evolución:** Contemplar un mecanismo que permita extender el protocolo en el futuro.
- **Compatibilidad:** Permitir la coexistencia del protocolo nuevo con el viejo.

El IETF hizo una convocatoria pública (RFC 1550) para que se presentaran propuestas de como podría ser el nuevo protocolo. De las propuestas iniciales se fueron descartando las consideradas menos interesantes, y finalmente la propuesta finalmente adoptada fue un híbrido de dos de las presentadas.

Durante el período en que se celebraban en el seno del IETF las discusiones sobre las diferentes propuestas el nuevo protocolo se denominó IPng o IP next generation. Finalmente el nombre oficial elegido fue IP Versión 6 o abreviadamente IPv6 (el IP actualmente en uso es Versión 4, y la Versión 5 se había utilizado ya para un protocolo experimental denominado ST, STream protocol). El protocolo se especificó en el RFC 1883 publicado en diciembre de 1995; este RFC ha sido sustituido posteriormente por el RFC 2460 que introdujo pequeños cambios en la cabecera relativos al campo Differentiated Services, publicado en diciembre de 1998.

Uno de los aspectos más polémicos que rodearon la estandarización de IPv6 fue la decisión sobre el tamaño de las direcciones a utilizar. Se barajaron alternativas que iban desde ocho hasta 20 bytes. Probablemente la opción de 20 bytes era la más razonable, no porque hicieran falta direcciones tan grandes sino porque este era el formato utilizado en las direcciones OSI y existía ya un protocolo muy similar a IP que utilizaba este formato de direcciones, que era CLNP. De esta forma IPv6 podría haberse reducido simplemente a la adopción de CLNP, y de hecho esta opción llegó a plantearse seriamente en el seno del IAB. Pero cuando la noticia llegó a oídos del IETF se produjo un clamor en contra de CLNP, fundamentalmente por razones políticas: después de los años de enfrentamiento vividos entre los partidarios de los protocolos OSI y los de TCP/IP se consideraba inaceptable y políticamente incorrecto que el IAB adoptara CLNP como el IP del futuro, lo cual habría significado que después de todo había algo aprovechable en los protocolos OSI. Finalmente el IAB aceptó el requerimiento del IETF, desestimó la propuesta de adoptar CLNP y se creó un protocolo con direcciones de 16 bytes.

En el mercado existen ya varias implementaciones de IPv6 para hosts y para routers, aunque su uso hasta ahora se ha limitado a experiencias piloto.

IPv6 coincide con IPv4 en ofrecer un servicio de entrega de datagramas sin garantías, es decir "best effort". Se contemplan sin embargo algunas opciones que permiten ofrecer calidad de servicio.

IPv6 no es realmente compatible con IPv4 ya que utiliza un formato de cabecera diferente, pero lo es (con pequeñas modificaciones) con los demás protocolos de Internet. La implantación del nuevo protocolo se está realizando de forma gradual mediante la creación de "islas" IPv6 en las que se utiliza el nuevo protocolo; para la interconexión de esas islas a través del backbone IPv4 se utilizan túneles, de forma similar a lo que se hace con la red MBone. La red experimental así formada se conoce como 6Bone (IPv6 Backbone) y empezó a funcionar en 1996. En España ya hay una red 6Bone funcionando entre varias universidades españolas desde 1997. La Universidad de Valencia se incorporó a esta red en 1998.

Los protocolos de routing se han tenido que modificar para tener en cuenta las características propias y el nuevo formato de direcciones que utiliza IPv6; así se ha creado por ejemplo RIPv6 y OSPFv6.

Muy brevemente algunas de las principales virtudes de IPv6 son las siguientes:

- Las direcciones son de 16 bytes, lo cual da un espacio de direcciones increíblemente grande, suficiente con creces para todo futuro uso previsible.
- La cabecera se simplifica, pasando de 13 a 8 campos<sup>4</sup>. Se acelera así el proceso en los routers.
- Se ha mejorado el soporte de los campos opcionales en la cabecera.
- La seguridad (validación y privacidad) es ahora una parte fundamental del protocolo.
- Se dan más facilidades que antes para especificar el tipo de servicio.

### **5.9.1. Direcciones en IPv6**

Una dirección IPv6 está compuesta por 16 bytes. Los primeros bits identifican el tipo de dirección, de manera análoga a IPv4. Existen muchas clases de direcciones, pero no todas tienen asignado el mismo rango, y la mayoría están reservadas para usos futuros.

Se ha previsto un rango específico para direcciones IPv4. De esta forma cualquier dirección IPv4 puede incluirse en un datagrama IPv6.

---

<sup>4</sup> En IPv4 no se consideran campos los bits reservados ni las opciones.

Una parte del espacio de direcciones se reserva para distribución geográfica, de manera similar a como se hace actualmente en la Internet con CIDR.

Otra parte se ha reservado para repartir direcciones por proveedor. Se ha contemplado la posibilidad de que la Internet (o una parte de ella) evolucione hacia una red que interconecte las redes de los grandes proveedores a nivel mundial, siendo secundaria en este caso la ubicación geográfica. Se contempla para este caso una estructura de direcciones jerárquica con varios niveles.

No se ha previsto ninguna dirección específica para broadcast, ya que esto se considera un caso particular de multicast. Para las direcciones multicast se ha previsto también un rango específico, y en la propia dirección multicast se ha reservado un campo de 4 bits que permite especificar el ámbito o alcance que se pretende tenga la emisión. El ámbito puede valer entre 0 y 15; el valor 14 se utiliza para indicar todo el planeta, y el valor 15 podría eventualmente utilizarse para transmisiones multicast que abarquen el espacio interestelar. Esto es similar a lo que se hace en IPv4 cuando se limita el ámbito en base a la dirección (RFC 2365).

Además de envíos unicast, multicast y broadcast pueden hacerse envíos *anycast*, en los que el paquete se envía a uno cualquiera de los miembros de un grupo, sin importar a cual. Esto permite por ejemplo acceder a un servidor multihomed haciendo balance de carga entre varias interfaces, o por aquella que esté mas cerca del solicitante. También facilita configuraciones redundantes donde un determinado servicio puede ser facilitado por mas de un servidor.

Se ha previsto un rango de direcciones de significado puramente local, equivalentes a las actuales redes privadas (intranets), para casos en que por razones de seguridad se quiera estar completamente aislado del exterior.

La escritura de direcciones de 16 bytes usando el sistema tradicional resulta muy farragosa, por ejemplo:

128.0.0.0.0.0.0.1.35.69.103.137.171.205.239

Para evitarlo se ha diseñado una notación en la que las direcciones se escriben como 8 grupos de 4 dígitos hexadecimales separados por dos puntos; por ejemplo la dirección anterior se escribiría así:

8000:0000:0000:0000:0123:4567:89AB:CDEF

Dado que muchas direcciones contendrán gran cantidad de ceros se ofrece la posibilidad de utilizar una notación abreviada en la que los ceros a la izquierda pueden omitirse, y además si uno o más grupos tienen todos los dígitos a cero se pueden omitir poniendo en su lugar dobles puntos (::). Así por ejemplo la dirección anterior se escribiría:

8000::123:4567:89AB:CDEF

Para evitar ambigüedad la notación abreviada :: sólo puede utilizarse una vez en una dirección.

Por último, para facilitar la escritura de direcciones IPv4 se prevé también el uso de la notación decimal si se desea utilizando puntos, por ejemplo :

::147.156.11.11

Gracias a la mucho mayor longitud del campo dirección en IPv6 se pueden reservar los últimos bytes de la dirección para incluir una parte local globalmente única en la dirección, que normalmente es una dirección MAC IEEE<sup>5</sup>. Este “truco” (utilizado ya en redes OSI y ATM) permite la autoconfiguración de los sistemas: el equipo fija la parte host de su dirección a partir de la dirección contenida en su tarjeta

<sup>5</sup> A partir del RFC 2373 se ha previsto utilizar para IPv6 el nuevo formato de direcciones MAC de ocho bytes aprobado por el IEEE, denominadas direcciones EUI-64 (EUI = Extended Unique Identifier). Las direcciones EUI-64 utilizan el mismo prefijo de tres bytes que las direcciones MAC tradicionales para identificar al fabricante pero amplían a cinco bytes la parte que identifica al equipo. Se ha definido una forma estándar de convertir las direcciones de 48 bits en direcciones de 64 bits.

de red y escucha por el cable para que el router le informe de la parte de red, con lo que la conexión de nuevos equipos a una red puede hacerse verdaderamente “Plug-and-Play”. Una empresa que estuviera conectada a Internet a través de un proveedor podría cambiar de proveedor y cambiar todas sus direcciones sin más que cambiar en el router la parte de red de la dirección. Los ordenadores obtendrían el nuevo prefijo del router y le añadirían cada uno la parte host correspondiente. El uso de la dirección MAC como sufijo garantiza que la dirección sea única. La autoconfiguración también facilita grandemente la movilidad de equipos.

Con 16 bytes es posible crear  $2^{128}$  direcciones, equivalente a aproximadamente  $3 \times 10^{38}$ ; aunque el reparto produce mucho despilfarro es evidente que IPv6 no andará escaso de direcciones en mucho tiempo.

### 5.9.2. La cabecera de IPv6

Para tener una idea más exacta de las características de IPv6 vamos a repasar brevemente los campos de la cabecera, cuya estructura aparece en la tabla:

Campo	Longitud (bits)
Versión	4
DS (Differentiated Services)	8
Etiqueta de flujo	20
Longitud de carga útil	16
Siguiente cabecera	8
Límite de saltos	8
Dirección origen	128
Dirección destino	128

Formato de la cabecera de un datagrama IP Versión 6

El campo **versión** vale siempre 6. En principio este campo debería servir para distinguir las versiones de IP, de forma que todas pudieran identificarse como un mismo protocolo a nivel de enlace, por ejemplo utilizando el mismo valor de Ethertype. En la práctica muchas implementaciones de IPv4 no comprueban este campo sino que suponen que el datagrama es IPv4 cuando la cabecera del nivel de enlace especifica protocolo IP, por lo que a pesar de existir el campo versión es necesario asignar a IPv6 un valor propio en el nivel de enlace, como si se tratara de un protocolo diferente de IPv4.

El campo **DS (Differentiated Services)** se utiliza para especificar parámetros de Calidad de Servicio de acuerdo con la arquitectura Diffserv que veremos más adelante.

El campo **etiqueta de flujo** permite identificar los paquetes que pertenecen a una sesión concreta entre dos hosts, normalmente para solicitar un trato preferente o una determinada Calidad de Servicio. A este campo y al concepto de flujo nos referiremos más adelante cuando hablemos de Calidad de Servicio.

El campo **longitud de carga útil** indica el tamaño del paquete en bytes, excluidos los 40 bytes de la cabecera. El valor máximo es 65535. El paquete máximo es pues 65575.

El campo **siguiente cabecera** indica si esta cabecera está seguida por alguna de las cabeceras opcionales. Si no hay cabecerasopcionales este campo indica el protocolo del nivel de transporte al que pertenece este paquete, para lo cual utiliza los mismos códigos numéricos que en IPv4 (ver tabla 3.2).

El campo **límite de saltos** equivale al antiguo campo TTL, pero se ha decidido ponerle un nombre que refleje su uso real. Como en IPv4 el campo tiene 8 bits, por lo que el máximo número de saltos que puede especificarse es también 255. Algunos opinaban que este valor era demasiado bajo y que en algunos casos podría plantear problemas. Actualmente ya hay situaciones en la Internet donde se está próximo a los 64 saltos.

Por último, los campos **dirección de origen** y **dirección de destino** corresponden a las direcciones de 16 bytes que hemos descrito.

Los campos de la cabecera IPv4 que han desaparecido de la IPv6 son los siguientes:

**IHL (Internet Header Length):** este campo no existe pues la cabecera tiene ahora una longitud fija.

El campo **protocolo** no aparece pues su función la desempeña el campo **siguiente cabecera**.

El campo *checksum* se ha suprimido ya que no se consideró justificada la pérdida de rendimiento que supone el cálculo del checksum en cada salto ante la rara eventualidad de que se produzca un error en el nivel de red, tomando en cuenta que normalmente tanto el nivel de enlace como el de transporte realizan su propia comprobación de errores, por lo que realmente el checksum del datagrama IP no protege de errores de transmisión sino solamente de errores internos que se puedan producir en el router (por fallos en la memoria RAM por ejemplo). La supresión de este campo fue algo muy debatido durante la elaboración del nuevo estándar.

Todos los campos relativos a fragmentación han desaparecido de la cabecera porque en IPv6 la fragmentación se controla mediante cabeceras adicionales o extendidas; además en IPv6 todos los nodos han de aceptar paquetes de 1280 bytes como mínimo y sólo se permite la fragmentación en origen, es decir el emisor debe generar datagramas suficientemente pequeños para que puedan llegar a su destino sin fragmentaciones adicionales. Normalmente el emisor realizará el *Path MTU Discovery*, como ya era habitual en muchas implementaciones de IPv4.

### 5.9.3. Cabeceras extendidas

En IPv4 la cabecera puede contener algunos campos opcionales, principalmente para diagnóstico de problemas de routing, pero debido a su escasa longitud esos campos son poco utilizados. El sistema es demasiado rígido y poco eficiente, ya que los campos opcionales tienen que ser procesados en todos los routers del trayecto. En IPv6 se ha habilitado un mecanismo más flexible y eficiente; las cabecerasopcionales aparecen como cabeceras adicionales al final de la cabecera estándar; su presencia queda indicada por el campo *siguiente cabecera*, que en caso de que no haya opciones indicará el protocolo de transporte (normalmente TCP o UDP). De esta forma los campos opcionales en IPv6 pueden extenderse cuando se considere necesario. Además se prevé un mecanismo por el cual se puede indicar si las opciones deben ser procesadas por todos los routers del trayecto o solo por el último, lo cual permite una reducción significativa del trabajo a desarrollar por los routers intermedios cuando se trata de una opción que solo debe ser procesada por el nivel de red en el host de destino.

La cabecera *salto-a-salto* indica información que debe ser examinada por todos los routers por los que transite este datagrama. Hasta ahora solo se le ha definido a esta cabecera una opción, que permite especificar datagramas de longitud superior a 64 KB; estos datagramas (que pueden llegar a tener hasta 4 GB) se conocen como *jumbogramas*.

La cabecera *routing* realiza las funciones combinadas del strict source routing y loose source routing de IPv4; el máximo número de direcciones que pueden especificarse es de 24.

La cabecera *fragment* se utiliza cuando hay que fragmentar un datagrama; el mecanismo utilizado es muy similar al de IPv4 (campos Identificación, Desplazamiento del fragmento y MF), con la diferencia de que en IPv6 solo se permite la fragmentación en origen. De esta forma se simplifica notablemente la complejidad de proceso en los routers.

La cabecera *authentication* permite el uso de técnicas criptográficas para incorporar un mecanismo de firma digital por el cual el receptor del paquete puede estar seguro de la autenticidad del emisor.

La cabecera *encrypted security payload* permite el envío de información encriptada para que solo pueda ser leída por el destinatario. Evidentemente la encriptación afecta únicamente a los datos, no incluye la cabecera del datagrama puesto que ésta ha de ser leída e interpretada por cada router por el que pasa.

**Tema VI****Capa de Transporte****6.1. INTRODUCCIÓN**

El nivel de transporte se encarga de suministrar el servicio de transporte de bits a las aplicaciones. Éstas funcionan generalmente según el paradigma cliente-servidor, por el cual una aplicación (cliente) toma la iniciativa y solicita los servicios a la otra (servidor).

Como ya sabemos la comunicación “peer to peer” entre dos entidades del nivel de transporte ocurre en realidad gracias a los servicios ofrecidos por el nivel de red. Mientras que el nivel de red se ocupa de resolver los problemas propios de la topología de ésta (rutas y congestión fundamentalmente) el nivel de transporte sólo existe en las dos entidades extremas de la comunicación, por lo que también se le llama nivel host-host o extremo a extremo. El nivel de transporte no es consciente, ni debe serlo, de la manera como físicamente están interconectados los dos hosts, que puede ser por una LAN, una WAN o una combinación de múltiples redes de ambos tipos.

La unidad básica de intercambio de información a nivel de enlace se denomina *trama* (porque los datos van “rodeados” de información de control por delante y por detrás). En el nivel de red esta unidad básica se conoce como *paquete (Datagrama)*. No existe un término equivalente para la unidad de transferencia de información en el nivel de transporte; a falta de mejor alternativa utilizaremos para este fin el término OSI *TPDU* (Transport Protocol Data Unit); en la Internet se suele utilizar el término *mensaje* en el caso de UDP (servicio no orientado a conexión), y *segmento* en el de TCP (servicio orientado a conexión), pero esta nomenclatura no es compartida por otros protocolos de transporte.

Generalmente las aplicaciones requieren que el nivel de transporte les garantice la entrega de los datos al destinatario, sin errores, pérdidas ni datos duplicados; para que esto sea posible el nivel de transporte ofrecerá normalmente un servicio orientado a conexión, con retransmisiones en caso necesario. Este es el caso por ejemplo del protocolo TCP de Internet, utilizado en muchas aplicaciones como FTP (File Transfer Protocol, transferencia de ficheros), SMTP (Simple Mail Transfer Protocol, correo electrónico), HTTP (HyperText Transfer Protocol, usado en tráfico Web), etc.

En ocasiones las aplicaciones se conforman (o incluso prefieren) un servicio menos fiable en el que los mensajes se envían sin pedir confirmación, de forma independiente unos de otros. Este tipo de servicio se suministra normalmente con un protocolo no orientado a conexión. El protocolo UDP de Internet es un ejemplo de este tipo de servicio. Entre los casos en que se quiere un servicio de este tipo se encuentran por ejemplo las aplicaciones en tiempo real ya que en ese caso no se quiere incurrir en el retardo propio de un protocolo orientado a conexión.

Al igual que en Internet en OSI también hay dos protocolos de transporte, uno orientado a conexión y uno no orientado a conexión. La tabla muestra los más importantes protocolos de nivel de red y de transporte de Internet y sus correspondientes protocolos OSI.

Tipo de protocolo	Internet	OSI
Nivel de red	IP (Internet Protocol)	CLNP (ConnectionLess Network Protocol)
Routing interno	OSPF (Open Shortest Path First)	IS-IS ( Intermediate System to Intermediate System)
Routing externo	BGP (Border Gateway Protocol)	IDRP (InterDomain Routing Protocol)
Nivel de transporte, orientado a conexión	TCP (Transmission Control Protocol)	TP4 (Transport Protocol clase 4)
Nivel de transporte, no orientado a conexión	UDP (User Datagram Protocol)	TP0 (Transport Protocol clase 0)

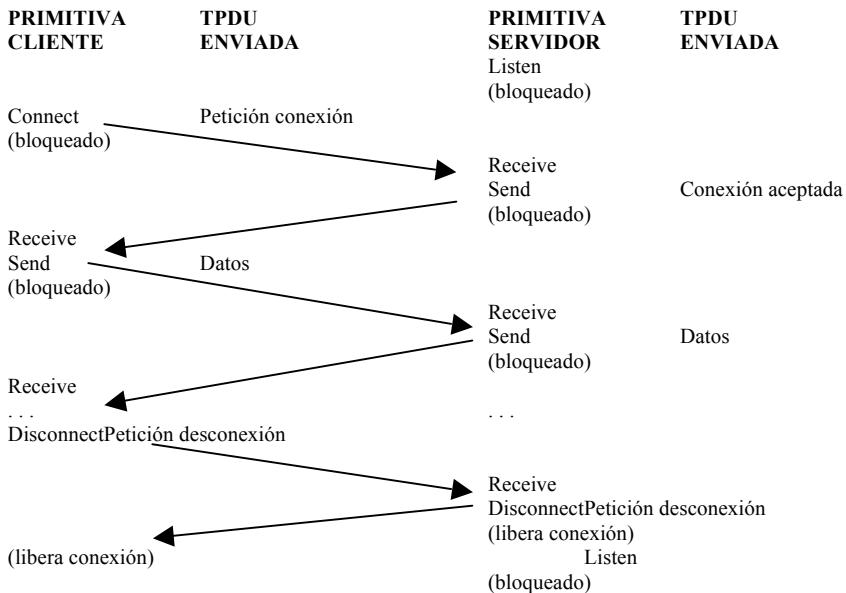
Correspondencia de protocolos Internet y OSI a nivel de red y de transporte

Normalmente las entidades del nivel de transporte se implementan como procesos en el sistema operativo del host, o bien procesos de usuario. El sistema operativo puede ser monousuario o multiusuario. En muchos casos el host tiene una sola instancia del nivel de red, y una sola del de transporte, pero muchas de los niveles superiores (aplicación o sesión); el nivel de transporte se encarga de multiplexar el tráfico recibido de las diversas entidades de nivel superior en una única conexión a través del nivel de red.

Los protocolos de transporte no orientados a conexión, como UDP, son protocolos muy sencillos, que existen únicamente para permitir la correcta conexión de la capa de aplicación y la de red; actúan como una capa de adaptación bastante primitiva. Por esto la mayoría de nuestra discusión se centrará en los protocolos orientados a conexión, a los que nos referiremos implícitamente la mayor parte del tiempo.

### 6.1.1. Primitivas del servicio de transporte

En un servicio de transporte básico orientado a conexión (CONS) la secuencia de primitivas podría ser algo como lo siguiente:



### 6.1.2. La interfaz sockets

La interfaz utilizada entre los diferentes niveles de un mismo sistema no forma parte de un protocolo de comunicaciones. Dos sistemas necesitan acordar las reglas que seguirá la comunicación entre ellos en cada uno de los niveles, pero la forma como se realiza la comunicación vertical, es decir, la que ocurre dentro de un sistema es asunto interno que no incumbe a los protocolos. Esta comunicación vertical se realiza normalmente mediante las denominadas APIs (Application Programming Interfaces).

Aunque no requerida por las comunicaciones, la estandarización de las APIs comporta unos beneficios evidentes por la posibilidad de aprovechar software entre sistemas diferentes. Esto es especialmente cierto en el nivel de transporte, ya que es aquí donde interaccionarán más programas diferentes, correspondientes a las diversas aplicaciones que se desarrollen. Muchas implementaciones de TCP/IP disponen de una API para programación aplicaciones denominada sockets (literalmente enchufes, aunque nunca se utiliza esta denominación). Los sockets se introdujeron con el sistema operativo UNIX BSD (Berkeley Software Distribution) en 1982. La interfaz sockets es multiprotocolo, soporta TCP, UDP y otros protocolos.

Aun cuando no forman parte de ningún estándar oficial ni están recogidos en ningún RFC, los sockets son la API más extendida en programación de aplicaciones TCP/IP y forman un estándar “de facto”. Existen implementaciones para muchos sistemas operativos no UNIX, e incluso en casos en que el sistema operativo no los incorpora suele haber una librería de rutinas sockets que permite adaptar programas con relativa facilidad. Por ejemplo, la interfaz WinSock permite adaptar aplicaciones MS Windows sobre diversas implementaciones de TCP/IP. Al no ser un estándar puede haber pequeñas diferencias entre implementaciones, por lo que es conveniente disponer siempre de la documentación correspondiente al software que se utiliza.

La filosofía básica de los sockets deriva directamente del sistema de entrada/salida de UNIX, con ampliaciones que permiten por ejemplo a un proceso servidor ponerse “a la escucha”. Algunas de las rutinas generan y envían las TPDUs a partir de sus argumentos; éstas (las TPDUs) sí forman parte del protocolo, por lo que deben de ser conformes con el estándar correspondiente.

## 6.2. ELEMENTOS DE PROTOCOLOS DE TRANSPORTE

Al ocuparse de la comunicación extremo a extremo o punto a punto, el nivel de transporte se parece en algunos aspectos al nivel de enlace. Así por ejemplo, entre los asuntos de los que normalmente habrá de ocuparse se encuentran el control de errores (incluyendo mensajes perdidos o duplicados) y el control de flujo. Aunque las técnicas que se aplican son parecidas, existen importantes diferencias entre ambos motivadas por el hecho de que en el nivel de enlace hay sólo un hilo físico (o su equivalente) entre las dos entidades comunicantes, mientras que en el nivel de transporte hay toda una red. Las mayores diferencias entre el nivel de transporte y el de enlace son las siguientes:

- El retardo que se observa en el nivel de transporte es normalmente mucho mayor y sobre todo más variable (mayor jitter) que en el de enlace.
- En el nivel de enlace el medio físico entre las dos entidades tiene una capacidad de almacenamiento de información normalmente muy reducida y siempre la misma; en el de transporte los routers intermedios pueden tener una capacidad considerable y esta puede variar con el estado de la red.
- En el nivel de enlace se asegura que las tramas llegarán al receptor en el mismo orden que han salido del emisor (salvo que se pierdan, en cuyo caso no llegarán); en el nivel de transporte esto es cierto solo cuando se utiliza un servicio orientado a conexión en el nivel de red; si se utiliza un servicio no orientado a conexión el receptor podría recibir los datos en orden distinto al de emisión.
- En el nivel de enlace las dos entidades se “ven” directamente (suponiendo una comunicación dúplex); a veces incluso de forma permanente, por ejemplo en una comunicación síncrona tipo HDLC están continuamente emitiendo la secuencia 01111110; esto permite que el emisor sepa en todo momento si el receptor está operativo, y el receptor sabe que los datos recibidos corresponden todos a una misma sesión del emisor. En el nivel de transporte la comunicación es indirecta; el emisor podría enviar datos, quedar fuera de servicio y más tarde entrar en funcionamiento otra vez; si no se adoptan las medidas oportunas el receptor podría recibir todos esos datos sin siquiera percatarse de que corresponden a dos sesiones distintas del emisor (incluso podrían pertenecer a dos usuarios distintos).

Recordemos que en el modelo OSI cada nivel presta sus servicios al nivel superior a través del SAP (Service Access Point), cada uno de los cuales es identificado por una dirección. Por ejemplo en Internet la dirección SAP por la que el nivel de red accede al servicio es la dirección IP del host. La dirección SAP por la que el nivel de transporte accede al servicio de red está formada por el campo *protocolo* del datagrama IP (6 para TCP y 17 para UDP, por ejemplo). A su vez el nivel de transporte ofrece sus servicios al nivel de aplicación a través de unos SAPs específicos, que en el caso de Internet son los denominados *ports* o *puertos*. Estos puertos se denominan también TSAPs (Transport Service Access Point).

Para que un proceso cliente de un host pueda comunicar con un proceso servidor en otro host haciendo uso de los servicios de su nivel de transporte es preciso que conozca el TSAP correspondiente en el host de destino. Normalmente esta información forma parte del estándar del protocolo, por lo que es universalmente conocido y cualquier cliente que lo deseé sabe que TSAP debe utilizar para acceder a dicho servidor. En cambio el TSAP del cliente no necesita ser conocido por otros usuarios y puede ser diferente para cada conexión.

### 6.2.1. Establecimiento de una conexión

En principio para establecer una conexión el cliente emite una TPDU de petición de conexión, y el servidor responde con una TPDU de aceptación; a partir de ese momento puede empezar el intercambio de datos. Sin embargo cuando analizamos el proceso de conexión con mayor detalle encontramos diversos problemas que pueden presentarse y que hay que prever.

Recordemos que en el nivel de transporte puede haber una gran fluctuación (a veces del orden de segundos) en el tiempo que tardan en llegar las TPDUs a su destino; las TPDUs pueden perderse o llegar duplicadas, ya que si el emisor no recibe confirmación reenviará la misma TPDU pasado el timeout. Imaginemos que el cliente intercambia una serie de TPDUs con el servidor, y cuando ya ha terminado la transacción cierra la sesión; segundos mas tarde de algún rincón de la red aparecen la misma secuencia de

TPDUs del cliente duplicadas que llegan al servidor de nuevo; éste realizaría la misma transacción otra vez, con efectos posiblemente desastrosos<sup>1</sup>.

Para evitar este tipo de problemas se utiliza para establecer la conexión el mecanismo conocido como *saludo a tres vías* (three-way handshake). La idea es que el servidor sólo aceptará la conexión después de haber pedido al cliente confirmación de que desea realizarla. En principio esto por sí solo no resuelve nuestro problema, ya que cabría pensar que después de la TPDU de petición inicial duplicada la red le entregue al servidor la TPDU de confirmación, también retrasada.

La solución a este problema es la siguiente: tanto el cliente como el servidor utilizan un protocolo de ventana deslizante para el envío de las TPDUs, para lo cual emplean un número de secuencia; a diferencia del número de secuencia que vimos en el nivel de enlace, el del nivel de transporte emplea rangos muy amplios (por ejemplo en TCP el número de secuencia se almacena en un campo de 32 bits, con lo que es un número módulo  $2^{32}$ ). Tanto el cliente como el servidor eligen de forma aleatoria o pseudo aleatoria el valor inicial del número de secuencia que van a utilizar, cada uno por separado para cada sentido de la comunicación. El cliente informa al servidor en su primera TPDU del número de secuencia elegido; por su parte el servidor le responde en otra TPDU con el número de secuencia elegido por él, incluyendo en ésta un ACK piggybacked de la TPDU recibida. De esta forma si el servidor recibe una TPDU de petición de conexión vieja responderá con una TPDU al cliente en la que pondrá en el campo ACK el número de secuencia recibido; cuando la respuesta llegue al cliente éste verá que ese número no corresponde con ninguna conexión que él tuviera pendiente de confirmación, por lo que rechazará la conexión; el servidor por su parte esperará recibir en el campo ACK de la siguiente TPDU un valor que corresponda con el que él ha enviado en la anterior.

La técnica de los números de secuencia aleatorios evita también el riesgo de que un proceso cliente que cae por algún motivo (por ejemplo por un fallo de corriente) utilice la misma conexión cuando reaparece más tarde, ya que normalmente el nuevo proceso intentará utilizar un número de secuencia diferente. Esta es una medida de seguridad ya que el nuevo proceso cliente podría pertenecer a otro usuario; supongamos por ejemplo que al inicio de la conexión se realiza una identificación con clave usuario/password ante el servidor, en tal caso el nuevo cliente podría acceder a todos los recursos del usuario anterior sin identificarse.

Generalmente se establece una vida máxima para las TPDUs en la red; de esta forma se reduce el riesgo de recibir duplicados retrasados. Cuando un sistema cae y vuelve a arrancar se recomienda esperar al menos el tiempo de vida de las TPDUs antes de activar el nivel de transporte; de esta manera es imposible que una TPDU de la sesión anterior pueda aparecer por alguna parte cuando se inicia la sesión nueva. En Internet por ejemplo el tiempo de vida máximo recomendado de las TPDUs es de 2 minutos, y se controla mediante el campo TTL en el datagrama IP.

Una vez establecidos los números de secuencia es posible utilizar para el intercambio de TPDUs cualquier protocolo de ventana deslizante. A diferencia del nivel de enlace, donde el protocolo se basa en numerar tramas, en el nivel de transporte se suelen numerar bytes, ya que el tamaño de las TPDUs puede ser muy variable. Para las retransmisiones se puede utilizar tanto *retroceso n* como *repetición selectiva*.

### 6.2.2. Terminación de una conexión

Una conexión puede terminarse de forma simétrica o asimétrica. La terminación asimétrica es unilateral, es decir uno de los dos hosts decide terminar y termina la conexión en ambos sentidos. En la terminación simétrica cada host corta la conexión únicamente en el sentido en el que emite datos; podemos considerar la terminación simétrica como dos circuitos simplex donde cada uno es controlado por el emisor.

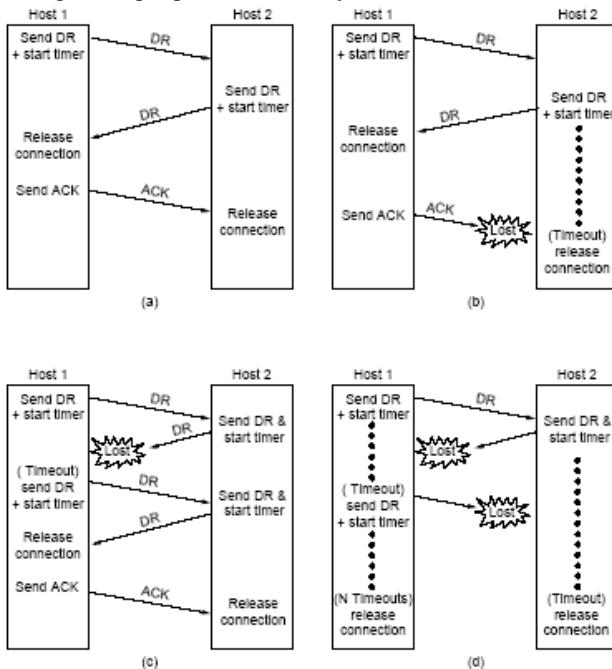
La terminación asimétrica se considera anormal y puede provocar la pérdida de información, ya que cuando un host ha enviado la TPDU de desconexión ya no acepta más datos; entretanto el otro host podría haber enviado una TPDU de datos que no será aceptada.

En la terminación simétrica (la más normal) el host 1 “invita” al host 2 a desconectar mediante una TPDU DISCONNECT REQUEST; el host 2 responde con otra DISCONNECT REQUEST, a la cual el host 1 responde con una TPDU ACK y cierra la conexión; el host 2 cerrará la conexión al recibir el ACK. Por este mecanismo se asegura que no se pierden TPDUs “en ruta” ya que ambos hosts tienen aviso previo de

<sup>1</sup> Si por ejemplo la transacción consiste en la transferencia de dinero entre cuentas bancarias se realizarían dos transferencias en vez de una.

la desconexión y dan su conformidad explícitamente. Este mecanismo supone el intercambio de tres mensajes de forma análoga al proceso de conexión, por lo que también se denomina saludo a tres vías (aunque quizás debería llamarse “despedida a tres vías”); no existe forma fiable de terminar la conexión en menos mensajes sin correr el riesgo de perder datos.

Si se pierde alguna de las TPDUs de desconexión el mecanismo del saludo a tres vías falla pues los hosts se quedan esperando eternamente la respuesta. Para evitar esto se utiliza un mecanismo de timeouts que resuelve el problema reenviando la TPDU perdida si se trata de un DISCONNECT REQUEST, o cerrando la conexión por timeout cuando lo que se ha perdido es el ACK. En Tanenbaum aparece una relación de los casos “patológicos” que pueden ocurrir y como se resuelven:



4 escenarios para liberar una conexión. (a) Caso Normal de saludo de tres-vías. (b) Final ACK perdido.  
(c) Respuesta perdida. (d) Respuesta perdida y subsecuentes DRs perdidas.

Existen muchas circunstancias que pueden provocar que una conexión se quede medio abierta, es decir abierta sólo por un lado. Por ejemplo, un host puede quedar fuera de servicio sin previo aviso y el otro, que tenía una conexión abierta con él, quedar a la espera sin saber que ha ocurrido. Para resolver estas situaciones se prevé normalmente un tiempo máximo durante el cual una conexión puede estar abierta sin tráfico; pasado ese tiempo los hosts se envían mensajes de prueba (denominados keep-alive en TCP) para comprobar que el otro lado aún responde. Los valores de timeout para el envío de mensajes keep-alive son increíblemente grandes (la documentación de TCP sugiere 2 horas como valor por defecto). Un valor muy pequeño podría provocar que un fallo momentáneo en la red cerrara conexiones a nivel de transporte, perdiendo así la principal ventaja de las redes de datagramas.

Analicemos ahora que ocurre si dos hosts tienen establecida una conexión entre ellos y falla la red que los une. En el caso de utilizar un servicio orientado a conexión (X.25, ATM) generalmente la sesión termina de manera abrupta, pues la vía de comunicación (el circuito virtual) ha desaparecido y para restaurar la comunicación habrá que restablecer el circuito, presumiblemente por un camino físico diferente. En el caso de utilizar un servicio de red no orientado a conexión (IP, OSI CLNP) la red reencaminará los datagramas por una ruta alternativa (suponiendo que exista) por lo que lo único que el nivel de transporte detectará es la pérdida de unas pocas TPDUs, pero la conexión no se cortará.

### 6.2.3. Control de flujo y de buffers

El control de flujo en el nivel de transporte es fundamental, ya que la velocidad con que los datos llegan al receptor puede ser muy variable al intervenir multitud de factores.

Como ya hemos dicho se suelen utilizar protocolos de ventana deslizante. Mientras que en el nivel de enlace se asignaba de manera estática un espacio de buffers a cada conexión, en el nivel de transporte esta estrategia no es adecuada, pues el número de conexiones simultáneas puede variar muchísimo al no haber

una interfaz física asociada a cada conexión. Por este motivo la asignación de espacio para buffers en el nivel de transporte tiene dos características singulares que le diferencian del nivel de enlace. En primer lugar el espacio de buffers es común y compartido por todas las conexiones, entrantes y salientes. En segundo lugar el reparto del espacio entre las conexiones activas se hace de forma dinámica de acuerdo con las necesidades; una conexión con poco tráfico recibirá menos asignación que una con mucho tráfico. En todo momento cada conexión tiene asignado un espacio para emisión y uno para recepción; el espacio de emisión está ocupado con TPDUs pendientes de ser enviadas o de confirmación; el espacio de recepción tiene una parte ocupada con TPDUs recibidas pendientes de ser aceptadas por el nivel de aplicación, y otra libre reservada para TPDUs que puedan llegar del otro host.

Otra diferencia respecto al nivel de enlace estriba en que, mientras que el tamaño de las tramas suele ser constante para una conexión física dada, el tamaño de las TPDUs puede ser muy variable. Para optimizar la utilización del espacio se asignan segmentos de buffer de longitud variable. Para una máxima flexibilidad en este sentido tanto los números de secuencia como los tamaños de ventana cuentan generalmente bytes, no TPDUs.

La parte de buffer que el receptor tiene reservada para TPDUs que puedan llegarle es anunciada al emisor regularmente, para que éste sepa qué cantidad de datos está dispuesto a aceptar el receptor. Este espacio puede fluctuar mucho con el tiempo en función de la actividad que tenga esa y el resto de conexiones que mantenga el host. Con este modo de funcionamiento el receptor realmente controla la situación, ya que si indica una ventana cero el emisor tendrá que esperar y no enviarle datos mientras el receptor no le anuncie una ventana mayor.

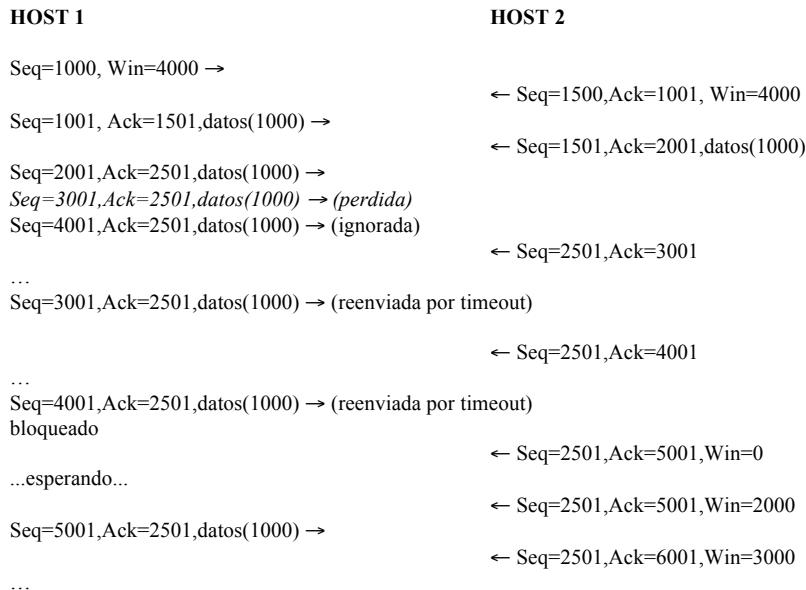
Veamos un ejemplo sencillo de cómo funcionaría una sesión TCP:

HOST 1	HOST 2
Seq=1000,Win=4000 →	← Seq=1500,Ack=1001,Win=4000
Seq=1001,Ack=1501,datos(1000) →	← Seq=1501,Ack=2001,datos(1000)
Seq=2001,Ack=2501,datos(1000) →	
Seq=3001,Ack=2501,datos(1000) →	
Seq=4001,Ack=2501,datos(1000) →	
bloqueado	← Seq=2501,Ack=5001,Win=0
...esperando...	← Seq=2501,Ack=5001,Win=2000
Seq=5001,Ack=2501,datos(1000) →	← Seq=2501,Ack=6001,Win=3000
...	

Supongamos ahora que en el ejemplo anterior se hubiera perdido la cuarta TPDU enviada de host1 a host2 (la que aparece en cursiva); en ese caso el host-2 no habría enviado el ACK 5001, y el host-1, al agotar el timeout correspondiente a esa TPDU la habría reenviado; funcionando con repetición selectiva la secuencia sería la siguiente:

HOST 1	HOST 2
Seq=1000,Win=4000 →	← Seq=1500,Ack=1001,Win=4000
Seq=1001,Ack=1501,datos(1000) →	← Seq=1501,Ack=2001,datos(1000)
Seq=2001,Ack=2501,datos(1000) →	
Seq=3001,Ack=2501,datos(1000) → (perdida)	
Seq=4001,Ack=2501,datos(1000) →	← Seq=2501,Ack=3001
...	
Seq=3001,Ack=2501,datos(1000) → (reenviada por timeout)	
bloqueado	← Seq=2501,Ack=5001,Win=0
...esperando...	← Seq=2501,Ack=5001,Win=2000
Seq=5001,Ack=2501,datos(1000) →	← Seq=2501,Ack=6001,Win=3000
...	

En caso de funcionar con retroceso n las cosas habrían sido ligeramente diferentes:



En redes no orientadas a conexión los datagramas (y por tanto las TPDUs) pueden llegar desordenados, por lo que el nivel de transporte debe estar preparado para recibir números de secuencia desordenados (siempre y cuando se encuentren dentro del rango correspondiente a la ventana vigente en ese momento).

#### 6.2.4. Multiplexación

En las redes públicas de commutación de paquetes (X.25, frame relay y ATM), que son orientadas a conexión, el usuario paga por cada circuito virtual, lo cual estimula a utilizar el mínimo número de circuitos posible. Generalmente es el nivel de transporte el encargado en estos casos de multiplexar las diferentes conexiones solicitadas por el nivel de aplicación en una única conexión a nivel de red; dicho en terminología OSI el nivel de transporte presenta diferentes TSAPs sobre un único NSAP. Esto se conoce como multiplexación hacia arriba, ya que visto en el modelo de capas supone que varias direcciones del nivel de transporte confluyan en una única dirección del nivel de red.

También en redes no orientadas a conexión (IP o ISO CLNP) el nivel de transporte suele ocuparse de multiplexar el tráfico de las diferentes aplicaciones y usuarios (cada aplicación puede estar siendo utilizada por varios usuarios) en una única dirección a nivel de red.

#### 6.2.5. Recuperación de caídas

Ya hemos visto el tipo de medidas preventivas que se adoptan para evitar que cuando una instancia del nivel de transporte en un host cae y se levanta mas tarde no sea posible recuperar la conexión previamente establecida, y haya que crear una nueva. Esto es una medida de seguridad fundamental para evitar inconsistencias en la información y accesos no autorizados.

Otro problema importante es que ocurre cuando cae todo un host, lo cual provoca la caída simultánea del nivel de transporte y el nivel de aplicación. Supongamos por ejemplo que un cliente está realizando una serie de actualizaciones en una base de datos, cada una de ellas contenida en una TPDU; a cada transacción el servidor responde con un ACK indicando que ha efectuado la operación correspondiente. En un determinado momento el servidor cae, rearmando a continuación; podemos concluir que si el cliente ha recibido el ACK es que la actualización se ha efectuado, y si no no, pero como la actualización y el envío del ACK son sucesos consecutivos y no simultáneos siempre ocurrirá uno primero y el otro después; cualquiera que sea el orden elegido siempre podrá ocurrir que el host caiga entre ambos eventos, con lo que tendremos o bien una actualización efectuada y no notificada, o una notificación enviada al cliente de una actualización no realizada. Este tipo de problemas solo puede resolverse a nivel de aplicación mediante una posterior verificación de lo que realmente ha ocurrido.

### 6.3. LOS PROTOCOLOS DE TRANSPORTE DE LA INTERNET: TCP Y UDP

Como ya hemos comentado existen dos protocolos de transporte en la Internet: TCP es fiable, orientado a conexión con control de flujo, y UDP es no fiable (sin confirmación) no orientado a conexión y sin control de flujo. La TPDU de TCP se denomina *segmento*, y la de UDP *mensaje* o también *datagrama UDP*.

TCP prevé una comunicación full dúplex punto a punto entre dos hosts, no hay soporte para tráfico multicast. En UDP la comunicación es simplex (aunque obviamente un datagrama UDP puede ser respondido por el receptor con otro); en UDP es posible el tráfico multicast o broadcast. El protocolo TCP es mucho más complejo que UDP.

#### 6.3.1. TCP (Transport Control Protocol)

Recordemos que el nivel de transporte debe ofrecer algún mecanismo que permita distinguir a qué aplicación van dirigidos los datos, lo que hemos denominado los TSAPs. En TCP los TSAPs se denominan *ports* o *puertos*.

En sentido estricto una conexión entre dos entidades usuarias del nivel de transporte queda identificada por los TSAPs en los que conectan cada una (podemos pensar en el TSAP como el conector telefónico, diríamos entonces que una conversación telefónica queda perfectamente especificada por los números de teléfono de las dos cajas donde están enchufados los aparatos con los que se está hablando). Recordaremos que un TSAP en Internet está especificado por:

- Dirección donde “conecta” el nivel de red: dirección IP de los dos hosts
- Dirección donde conecta el nivel de transporte (campo protocolo del datagrama IP): normalmente TCP ya que UDP al ser no orientado a conexión no puede establecer conexiones.
- Dirección donde conecta el nivel de aplicación: esto es el puerto.

Dado que la conexión en el nivel de transporte siempre se suele realizar con el protocolo TCP este dato es innecesario y se suele omitir. Sin embargo en un mismo host un número de port puede ser utilizado simultáneamente por una aplicación para UDP y por otra para TCP; esto no plantea ningún conflicto ya que son TSAPs diferentes.

Así pues, una conexión de dos entidades usuarias del nivel de transporte se especifica por la combinación:

**Dirección IP host 1 + port host 1 + dirección IP host 2 + port host 2**

El port es un número entero entre 0 y 65535. Por convenio los números 0 a 1023 están reservados para el uso de servicios estándar, por lo que se les denomina *puertos bien conocidos* o *well-known ports*. Cualquier número por encima de 1023 está disponible para ser utilizado libremente por los usuarios. Los valores vigentes de los puertos bien conocidos se pueden consultar por ejemplo en el web de la IANA (Internet Assigned Number Authority) [www.iana.org/numbers.html](http://www.iana.org/numbers.html). En la tabla se recogen algunos de los más habituales.

Puerto	Aplicación	Descripción
9	Discard	Descarta todos los datos recibidos (para pruebas)
19	Chargen	Intercambia cadenas de caracteres (para pruebas)
20	FTP-Data	Transferencia de datos FTP
21	FTP	Diálogo en transferencia de ficheros
23	TELNET	Logon remoto
25	SMTP	Correo electrónico
110	POP3	Servidor de correo
119	NNTP	News

Algunos ejemplos de puertos “bien conocidos” de TCP

Para comprender la relación entre los puertos y las conexiones en TCP veamos un ejemplo concreto: supongamos que cinco usuarios desde un host de dirección IP 134.123.1.2 inicien una sesión de login remoto (Telnet) hacia el host de dirección 221.198.34.21; cada uno de ellos ejecutará en su host un programa telnet cliente que abrirá una conexión con el servidor Telnet (puerto 23) en el otro; las conexiones establecidas podrían ser por ejemplo:

Usuario 1: 134.123.1.2.1024 con 221.198.34.21.23  
 Usuario 2: 134.123.1.2.1025 con 221.198.34.21.23  
 Usuario 3: 134.123.1.2.1026 con 221.198.34.21.23  
 Usuario 4: 134.123.1.2.1030 con 221.198.34.21.23  
 Usuario 5: 134.123.1.2.1031 con 221.198.34.21.23

Aquí hemos empleado la notación “dirección IP. Puerto” para identificar el socket<sup>2</sup>; cada conexión queda identificada de forma no ambigua por los dos sockets que conecta. Obsérvese que la asignación de puertos para los clientes se hace por simple orden de llegada a partir del primer número de puerto no reservado. En el servidor todas las conexiones utilizan el puerto 23 (pues todas acceden al mismo proceso, el servidor telnet); en cambio en el cliente cada usuario es un proceso diferente y utiliza un puerto distinto.

Complicando un poco más el ejemplo anterior podríamos imaginar que el host cliente estuviera “multihomed”, es decir que tuviera dos interfaces físicas (por ejemplo dos tarjetas LAN) y por tanto tuviera dos direcciones IP; supongamos que los usuarios utilizan ambas interfaces alternativamente, en ese caso las conexiones podrían ser:

Usuario 1: 134.123.1.2.1024 con 221.198.34.21.23  
 Usuario 2: 134.123.1.3.1024 con 221.198.34.21.23  
 Usuario 3: 134.123.1.2.1025 con 221.198.34.21.23  
 Usuario 4: 134.123.1.3.1025 con 221.198.34.21.23  
 Usuario 5: 134.123.1.2.1030 con 221.198.34.21.23

Por otro lado el host cliente podría simultáneamente a las sesiones telnet enviar datagramas UDP al servidor. Aunque en este caso no se establece una conexión (pues se trata de un servicio CLNS) hay un puerto de origen y uno de destino; podría haber datagramas que tuvieran como puerto de origen el 1024 en el host 134.123.1.2 y como destino el 23 en 221.198.34.21; esto no causaría ninguna ambigüedad ya que el campo protocolo de la cabecera IP permitiría distinguir ambos tipos de paquetes entregando cada uno al servicio correspondiente del nivel de transporte en el host de destino.

### 6.3.2. La cabecera de segmento TCP

La cabecera de un segmento TCP tiene la estructura que se muestra en la tabla.

Campo	Longitud (bits)
Puerto origen	16
Puerto destino	16
Número de secuencia	32
Número de ACK	32
Longitud de cabecera TCP	4
Reservado	4
CWR (Congestion Window Reduced)	1
ECE (ECN Echo)	1
URG (Urgent)	1
ACK (Acknowledgement)	1
PSH (Push)	1
RST (Reset)	1
SYN (Synchronize)	1
FIN (Finish)	1
Tamaño de ventana	16
Checksum	16
Puntero de datos urgentes	16
Opciones	0, 32, 64, ...
Datos	0-523960 (65495 bytes)

Estructura de la cabecera de un segmento TCP

*Puerto origen* y *punto destino*: identifican los puertos que se van a utilizar en cada host para comunicar con las aplicaciones que intercambian datos.

*Número de secuencia*: indica el número de secuencia que corresponde en la conexión al primer byte que se envía en el campo datos de ese segmento.

<sup>2</sup> La denominación socket empleada para la combinación IP.puerto es la misma que la de las APIs utilizadas en UNIX para acceder a los servicios TCP.

*Número de ACK*: indica el número de secuencia del primer byte del próximo segmento que se espera recibir del otro lado.

*Longitud de cabecera TCP*: especifica la longitud en palabras de 32 bits, excluido el campo datos (el campo opciones hace que dicha longitud pueda variar).

A continuación hay 4 bits no utilizados, seguidos por ocho flags indicadores de un bit de longitud cada uno:

- CWR: Congestion Window Reduced. Tiene que ver con el control de congestión de IP que no describiremos aquí
- ECE: ECN Echo (ECN=Explicit Congestion Notification). Tiene que ver con el control de congestión de IP que no describiremos aquí.
- URG (urgent): sirve para indicar que el segmento contiene datos urgentes; en ese caso el campo puntero de datos urgentes contiene la dirección donde terminan éstos.
- ACK (acknowledgement): indica que en este segmento el campo *Número de ACK* tiene el significado habitual (número del próximo byte que se espera recibir), de lo contrario carece de significado. En la práctica el bit ACK esta a 1 siempre, excepto en el primer segmento enviado por el host que inicia la conexión.
- PSH (push): indica que el segmento contiene datos PUSHed. Esto significa que deben ser enviados rápidamente a la aplicación correspondiente, sin esperar a acumular varios segmentos.
- RST (reset): se usa para indicar que se debe abortar una conexión porque se ha detectado un error de cualquier tipo; por ejemplo una terminación unilateral de una conexión o que se ha recibido un segmento con un valor inadecuado del *número de secuencia* o *número de ACK*, posiblemente producido por un duplicado retrasado de un intento de conexión.
- SYN (synchronize): este bit indica que se está estableciendo la conexión y está puesto sólo en el primer mensaje enviado por cada uno de los dos hosts en el inicio de la conexión.
- FIN (finish): indica que no se tienen más datos que enviar y que se quiere cerrar la conexión. Para que una conexión se cierre de manera normal cada host ha de enviar un segmento con el bit FIN puesto.

*Tamaño de ventana*: indica la cantidad de bytes que se está dispuesto a aceptar del otro lado en cada momento. Se supone que se garantiza una cantidad suficiente de espacio en buffers. Mediante este parámetro el receptor establece un control de flujo sobre el caudal de datos que puede enviar el emisor.

*Checksum*: sirve para detectar errores en el segmento recibido; estos podrían ser debidos a errores de transmisión no detectados, a fallos en los equipos (por ejemplo en los routers) o a problemas en el software (por ejemplo reensamblado incorrecto de fragmentos). Recordemos que el datagrama IP contenía un checksum, pero aquel solo comprendía la información de cabecera y además ese checksum desaparece en IPv6. El algoritmo utilizado en TCP es el mismo que el de IP: sumar todos los campos de 16 bits usando aritmética de complemento a 1 y calcular el complemento a 1 del resultado, pero en este caso el checksum se hace sobre todo el segmento, incluidos los datos, no sólo sobre la información de cabecera. Para el cálculo del checksum se antepone al segmento una pseudo cabecera que incluye la dirección IP de origen y destino, el protocolo de transporte utilizado (TCP en este caso) y la longitud del segmento. La pseudo cabecera (así denominada porque sólo se utiliza en el cálculo, pero no forma parte del segmento) permite a TCP comprobar que no ha habido errores en la transmisión a nivel IP (detectar por ejemplo cuando un segmento ha sido entregado al host equivocado).

*Puntero de datos urgentes*: indica el final de éstos, ya que el segmento podría contener datos no urgentes. TCP no marca el principio de los datos urgentes, es responsabilidad de la aplicación averiguarlo.

El campo opciones habilita un mecanismo por el cual es posible incluir extensiones al protocolo. Entre las más interesantes se encuentran las siguientes:

- Tamaño máximo de segmento
- Uso de repetición selectiva (en vez de retroceso n)
- Uso de NAK (acuse de recibo negativo en caso de no recepción de un segmento)
- Uso de ventana mayor de 64 Kbytes mediante el empleo de un factor de escala

### 6.3.3. Tamaño de segmento y fragmentación

TCP divide (o agrupa) los mensajes recibidos del nivel de aplicación en TPDUs denominadas segmentos; en el momento de establecer la conexión cada host informa al otro del máximo tamaño de segmento que está dispuesto a aceptar; este tamaño deberá ser como mínimo de 536 bytes, correspondiente normalmente a un datagrama IP de 576 bytes menos 20 bytes de cabecera IP y 20 de cabecera TCP (la longitud de segmento se refiere a la parte de datos de TCP).

Un segmento TCP siempre se transporta en un datagrama IP. Cuando la red ha de fragmentar en algún punto un datagrama IP el datagrama *sigue fragmentado el resto del viaje hasta el host de destino*; una vez allí *el nivel de red* se ocupa de juntar todos los fragmentos en su buffer y reconstruir el datagrama original, del cual extrae entonces el segmento original y lo pasa a TCP; si uno de los fragmentos se pierde el nivel de red del receptor será incapaz de reconstruir el datagrama original, y por tanto descartará, una vez expirado el TTL, todos los fragmentos recibidos y no pasará ninguna parte de ese segmento al nivel de transporte; TCP agotará el timer, detectará el segmento faltante y pedirá retransmisión al emisor. Por tanto *cuando un fragmento de un datagrama se pierde todos los fragmentos se retransmiten nuevamente*; el proceso se repite hasta que todos los fragmentos consiguen llegar correctamente al receptor. También puede suceder que la fragmentación se produzca ya en el host emisor del segmento, en cuyo caso realizará fragmentado todo el trayecto.

#### 6.3.4. Flujo de datos en TCP

Los segmentos, al viajar en datagramas IP, pueden perderse, llegar desordenados o duplicados. Es la responsabilidad de TCP resolver todos estos problemas y generar un flujo de bits fiable para el nivel de aplicación. Cada segmento es acomodado el solo en un datagrama, si bien luego puede tener que ser fragmentado para su envío, como hemos visto en el apartado anterior. Nunca se combinan en un segmento datos pertenecientes a dos conexiones TCP diferentes.

Las aplicaciones que comunican haciendo uso de los servicios que ofrece TCP no tienen conciencia de la existencia de segmentos o datagramas. Para ellas la comunicación se realiza como un flujo continuo de bits (stream), como si hubiera un hilo físico que las comunicara. Si desean que la información sea transmitida a la aplicación receptora en mensajes discretos deberán incluir los delimitadores adecuados, ya que no hay ninguna garantía de que TCP genere un segmento por cada mensaje recibido de la aplicación. TCP puede tomarse la libertad de agrupar o separar los datos recibidos de la aplicación según le convenga; por ejemplo, podría decidir retener los mensajes recibidos de la aplicación para agruparlos y crear segmentos de mayor tamaño usando así la red de manera más eficiente.

Para poder enviar datos prioritarios y responder ante situaciones urgentes existen dos mecanismos extraordinarios por los que las aplicaciones pueden indicar a TCP su deseo de enviar rápidamente datos al otro extremo.

Uno de ellos es el denominado indicador PUSH. Cuando una aplicación desea que los datos entregados a TCP salgan inmediatamente, sin esperar más datos de la aplicación, lo indica poniendo a 1 el indicador PUSH; este indicador se utiliza por ejemplo cuando en una transferencia FTP se envía el último segmento, o cuando en una sesión telnet el usuario pulsa la tecla return<sup>3</sup>; en estos casos si no se utilizara PUSH se correría el riesgo de que TCP se quedara esperando mas datos de la aplicación para construir un segmento mayor.

El otro mecanismo de envío rápido de datos se denomina *datos urgentes*, y como su nombre indica es aún más expeditivo que el anterior. Por ejemplo en una sesión telnet se utiliza este procedimiento para enviar los caracteres DEL, CTRL-C, o cuando se pulsa la tecla BREAK o ATTN. En estos casos no solo se desea enviar cuanto antes los caracteres recién tecleados, sino que se quiere que estos pasen por delante de cualesquiera otros que hubiera pendientes de enviar a la aplicación y se le entreguen a ésta sin esperar a que los solicite (por ejemplo para abortar una ejecución en marcha cuando ésta no espera leer datos).

La existencia de datos “Pushed” y de datos urgentes se indica mediante los correspondientes bits indicadores o “flags” en la cabecera TCP. En el caso de datos urgentes se indica además el punto en el segmento donde terminan éstos; es responsabilidad de la aplicación averiguar en qué punto del segmento empiezan los datos urgentes.

---

<sup>3</sup> En realidad esta afirmación solo es correcta cuando se utiliza telnet con eco local; en la mayoría de los casos el telnet utiliza eco remoto, es decir el telnet remoto (servidor) ha de procesar lo que se teclea carácter a carácter, realizando la representación en pantalla si procede. Cuando se funciona con eco remoto el telnet cliente pone el bit PUSH para cada carácter que se teclea. Aunque es mucho menos eficiente que el eco local el eco remoto es hoy en día la forma de funcionamiento más habitual, ya que muchos editores de pantalla completa (el vi por ejemplo) necesitan eco remoto para funcionar correctamente.

### 6.3.5. Intercambio de información en TCP

El intercambio de segmentos en TCP se desarrolla de acuerdo con un protocolo de ventana deslizante con un número de secuencia de 32 bits. El número de secuencia cuenta bytes transmitidos, por lo que la secuencia se reinicia cada 4 GB (equivalentes a 4,3 minutos en una línea SDH de 155 Mb/s, suponiendo que toda la capacidad útil de la línea se utilizara para esa conexión TCP).

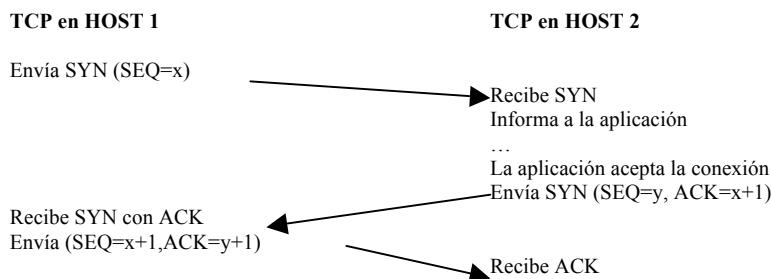
TCP utiliza ACK piggybacked, por lo que en la cabecera de cada segmento hay previstos dos campos de 32 bits, uno para el número de secuencia y otro para el número de ACK. El campo número de secuencia indica el número del primer byte transmitido dentro de ese segmento. El campo ACK indica el número del primer byte que se espera recibir en el siguiente segmento (o sea se sigue el estilo del campo “next” en HDLC). En la práctica el ACK piggybacked raramente se aprovecha ya que la mayoría de las aplicaciones generan tráfico muy asimétrico; normalmente uno de los dos TCP se ve obligado a enviar muchos segmentos vacíos con el único fin de informar al emisor que los datos han sido recibidos.

El mecanismo normal de funcionamiento de TCP es retroceso n, aunque también puede utilizarse repetición selectiva si las dos entidades participantes lo soportan y lo negocian al iniciar la conexión.

Cada segmento enviado incluye un tamaño de ventana en el que el receptor anuncia la cantidad de datos que está dispuesto a recibir. De esta forma el receptor puede ejercer control de flujo sobre el emisor. El tamaño de ventana es un campo de 16 bits, por lo que el valor máximo es de 64 Kbytes.

### 6.3.6. Gestión de conexión TCP

El mecanismo utilizado en TCP para establecer una conexión es el de saludo a tres vías. Un proceso normal sería el siguiente:



En el primer segmento el host 1 indica al host 2 que desea establecer una conexión (bit SYN puesto) y le informa del número de secuencia que ha elegido (x); el host 2 le responde con otro segmento en el que acepta la conexión (bit SYN puesto) y le indica el número de secuencia que él ha elegido para las transmisiones en el sentido contrario (y); además le acusa recibo de su número de secuencia al indicarle en el ACK que el próximo byte que espera recibir de él es x + 1. Host 1 responde con un tercer mensaje en el que acusa recibo del número de secuencia de host 2.

En este ejemplo hemos supuesto el caso más simple de establecimiento de una conexión. El Host 1 podría incluir ya en esos primeros segmentos datos dirigidos a la aplicación; esto está permitido siempre y cuando los datos sean retenidos por el TCP receptor hasta que la aplicación correspondiente decida si acepta la conexión.

Una vez establecida la conexión puede empezar el intercambio de información; cada lado puede enviar datos al otro de forma independiente, sin necesidad de que el otro le corresponda con más datos. Normalmente si fluyen datos en ambos sentidos los ACK irán incluidos (“piggybacked”) en los segmentos de vuelta, pero si el tráfico discurre predominantemente en un sentido (como es lo normal) se producirán segmentos vacíos con el único fin de acusar recibo de los envíos. Los valores de SEQ y ACK se van incrementando a medida que progresla la transmisión. Los valores de ventana anunciados por cada host permiten al otro conocer la disponibilidad que tiene de recibir datos.

El número de secuencia inicial es elegido por cada host de forma pseudoaleatoria. El RFC 793, que describe el estándar TCP, recomiendan utilizar para esto un contador que se incremente en una unidad cada 4μs, aproximadamente; esto se puede conseguir fácilmente basándose en algún reloj del sistema. Con este algoritmo el número de secuencia se da la vuelta cada 4 horas 46 minutos, aproximadamente.

Esta forma de elegir el número de secuencia inicial reduce la probabilidad de que si uno de los dos hosts cae y rearanca pueda coincidir el número de secuencia nuevo con el viejo, lo cual podría producir que se tomaran como válidos segmentos duplicados retrasados, o que el TCP del otro lado continuara dialogando con el nuevo proceso como si fuera el viejo. Para aumentar aún más la seguridad el estándar recomienda que se garantice una separación mínima en el tiempo de 2 minutos desde que cae un TCP hasta que se levanta el nuevo, para asegurar de esa forma que no pueden aparecer en el TCP de destino duplicados retrasados que puedan mezclarse en la nueva conexión (2 minutos es un valor máximo bastante normal del parámetro TTL o tiempo de vida, que fija el tiempo máximo que un datagrama puede estar pululando por la red antes de desaparecer).

Para comprender hasta qué punto es importante la no coincidencia de números de secuencia entre sesiones imaginemos la siguiente situación: dos usuarios X e Y mantienen ambos una conexión telnet desde la máquina 167.172.23.43 a la máquina 144.38.76.3; en un primer momento sus conexiones son:

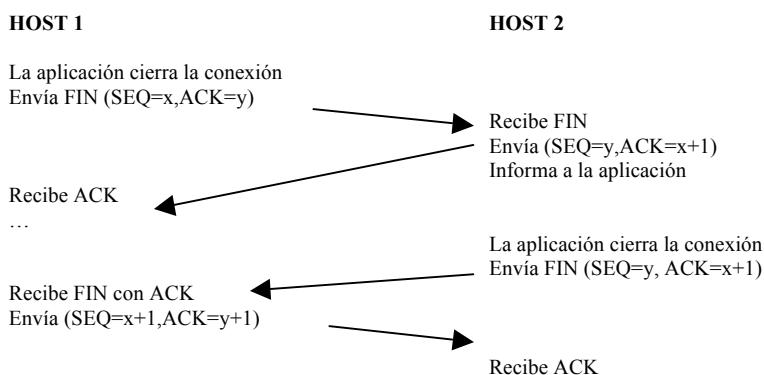
Usuario X:	167.172.23.43.1024 con 144.38.76.3.23
Usuario Y:	167.172.23.43.1025 con 144.38.76.3.23

1024 y 1025 son los puertos utilizados por los procesos clientes telnet en 167.172.23.43 y 23 es el puerto utilizado por el proceso servidor telnet en 144.38.76.3.

Supongamos ahora que el host 167.172.23.43 (cliente) cae y se levanta a continuación, y que los dos usuarios reanudan sus respectivas sesiones telnet (que han quedado medio abiertas en el servidor) pero ahora se conecta Y antes que X. Las conexiones antiguas han desaparecido, y dado que los puertos se asignan por orden de llegada ahora se asignarán en orden inverso:

Usuario X:	167.172.23.43.1025 con 144.38.76.3.23
Usuario Y:	167.172.23.43.1024 con 144.38.76.3.23

En condiciones normales los clientes telnet intentarán abrir nuevas conexiones, con lo que el servidor cerrará las antiguas. Pero si de alguna forma los clientes telnet entraran en las conexiones viejas del servidor sin cerrarlas cada usuario entraría directamente en la sesión del otro sin necesidad de identificarse con la contraseña correspondiente, lo cual evidentemente no es aceptable. La utilización de números de secuencia grandes y aleatorios suministra un cierto nivel de seguridad ante estas situaciones. Para terminar una conexión TCP se utiliza normalmente el procedimiento simétrico del saludo a tres vías, en el cual cada lado cierra su parte de forma independiente como si se tratara de una conexión simplex. El intercambio de mensajes típico es el siguiente:



### 6.3.7. Estados de TCP

El software (o proceso) TCP de un host puede mantener en un momento dado múltiples conexiones simultáneas con homólogos suyos en otros hosts. Para cada una de esas conexiones TCP conserva en todo momento información del estado en que se encuentra (por ejemplo conexión establecida, pendiente de establecer o no conexión).

La secuencia de estados que se suceden en el establecimiento de una conexión TCP en un servidor aparece en la tabla

Estado del servidor	Evento	Descripción
CLOSED		Estado (ficticio) en que se encuentra una conexión antes de existir
	La aplicación en el servidor hace una apertura pasiva	

LISTEN		El servidor espera una conexión del cliente
	El TCP del servidor recibe un SYN, devuelve un SYN/ACK	
SYN-RECEIVED		El servidor espera un ACK
	El TCP del servidor recibe un ACK	
ESTABLISHED		El ACK ha sido recibido y la conexión se ha establecido

Secuencia de estados que ocurren en el establecimiento de una conexión TCP en un servidor

Mientras que la siguiente tabla nos muestra la secuencia equivalente en el caso de un cliente.

Estado del cliente	Evento	Descripción
CLOSED		Estado (ficticio) en que se encuentra una conexión antes de existir
	La aplicación cliente solicita una conexión. El TCP del cliente envía un SYN	
SYN-SENT		El TCP cliente ha enviado un SYN al servidor
	El TCP cliente recibe SYN/ACK y envía ACK	
ESTABLISHED		La transferencia de datos puede comenzar

Secuencia de estados que ocurren en el establecimiento de una conexión TCP en un cliente

Una vez en el estado ESTABLISHED ambos TCP permanecerán así hasta que se inicie el procedimiento de cierre de la conexión. En condiciones normales cualquiera de los dos procesos puede iniciar la desconexión enviando un segmento con el bit FIN puesto; la secuencia de estados en el TCP que inicia la desconexión es la que se muestra en la tabla

Estado del TCP que cierra la conexión	Evento	Descripción
ESTABLISHED	La aplicación local solicita cerrar	
	TCP envía FIN/ACK	
FIN-WAIT-1		El TCP local está esperando respuesta del otro lado. En este punto aún pueden llegar datos válidos.
	TCP recibe un ACK	
FIN-WAIT-2		El TCP local ha recibido un ACK del otro lado, pero no un FIN. Se siguen aceptando los datos que pudieran llegar del otro lado
	TCP recibe FIN/ACK. Envía ACK	
TIME-WAIT		La conexión se mantiene en el limbo ante la posibilidad de recibir aun datos duplicados o un FIN duplicado. El tiempo de espera es igual al doble del tiempo de vida de un segmento
CLOSED		Se borra toda la información relativa a esta conexión.

Secuencia de estados que ocurren en el cierre de una conexión TCP en el host que inicia la desconexión

Mientras que la siguiente tabla muestra la secuencia de estados en el TCP que es “invitado” a cerrar la conexión.

Estado del TCP “invitado” a cerrar	Evento	Descripción
ESTABLISHED	TCP recibe FIN/ACK	
CLOSE-WAIT		Ha llegado un FIN
	TCP envía un ACK	
		TCP espera a que su aplicación cierre la conexión. La aplicación podría aún enviar más datos
	La aplicación local cierra la conexión	
	TCP envía FIN/ACK	
LAST-ACK		TCP Está esperando el ACK final
	TCP recibe un ACK	
CLOSED		Se borra toda la información sobre la conexión

Secuencia de estados que ocurren en el cierre de una conexión TCP en el host que es invitado a terminar la conexión

Los nombres utilizados en estas tablas corresponden con los empleados en el RFC 793, y en muchas implementaciones de TCP. El comando *netstat -an*, que permite examinar el estado de las conexiones existentes en un host UNIX, utiliza esta misma nomenclatura.

### 6.3.8. Conexiones medio abiertas y timer de keepalive

En principio el estándar TCP no establece el requerimiento de que una conexión TCP tenga un tráfico mínimo para permanecer establecida. Cabría pensar en la posibilidad de que una conexión TCP estuviera abierta durante días sin transmitir ni un solo segmento, y en principio no habría razón para terminarla. En la práctica esto supone que si en una conexión desaparece uno de los dos TCP el otro podría quedar eternamente esperando dándose lo que se denomina una conexión medio abierta. Si todo funcionara como es debido las conexiones medio abiertas nunca deberían ocurrir, puesto que cada TCP debería cerrar ordenadamente sus conexiones. Pero a veces los procesos TCP terminan de forma abrupta, no dando tiempo al cierre ordenado de sus conexiones; esto es especialmente común en tiempos recientes a partir de la proliferación de los ordenadores personales conectados directamente a Internet en los que el usuario con frecuencia no termina los procesos de la forma correcta, o desconecta de la red su ordenador dejando las conexiones medio abiertas en el otro lado.

Esas conexiones medio abiertas consumen recursos inútilmente, por lo que es conveniente establecer algún mecanismo por el cual un TCP pueda “sondear” periódicamente sus conexiones para comprobar que el otro lado aún está operativo; si el TCP de una conexión deja de responder durante un número determinado de mensajes de sondeo se considera que esa conexión está medio abierta y se procede a cerrarla de la forma más civilizada posible, liberando así los recursos que está utilizando.

Los mensajes de sondeo son lo que se conoce como mensajes “keep-alive” y la periodicidad con la que se producen viene fijada por el parámetro denominado timer de keepalive. Los mensajes de keepalive fueron añadidos a posteriori al TCP, y se implementan de una forma muy sencilla: el TCP envía un segmento que repite el último byte enviado; el TCP receptor no pasará este byte a la aplicación, pero debe generar un segmento de ACK; con esto el emisor ya sabe que su interlocutor está operativo.

Los keepalive se suelen implementar en servidores, que son los que más sufren el problema de las conexiones medio abiertas. Por la forma como se implementan los mensajes de keepalive funcionan aun cuando el TCP remoto no implemente keepalive, ya que se emplea una característica que forma parte del funcionamiento estándar de TCP. El mecanismo de keepalive no debe ser tan estricto que una pérdida momentánea de conectividad produzca el cierre de una conexión TCP, ya que en una red como Internet es fundamental permitir que las cosas funcionen aun cuando haya fallos momentáneos en el nivel de red, que sabemos que no es fiable. El timer de keepalive puede tener valores en torno a los dos minutos; los tiempos recomendados para cortar una conexión TCP inactiva pueden llegar a ser de hasta dos horas.

### 6.3.9. Política de transmisión de TCP

El receptor en una transmisión TCP anuncia regularmente el tamaño de ventana que tiene disponible para que el emisor sepa cuantos datos más puede enviarle. Cuando un receptor tiene lleno su buffer anuncia una ventana de 0 bytes, con lo que el emisor queda bloqueado hasta nueva orden.

Existen dos circunstancias en las que con ventana cero (ventana cerrada) el emisor puede enviar datos. Una es cuando se presentan datos urgentes; estos siempre deben ser aceptados por el receptor, ya que se supone que no pueden esperar. La otra excepción es que el emisor puede siempre enviar un segmento de un byte de datos, para forzar al receptor a devolver un ACK y así comprobar cual es la situación; esto evita situaciones de bloqueo que de otro modo podrían producirse por la pérdida de segmentos ACK en la red. La periodicidad con la que el emisor “tantea” al receptor con segmentos de un byte para comprobar que su ventana sigue cerrada se fija con el parámetro conocido como timer de persistencia.

Salvo por lo requerido en el bit PUSH o en datos urgentes, los emisores TCP pueden organizarse los envíos como mejor les convenga. TCP podría por ejemplo decidir agrupar la información que recibe de la aplicación para enviar segmentos mas grandes y así reducir el overhead de proceso y de cabeceras que supone el envío de segmentos pequeños. Incluso el uso del bit PUSH no garantiza la inmediata expedición de un segmento, en situaciones de verdadera congestión el TCP puede decidir ignorar el bit Push y agrupar varios mensajes de la aplicación en el mismo segmento.

### 6.3.10. Problemas de paquetes pequeños

#### 6.3.10.1. Algoritmo de Nagle

Un caso extremo de baja eficiencia se produce cuando la aplicación en el lado del emisor genera un byte de datos cada vez; imaginemos por ejemplo lo que ocurre cuando se efectúa una conexión telnet mediante una emulación de terminal ANSI (VT100 o similar); muchos programas, como por ejemplo el editor vi de UNIX, necesitan para funcionar correctamente que el host remoto procese cada carácter que se teclea, por lo que es preciso utilizar el modo de eco remoto mediante el cual el host remoto se encarga de representar en pantalla cada carácter que se teclea. En estas condiciones por cada tecla pulsada la aplicación envía a TCP los caracteres uno a uno; en principio TCP debería enviar cada carácter en un segmento contenido 20 bytes de cabecera al cual el nivel de red añadiría 20 de la cabecera IP; ese segmento TCP recibe a continuación su correspondiente segmento vacío (es decir sin datos) con el ACK del anterior. Como la representación en pantalla en el host local se realiza a través del host remoto instantes más tarde la aplicación del sistema remoto (el servidor telnet) responde con el eco del carácter pulsado, que es de nuevo un datagrama de 41 bytes, a lo cual el host cliente responde con otro segmento ACK vacío de 40 bytes; en total se transfieren 162 bytes para dos caracteres transmitidos, lo cual da una eficiencia del 1,2% (2/162) (y aquí no hemos considerado la información de control del nivel de enlace, tramas Ethernet por ejemplo, que añadiría aún más overhead).

Para evitar estas situaciones la mayoría de las implementaciones de TCP fijan un timeout de 500 mseg antes de enviar un ACK vacío; si en ese tiempo se genera algún tráfico de vuelta el ACK puede viajar “piggybacked” en él; por ejemplo en nuestro caso salvo que el host estuviera muy cargado la aplicación telnet respondería antes de 500 mseg y el ACK podría viajar en el mismo segmento que lleva el carácter de vuelta; de esta forma la eficiencia mejora ya que se suprime un ACK; la eficiencia sería entonces del 1,6 % (2/122).

Para mejorar aún más la eficiencia en estas situaciones se utiliza lo que se conoce como el *algoritmo de Nagle*. La idea es muy simple: cuando el tráfico de la aplicación llega al TCP en bytes uno por uno se envía el primero en un segmento y se retienen los demás hasta recibir el ACK correspondiente al byte enviado; una vez recibido el ACK se envía un segmento con todos los bytes que hubiera pendientes y se empieza a acumular de nuevo hasta recibir el siguiente ACK. También se envía un segmento si el número de caracteres acumulados en el buffer es igual a la mitad de la ventana, o al máximo tamaño del segmento.

En cierto modo el algoritmo de Nagle sustituye el protocolo de ventana deslizante por un mecanismo de parada y espera.

El algoritmo de Nagle es autoadaptativo, ya que en una red muy cargada los ACK tardarán más en llegar, con lo que automáticamente los segmentos que se envíen serán mayores y el overhead disminuirá; el usuario observará una latencia mayor en la red, pero esto es un mal menor cuando se trata de reducir la congestión en la red. Cuando la red esté descargada y responde muy rápido cada carácter tecleado viaja en un segmento independiente, y será respondido con otro que contendrá el carácter de eco, sin más retardo que el tiempo que tarde el host en responder.

El algoritmo de Nagle se puede aplicar a datos “Pushed” pero no a datos urgentes, y se debe inhibir cuando se desea transferir información en tiempo real; por ejemplo la posición del ratón en una sesión X-Windows debe ser transmitida inmediatamente ya que de lo contrario el movimiento en pantalla resulta errático (como es bien sabido el uso de terminales X es poco eficiente y requiere gran cantidad de recursos).

#### 6.3.10.2. Síndrome de la ventana tonta y solución de Clark

Imaginemos ahora la situación inversa; en vez de enviar los datos byte a byte es la aplicación en el TCP receptor la que recoge los bytes de uno en uno. La situación que se daría sería la siguiente:

1. El TCP emisor envía datos sin parar al TCP receptor
2. El buffer del TCP receptor se llena
3. El TCP receptor notifica al emisor que su ventana está cerrada (ventana 0)
4. El TCP emisor deja de enviar datos
5. La aplicación receptora lee 1 byte del buffer de TCP

6. El TCP receptor envía un ACK al emisor para anunciarle que dispone de 1 byte de espacio
7. El TCP emisor envía un segmento con 1 byte de información útil
8. Volvemos al punto 2.

El bucle se repite hasta terminar la sesión. Se están generando como antes segmentos con un byte de información útil, pero esta vez el causante es el receptor que los recoge en pequeñas dosis. Este comportamiento se conoce como *síndrome de la ventana tonta*.

La solución a esto, propuesta por Clark en el RFC 813, consiste en que el TCP del receptor no debe notificar el cambio de ventana al emisor entretanto no tenga una cantidad razonable de espacio libre en su buffer; por razonable se entiende cualquiera de las dos condiciones siguientes: el espacio suficiente para aceptar un segmento de la longitud máxima admitida en esa conexión, o la mitad del espacio total del buffer.

El algoritmo de Nagle y la solución de Clark son dos mecanismos complementarios que intentan resolver un mismo problema: el causado por la transmisión innecesaria de segmentos pequeños.

### 6.3.11. Control de congestión en TCP

Como ya hemos explicado el TCP receptor puede controlar el flujo de datos que recibe del emisor anunciando un valor adecuado del tamaño de ventana. Si el receptor anuncia un tamaño de ventana 0 (lo que se conoce como “cerrar la ventana”) el emisor dejará de transmitir hasta nueva orden (salvo por las dos excepciones ya mencionadas, datos urgentes y segmentos con un byte).

Sin embargo la falta de espacio de buffer en el receptor es solo una de las causas por las que el emisor puede verse obligado a bajar el ritmo de emisión; la otra es la presencia de congestión en la red. Para aclarar la diferencia entre estos dos importantes conceptos imaginemos el siguiente experimento:

- Un supercomputador establece una conexión TCP con un ordenador personal poco potente a través de una conexión directa ATM OC3 (155,52 Mb/s) “back to back”. El circuito se establece mediante la categoría de servicio UBR. Al transferir datos se mide un rendimiento máximo de 50 Mb/s; analizando la situación se observa que el ordenador personal tiene su buffer lleno prácticamente todo el tiempo, por lo que su TCP está continuamente cerrando su ventana; evidentemente el subsistema de entrada/salida del ordenador personal no es capaz de absorber los datos al ritmo que los envía el supercomputador.
- En una segunda prueba se repite la misma transferencia entre ambos ordenadores, pero en vez de un enlace directo se conectan a través de una red ATM pública (utilizando también un servicio UBR). En este caso el rendimiento de la transferencia en horas punta es de solo 10 Mb/s, aun cuando se observa que el TCP del ordenador personal tiene espacio de sobra en sus buffers de entrada.

La diferencia estriba en que en el primer caso está actuando como factor limitante el control de flujo en el receptor y en el segundo la congestión en la red. Presumiblemente la red pública no es capaz en horas punta de dedicar a esa conexión los recursos necesarios para transmitir los 50 Mb/s que puede absorber el ordenador personal. Basta que uno solo de los enlaces del trayecto se vea afectado de congestión para limitar el tráfico en todo el trayecto, y por tanto el rendimiento de la comunicación entre los hosts.

Imaginemos por un momento que pasaría si TCP no incluyera ningún mecanismo de control (mejor dicho de autocontrol) en situaciones de congestión; dado que el receptor anuncia buffers disponibles el emisor inyectaría paquetes en la red al ritmo que se lo permita su conexión física; cuando los paquetes lleguen a la parte congestionada de la red, no pudiendo aceptar todo el tráfico entrante, los routers empezarán a acumularlos en sus buffers y cuando estos se llenen los descartarán; en el lado del TCP receptor se recibirán solo una parte de los segmentos, por lo que o bien se devolverán segmentos NAK al emisor, o bien sencillamente no se enviarán los ACKs y se esperará que el emisor reenvíe por timeout; en cualquier caso el emisor tiene que reenviar datos. Los segmentos descartados en ruta son inútiles y cargan innecesariamente las líneas por las que pasan; este tráfico inútil podría propagar la congestión a zonas de la red que en principio no la sufrían, aumentando así la magnitud del problema y disminuyendo aún más el rendimiento en toda la red. Este efecto de “bola de nieve” se denomina colapso de congestión (“congestion collapse”) y puede llegar a dejar toda una red completamente fuera de servicio.

Evidentemente TCP no puede ser ajeno a las situaciones de congestión en la red, pero como notificamos al emisor que hay saturación en algún punto del trayecto y que debe bajar el ritmo con que envía datos? TCP no emplea mecanismos explícitos para notificar la congestión. El mecanismo que emplea es indirecto y se basa en la pérdida de datagramas por la red. Esto se basa en una premisa fundamental: el medio físico se considera *altamente fiável* por lo que siempre que el TCP emisor detecte que los segmentos no han sido recibidos en su destino (al no recibir los correspondientes ACKs) deducirá que la red está descartando paquetes por congestión y bajará el ritmo de sus envíos. Dado que el control de congestión de TCP se basa en la fiabilidad del medio físico cuando esta condición no se cumple (radioenlaces móviles, por ejemplo) es preciso realizar modificaciones a los algoritmos normales de TCP o incorporar en el nivel de enlace mecanismos de comprobación o corrección de errores que suministren esa fiabilidad, ya que de lo contrario TCP interpreta como congestión los problemas debidos al medio físico y si en estas condiciones se baja el ritmo el rendimiento decrece aún más.

Para autoregularse el TCP emisor maneja una *ventana de congestión* que le indica que cantidad de datos puede inyectar en la red en un momento dado. Dicha ventana actúa simultáneamente y en paralelo a la ventana que anuncia el receptor indicando los buffers disponibles, que podríamos denominar *ventana de control de flujo*. En cada momento el emisor tomará en consideración la más pequeña de las dos ventanas, para asegurarse de que no satura al receptor y que tampoco provoca, o contribuye a agravar, una situación de congestión en la red.

Mientras que la ventana de control de flujo es notificada al emisor por el receptor, la ventana de congestión es calculada por el emisor a partir de la cantidad de retransmisiones que tiene que realizar; cuando ve que no se produce ninguna retransmisión va aumentando paulatinamente la ventana, hasta llegar al punto donde falla algún segmento (es decir, agota el timer y se retransmite), momento en el cual la reduce (suponiendo que la ventana de control de flujo no imponga ninguna limitación). Generalmente la ventana crece de forma lenta y gradual, mientras que la reducción se lleva a cabo de manera drástica. Los algoritmos de crecimiento y disminución de la ventana de congestión en TCP son siempre autoadaptativos y forman una parte fundamental del rendimiento del protocolo; estos algoritmos han sido y son objeto de detallados estudios y experimentaciones, por lo que son altamente sofisticados y funcionan bien en situaciones muy diversas.

Inicialmente la ventana de congestión se fija a un valor igual al del máximo tamaño de segmento negociado en el momento de establecer la conexión (que depende a su vez del MTU); supongamos por ejemplo que dicho tamaño es de 1 Kbyte y que todos los segmentos que se van a generar tendrán este tamaño. Inicialmente TCP envía un segmento de 1 KByte e inicia un timer; si se recibe el ACK antes de expirar el timer significa que el segmento ha llegado a su destino correctamente, por lo que la ventana de congestión se amplía a 2 KBytes; a continuación se envían 2 segmentos, y se inicia el timer (en realidad 2 timers, uno por segmento); por cada segmento confirmado dentro del intervalo previsto se amplía la ventana en un segmento (1 KByte), por lo que, suponiendo que no se pierda ninguno, en el ciclo siguiente la ventana pasará de 2 a 4 KBytes; en condiciones normales esto supone que la ventana crece exponencialmente, ya que se duplica en cada envío; esta técnica se denomina *slow-start*, aunque no es precisamente lenta, sino todo lo contrario; por ejemplo empezando en 1 KByte en sólo 7 iteraciones llegaría a 64 Kbytes, valor máximo permitido por el tamaño de ventana de TCP. En condiciones normales (sin congestión) el slow-start provoca que la ventana de congestión crezca rápidamente, con lo que pronto supera a la ventana de control de flujo, momento a partir del cual prevalece ésta y la ventana de congestión deja de crecer.

Supongamos que la ventana de control de flujo del receptor es siempre mayor que la de congestión (y por tanto no se ejerce control de flujo) y que la pérdida de paquetes se presenta siempre justo cuando la ventana de congestión supera los 20 KBytes; supongamos también que todos los segmentos que se transmiten son de 1 Kbyte; la evolución de la ventana de congestión sería entonces la siguiente:

Fase	Umbral de peligro (en Kbytes)	Tamaños sucesivos de la ventana (en Kbytes)
Primera	64 (valor por defecto)	1,2,4,8,16,32
Segunda	16	1,2,4,8,16,17,18,19,20,21
Tercera	10,5	1,2,4,8,10,11,12,13,14,15,16,17,18,19,20,21

A partir de la tercera fase el proceso se repite indefinidamente. Cabría pensar que el mecanismo no se estabiliza nunca, y en efecto así es; aun en el caso de que fijáramos la ventana de congestión en el valor de 20 KBytes tampoco se estabilizaría, ya que en la práctica el tamaño de la ventana de congestión cambia continuamente. En última instancia la única forma que tiene el emisor de ajustar la ventana de

congestión al límite de sus posibilidades es tanteando y fallando de vez en cuando. Se podría argumentar que no es preciso retroceder en tan gran medida en caso de fallo, pero recordemos que cuando se produce congestión en una red es mejor pasarse de precavido, pues de lo contrario el problema se puede hacer inmanejable.

### 6.3.12. Gestión de timers en TCP

Hasta ahora hemos supuesto que TCP era capaz de detectar los segmentos perdidos fijando un valor adecuado para el timer de retransmisión; en realidad para estar completamente seguros de que no va a llegar el segmento ACK habría que esperar dos veces el valor del TTL, lo cual en prácticamente todos los casos es excesivo. Para el timer se suele elegir un tiempo por encima del cual la probabilidad de recibir el ACK sea muy pequeña. La elección de un valor adecuado para este timer tiene una consecuencia directa en el funcionamiento eficiente de TCP; si el timer es demasiado alto el emisor esperará innecesariamente en muchos casos por ACKs que nunca llegarán, y si es demasiado bajo se producirán reenvíos innecesarios de segmentos que habían sido correctamente recibidos.

La elección de valores de timer adecuados es mucho más compleja en el nivel de transporte que en el nivel de enlace. Además de las fluctuaciones naturales debidas a las diferencias en capacidad y retardo de unas conexiones a otras, el nivel de transporte ha de hacer frente a oscilaciones debidas a la presencia de elementos intermedios (routers y enlaces) y de situaciones de congestión que están fuera de su control; aun en ausencia de congestión los routers pueden tener largas colas de paquetes que atender, los enlaces pueden ser de diversas velocidades, y la ruta puede variar durante la conexión.

Por todo esto los valores del timer de retransmisión en el nivel de transporte se establecen mediante algoritmos autoadaptativos que dinámicamente ajustan los valores al estado de la red, según es percibido éste por el nivel de transporte en el host emisor.

El algoritmo utilizado en TCP para el cálculo de los timers fue diseñado por Van Jacobson, que es también el autor de la técnica slow-start que hemos visto antes. En realidad la gestión de los timers es una parte del slow-start necesaria para un efectivo control de la congestión en TCP.

Para estimar el timer de retransmisión TCP mide lo que tardan en llegar los ACK de los segmentos enviados; se supone que estos tiempos son una buena estimación del tiempo de ida y vuelta o RTT (Round Trip Time) de los segmentos, en base al cual ha de calcularse el valor del timer de retransmisión. El valor medio del RTT (que denominaremos MRTT) se estima mediante la fórmula iterativa siguiente:

$$MRTT_n = \alpha MRTT_{n-1} + (1-\alpha) RTT_n$$

donde  $RTT_n$  es el tiempo de ida y vuelta medido para el último ( $n$ -ésimo) ACK recibido. El parámetro  $\alpha$  permite ajustar el peso o la importancia que se quiere dar al último valor frente a los anteriores; con un  $\alpha$  pequeño se consigue que los valores anteriores tengan poca relevancia, adaptándose así a situaciones cambiantes con rapidez. Con  $\alpha$  grande se reacciona con más inercia a los cambios. En TCP  $\alpha$  vale normalmente 7/8. Lo que calcula esta fórmula es pues una media aritmética ponderada de los valores de RTT, dándoles un peso inversamente proporcional a su antigüedad (cuanto más viejo menos importante).

Obtener una buena estimación del valor medio de RTT resuelve sólo una parte del problema; sabemos que los valores de RTT se distribuirán alrededor del valor medio, pero cual es el valor adecuado del timer de retransmisión, o sea, ¿cuál es el valor umbral a partir del cual podemos considerar que el ACK no llegará?. Las primeras implementaciones de TCP utilizaban  $2*MRTT$  como valor del timer, pero eso tenía el inconveniente de que cuando los valores de RTT fluctuaban mucho el umbral de  $2*MRTT$  resultaba demasiado bajo y se producían excesivas retransmisiones innecesarias; en cambio cuando la dispersión era pequeña  $2*MRTT$  daba un valor excesivo ya que en la mayoría de los casos no había que esperar tanto para dar por perdido un segmento. Para tener una estimación más precisa del timeout necesitamos saber además del valor medio el grado de dispersión, o dicho de otro modo conocer la anchura de la campana de distribución de los valores, lo que en estadística se conoce como la desviación estándar. Para estimar esta magnitud de manera sencilla se utiliza la siguiente fórmula:

$$D_n = \beta D_{n-1} + (1-\beta) | MRTT_{n-1} - RTT_n |$$

Como antes  $\beta$  es un factor que permite regular la inercia a los cambios (a mayor  $\beta$  mayor inercia); normalmente suele valer  $3/4^4$ . De forma parecida al cálculo de MRTT el valor actual es una media ponderada del valor instantáneo y de los valores anteriores, con un peso decreciente en función de la antigüedad.

Una vez obtenidos MRTT y D podemos calcular el timeout de retransmisión. Para esto se utiliza generalmente la fórmula siguiente:

$$\text{Timeout de retransmisión} = \text{MRTT} + 4 * \mathbf{D}$$

El cálculo del RTT de los segmentos reenviados plantea un problema. No es posible saber con seguridad si el ACK se debe al primer o al segundo envío y una interpretación errónea podría alterar de manera importante el valor de MRTT. La solución a este problema, conocida como *algoritmo de Karn*, consiste sencillamente en ignorar a efectos del cálculo del MRTT (y de D) los ACK de los segmentos que son retransmitidos. Sin embargo esto plantea otro problema: podría ocurrir que el RTT aumentara de forma repentina (por ejemplo por un cambio en la ruta de los datagramas) hasta el punto que superara el timeout; a partir de ese momento todos los segmentos serían retransmitidos, con lo que los valores de MRTT y D (y por tanto el timeout de retransmisión) se mantendrían constantes, puesto que los ACK recibidos serían ignorados; estas múltiples retransmisiones provocarían una notable pérdida de eficiencia. Para evitarlo el algoritmo de Karn prevé que cada vez que se produzca una retransmisión el timeout de retransmisión se duplique; de esta forma si por alguna razón el RTT aumenta de forma repentina el timer de retransmisión crece rápidamente evitando así que se produzcan muchas retransmisiones; una vez el TCP emisor vuelve a recibir ACKs de segmentos no retransmitidos vuelve a calcular el timeout de retransmisión a partir de los nuevos valores de MRTT y D de acuerdo con la ultima fórmula. Esta técnica, denominada *retroceso exponencial* del timer, tiene el interesante efecto colateral de reducir el tráfico en situaciones de congestión.

### 6.3.13. Opciones del protocolo TCP

El protocolo TCP está en evolución permanente; las mejoras se experimentan en prototipos y luego se documentan en RFCs, convirtiéndose en extensiones opcionales al protocolo básico. Las extensiones son generalmente compatibles entre sí y se van incorporando paulatinamente en muchas implementaciones de TCP. Cuando dos TCPs conectan negocian entre ellos la relación de extensiones que cada uno quiere utilizar, y emplean solo aquellas que están soportadas por ambos. Veamos algunas de esas extensiones.

El tamaño de ventana estándar de TCP es de 64 KBytes. El tamaño de ventana establece la máxima cantidad de datos que pueden estar pendientes de confirmación en una comunicación. Cuando la comunicación utiliza un canal de elevada capacidad o gran latencia (es decir elevado valor de RTT) es posible que el emisor tenga que esperar a recibir confirmación antes de seguir enviando; por ejemplo en una comunicación vía satélite es normal tener valores de RTT de 500 mseg; si un host transmite datos a otro mediante TCP por un enlace vía satélite de 2 Mb/s no podrá enviar más de 64 Kbytes cada 500 mseg, ya que una vez ha llenado la ventana tiene que esperar a recibir el ACK; pero 64 Kbytes cada 500 mseg equivale a  $64 * 1024 * 8 / 0,5 = 524.288 / 0,5 = 1,049$  Mb/s, por lo que el enlace solo podrá aprovecharse al 50% aproximadamente. En general el rendimiento de una conexión siempre vendrá limitado por la fórmula:

$$\text{Capacidad máxima} = \text{Tamaño de ventana} / \text{RTT}$$

Es decir, siempre que el producto capacidad\*RTT de una conexión sea superior al tamaño de ventana el rendimiento vendrá limitado por ésta ya que la conexión no es capaz de “llenar la tubería” de datos y se producirán tiempos de espera en la comunicación. Cuando se diseñó TCP era impensable tener este tipo de problemas, que aparecieron inicialmente con la disponibilidad de enlaces vía satélite de alta velocidad (2 Mb/s). Hoy en día la posibilidad de tener enlaces de alta velocidad en redes de área extensa plantea otras situaciones en las que también se da este problema, por ejemplo una comunicación TCP sobre una línea ATM OC3 de 155,52 Mb/s puede obtener con la ventana estándar un ancho de banda máximo de  $524.288 / 0,008 = 65,5$  Mb/s; para aprovechar completamente la línea sería precisa una ventana mínima

---

<sup>4</sup> Se suelen preferir en estas fórmulas los factores que son potencias enteras de 2 en el denominador ya que esto hace más sencillos, y por tanto más rápidos, los cálculos.

de  $155.520.000 * 0,008 = 1.244.160$  bits = 152 Kbytes<sup>5</sup>. El RFC 1323, incorporado en muchas implementaciones de TCP, resuelve este problema ya que permite utilizar un factor de escala para ampliar el tamaño de ventana hasta  $2^{30}$  (1 GByte).

Otro campo en el que TCP ha sido mejorado es el tipo de actuación en caso de errores. Las primeras implementaciones utilizaban retroceso n, con lo que cuando se perdían segmentos el rendimiento caía de forma apreciable. El RFC 1106, incorporado en muchas implementaciones, prevé el funcionamiento con repetición selectiva, de manera que el emisor sólo tiene que reenviar el segmento o segmentos que han llegado erróneos.

RFC 1106 también propone un mecanismo de envío de acuses de recibo negativos (NAK) cada vez que se recibe un segmento sin haber recibido el anterior. Sin embargo el RFC 1106 también toma en cuenta que al ser transportados en datagramas los segmentos pueden no llegar en orden, por lo que la no recepción de un segmento en el momento esperado no quiere decir necesariamente que éste se haya perdido; lo que hace el emisor en estos casos es esperar a que el receptor insista, por ejemplo si le comunica que ha recibido los tres segmentos siguientes a uno dado y sigue faltando éste es bastante probable que se haya perdido, con lo que lo reenvía sin esperar a agotar el timeout. Evidentemente este mecanismo solo se aplicará si la retransmisión no ha sido provocada antes por agotamiento del timer de retransmisión.

#### 6.3.14. UDP (User Datagram Protocol)

TCP tiene la robustez y funcionalidades propias de un protocolo de transporte orientado a conexión; sin embargo esa robustez y funcionalidad conllevan una cierta complejidad; por ejemplo cualquier transmisión de información TCP requiere como mínimo el intercambio de seis mensajes para establecer la comunicación y terminarla; además mientras una conexión existe ocupa una serie de recursos en el host. A veces no se requiere toda esa funcionalidad; en esos casos se prefiere que el nivel de transporte preste un servicio más sencillo, no orientado a conexión y no fiable (por no fiable queremos decir que el receptor no acusa recibo de las TPDUs enviadas). Algunos ejemplos de situaciones en las que es más conveniente un servicio no orientado a conexión son las siguientes:

- El tipo de aplicación no requiere una fiabilidad total y no puede tolerar el retardo producido por los ACKs y las retransmisiones de TCP; este es el caso por ejemplo en la transmisión de vídeo o audio en tiempo real.
- La aplicación por su propia naturaleza requiere el envío de uno o dos mensajes únicamente; ejemplos de estas aplicaciones son las siguientes: sincronización de relojes (NTP), consultas al servidor de nombres (DNS), mensajes de gestión de la red (SNMP), etc.; en el DNS no se considera necesario un transporte fiable porque si se pierde el datagrama la aplicación lo reenviará; en el caso de NTP o SNMP la pérdida de un datagrama no es importante porque la información se está actualizando a intervalos regulares (por ejemplo cada 5 minutos).
- Se desea hacer envíos multidiestino (multicast o broadcast); esto sólo es posible con un protocolo no orientado a conexión, ya que por su propia naturaleza los protocolos orientados a conexión son punto a punto (en TCP no es posible establecer conexiones multipunto).

El protocolo no orientado a conexión de Internet se conoce como UDP (User Datagram Protocol); su nombre ya da una idea de la naturaleza no fiable del servicio de transporte ofrecido. Entre las aplicaciones que utilizan UDP se encuentran TFTP (Trivial File Transfer Protocol), DNS (Domain Name Server), SNMP (Simple Network Management Protocol), NTP (Network Time Protocol) y NFS (Network File System)<sup>6</sup>, etc.

Las TPDUs intercambiadas por UDP se denominan *mensajes* o *datagramas UDP*. Recordemos que los mensajes UDP se identifican por el valor 17 en el campo protocolo del datagrama IP.

<sup>5</sup> En realidad la capacidad efectiva en IP de un enlace OC-3 después de restar el overhead de SDH y ATM es de unos 135 Mb/s, por lo que sería suficiente con una ventana de 132 Kbytes.

<sup>6</sup> La arquitectura de NFS presupone la no existencia de una conexión entre el servidor y el cliente, razón por la cual se adapta mejor a funcionar sobre UDP. En principio NFS fue una aplicación diseñada para su uso en redes locales, donde el retardo suele ser bajo y constante y la pérdida de datagramas muy rara. En este entorno UDP da normalmente un rendimiento aceptable. Sin embargo cuando se utiliza NFS en redes de área extensa el rendimiento puede no ser satisfactorio debido a la mayor fluctuación del retardo y la pérdida de datagramas debido a congestión. Para resolver estos problemas algunos fabricantes han desarrollado implementaciones de NFS que utilizan TCP como protocolo de transporte. Aunque supone una mejora interesante esto no está ampliamente disponible.

Una característica interesante de UDP es que puede ser utilizado por aplicaciones que necesitan soporte de tráfico multicast o broadcast. Con TCP esto no es posible debido a la naturaleza punto a punto, orientada a conexión del protocolo.

UDP no suministra ningún mecanismo de control de flujo o control de congestión. Cuando lo que se envía es únicamente un mensaje (por ejemplo una consulta al DNS) esto es innecesario, ya que presumiblemente un mensaje aislado no creará problemas de congestión y será siempre aceptado en destino (y de no ser así el mismo problema habría surgido con TCP para el inicio de la conexión). Si se va a enviar un flujo de mensajes, por ejemplo video o audio en tiempo real, se deberán tomar las medidas adecuadas para asegurar la capacidad suficiente en la red (mediante mecanismos de reserva de capacidad, por ejemplo) y evitar la congestión no excediendo lo solicitado en el momento de hacer la reserva; afortunadamente en estos casos se suele conocer a priori con bastante aproximación el tráfico que se va a introducir en la red.

En caso de congestión en la red parte de los datagramas serán descartados por la red sin informar por ningún mecanismo al emisor, ni al receptor. En caso de saturación del receptor este sencillamente ignorará los datagramas que no pueda aceptar. En algunos se contemplan a nivel de aplicación mecanismos de control que permiten al receptor detectar si se producen pérdidas (por ejemplo numerando los datagramas) informando al emisor para que baje el ritmo de emisión si se rebasa un umbral determinado.

De forma similar a los segmentos TCP, los mensajes UDP se dirigen a la aplicación adecuada mediante el puerto de destino, especificado en la cabecera. Análogamente a TCP los puertos UDP se identifican mediante un campo de 16 bits (números entre 0 y 65535). Aun en el caso de coincidir en número con un puerto TCP son TSAPs diferentes, como ya hemos comentado antes. Al igual que en TCP los valores por debajo de 1024 están reservados para los puertos denominados “*bien conocidos*” (*well-known ports*), aunque su significado es diferente en la mayoría de los casos, pues también lo son los servicios. La tabla 6.8 muestra algunos de los puertos UDP más utilizados.

Servicio	Puerto	Descripción
Echo	7	Devuelve el datagrama al emisor
Discard	9	Descarta el datagrama
Daytime	13	Devuelve la hora del día
Quote	17	Devuelve una “frase del día”
Chargen	19	Generador de caracteres
Nameserver	53	Servidor de nombres de dominios
Bootps	68	Puerto servidor utilizado para cargar información de configuración Bootp
Bootpc	68	Puerto cliente utilizado para recibir información de configuración
TFTP	69	Trivial File Transfer Protocol
SunRPC	111	Sun Remote Procedure Call
NTP	123	Network Time Protocol
SNMP	161	Usado para recibir consultas de gestión de la red
SNMP-trap	162	Usado para recibir avisos de problemas en la red

Algunos de los puertos UDP más utilizados

La estructura de un mensaje UDP se muestra en la siguiente tabla.

Campo	Longitud (bits)
Puerto origen	16
Puerto destino	16
Longitud	16
Checksum	16
Datos	0-524056 (65507 bytes)

Estructura de un mensaje UDP

A continuación describimos el significado de cada uno de los campos de la cabecera UDP:

- *Puerto origen* especifica el puerto de la aplicación que genera el mensaje. Este valdrá normalmente cero, salvo que la aplicación solicite una respuesta.
- *Puerto destino* especifica el puerto de la aplicación a la que va dirigido el mensaje.
- *Longitud* indica la longitud del mensaje, incluyendo los campos de cabecera.

- *Checksum*: el uso de este campo es opcional en IPv4, obligatorio en IPv6 (ya que en ese caso se ha suprimido el checksum a nivel de red). Cuando se envía información en tiempo real (audio o vídeo digitalizado) su uso puede omitirse. Para el cálculo se aplica el mismo algoritmo que en TCP (suma complemento a 1 de todo el mensaje dividido en campos de 16 bits, y complemento a 1 del resultado). En el cálculo se utiliza todo el mensaje, incluida la cabecera y se antepone una pseudo cabecera similar a la utilizada en TCP (con la dirección IP de origen, de destino, el tipo de protocolo de transporte y la longitud del mensaje) de forma que se verifica que sean correctos no solo los datos del mensaje UDP sino también los datos fundamentales de la cabecera IP. Si la verificación del checksum en el receptor da error el mensaje es simplemente descartado sin notificarlo al nivel de aplicación ni al emisor.
- *Datos* contiene los datos a transmitir. Un mensaje UDP ha de estar contenido necesariamente en un datagrama IP, lo cual fija la longitud máxima de este campo.

De la misma forma que un host o un router pueden tener que fragmentar un datagrama que contenga un segmento TCP, es posible que el host emisor o algún router intermedio tengan que fragmentar un mensaje UDP porque sea mayor que la MTU permitida en la red por la que ha de enviarse. Análogamente a los segmentos TCP la fragmentación ocurre de forma transparente a UDP y la cabecera del mensaje solo aparecerá en el primer fragmento; en cambio cada fragmento deberá incluir una nueva cabecera IP.

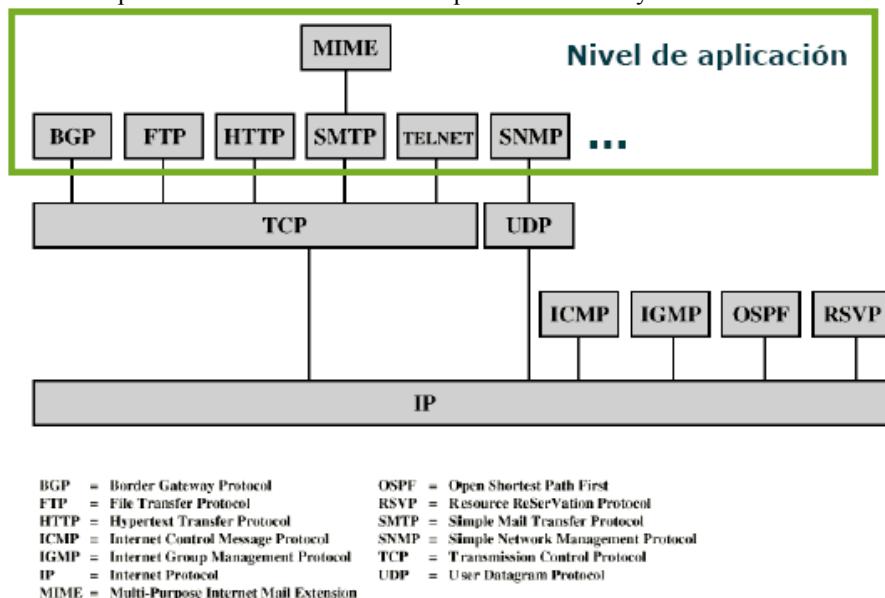
El funcionamiento del protocolo UDP está descrito en el RFC 768.

## Tema VII

### El nivel de Aplicación

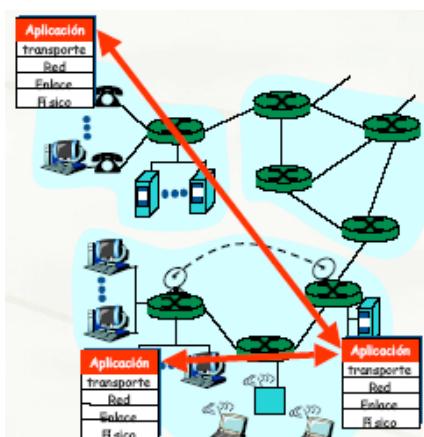
Está formado por un conjunto de protocolos

- Cada uno de ellos se utiliza para un propósito específico
- Cada uno de los protocolos es independiente
- Puede convivir varios dentro de una misma red y dispositivo
- Son utilizados por aplicaciones a las que se denomina servicios
- Utilizan como protocolos de nivel inferior los protocolos: TCP y UDP



Las aplicaciones son software

- Diferentes máquinas y Sistemas Operativos
- Quienes se comunican son procesos
- IPC: Inter Process Communication
- Nos interesan procesos ejecutándose en diferentes máquinas
- Se comunican a través de una red
- Intercambian mensajes
- Emplean Protocolos de nivel de aplicación...



Algunas aplicaciones en red son E-mail, Web, Mensajería instantánea, login remoto, Compartición de ficheros P2P, Juegos multiusuario en red, Streaming de video clips, Telefonía por Internet, Videoconferencia en tiempo real, Computación masiva en paralelo

- **Protocolos de servicios orientados al usuario**

- HTTP
- RTSP
- TELNET
- FTP
- SMTP
- SIP
- ...

Utilizados por servicios que el usuario utiliza directamente

- **Protocolos para servicios básicos**

- DNS
- DHCP
- NAT
- SNMP
- ...

Utilizados por servicios que se utilizan para el funcionamiento de la red u otros servicios

Cada servicio tiene su propio nivel de aplicación para comunicarse con las entidades de ese servicio

### 7.1 Aplicaciones y Protocolos.

Los Protocolos de aplicación son una parte de las aplicaciones de red.

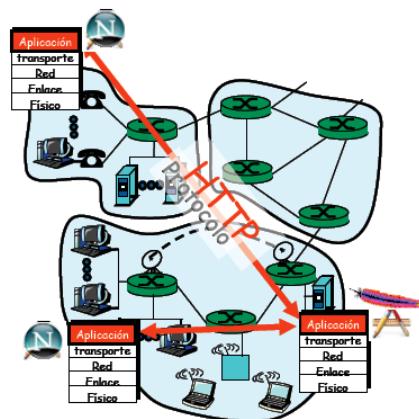
Definen:

- Tipos de mensajes
- Sintaxis/formato de mensajes
- Significado del contenido
- Reglas de funcionamiento

Ejemplo: La Web

- Navegador, Servidor Web...
- HTTP ...

Muchos protocolos son estándares abiertos (en RFCs)



### 7.2 Paradigmas

Son filosofías para escribir/organizar las aplicaciones distribuidas

- Cliente-servidor  
Asimetría, hay proveedores de servicios y usuarios de los servicios
- Peer-to-peer (P2P)  
Simetría, comunicación entre iguales (pares)
- *Híbrido de cliente-servidor y P2P*  
Una aplicación puede usar las dos filosofías para diferentes cosas

#### 7.2.1 Arquitectura cliente-servidor

**Servidor:** cualquier programa que ofrece un servicio que se puede obtener en una red. Un servidor acepta la petición desde la red, realiza el servicio y devuelve el resultado al solicitante.

En el caso de los servicios más sencillos, cada petición llega en un solo datagrama IP y el servidor devuelve la respuesta en otro datagrama

**Cliente:** un programa ejecutable se convierte en un cliente cuando manda una petición a un servidor y espera una respuesta

En general los servidores tienen dos partes:

- Un programa maestro sencillo, responsable de aceptar las nuevas peticiones
- Un conjunto de esclavos, responsables de manejar cada una de las peticiones

## TAREAS DEL MAESTRO:

- Abrir el puerto.** Abre un puerto conocido al que se puede acceder
- En espera del cliente.** Espera a que un cliente envíe una petición
- Elección de puerto.** Si es necesario, abre un nuevo puerto para procesar la petición
- Se inicia el esclavo.** Inicia un nuevo proceso concurrente para que procese la petición
- Continúa.** Regresa al paso de espera y continúa aceptando peticiones mientras los esclavos procesan las peticiones

## TÉCNICAS DE GESTIÓN DE ESCLAVOS

**Esclavo por petición:** cada vez que llega una petición se crea un esclavo para procesarla

**Esclavo por sesión:** cada vez que se inicia una sesión se crea un esclavo para gestionarla

**Sesión:** periodo entre el establecimiento de una conexión entre un cliente y un servidor. Una sesión puede contener una o varias peticiones

**Conjunto de esclavos:** el servidor tiene inicialmente un conjunto de esclavos activos inicialmente que va repartiendo según llegan las peticiones. Cuando estas terminan los esclavos se liberan (pero no se destruyen) hasta que llegue otra petición

## COMPLEJIDAD EN EL SERVIDOR

Deben mantener reglas de autorización y protección

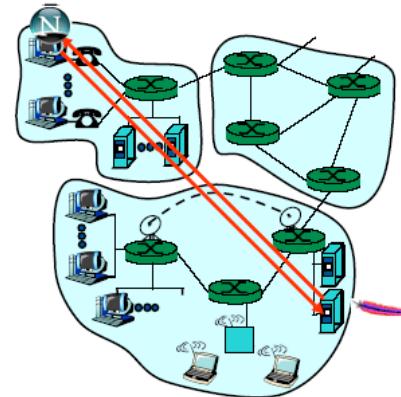
Comprobar usuarios

Restringir el acceso a ciertas zonas

No deben cumplir a ciegas las órdenes provenientes del exterior

Deben protegerse contra peticiones formadas equivocadamente y contra peticiones que causen el programa abortar

Típicos ataques a sistemas mediante este tipo de peticiones



### Servidor:

- Comienza a ejecutarse primero...
- Espera a ser contactado
- Host siempre disponible
- Dirección permanente

### Cliente:

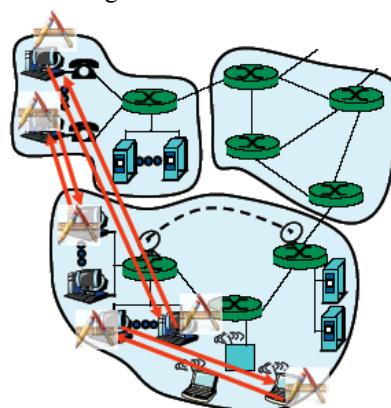
- Lanzado más tarde por el usuario...
- Inicia la comunicación con un servidor...
- No con clientes
- Termina cuando el usuario deja de usarlo
- Puede no tener siempre la misma dirección

Ejemplos: casi todos los servicios clásicos Web, mail, FTP, News, IRC, Streaming...

## 7.2.2 Arquitectura Peer-to-Peer

- No hay un servidor siempre disponible
- Hosts extremos cualesquier se comunican (peers)...
- Pueden no estar siempre conectados...
- Los peers pueden cambiar de dirección
- El mismo proceso puede ser cliente o servidor

Ejemplo: Gnutella (Escalable Difícil de controlar)



## 7.2.3 Híbrido de cliente-servidor y P2P

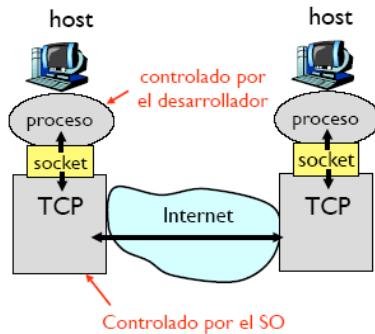
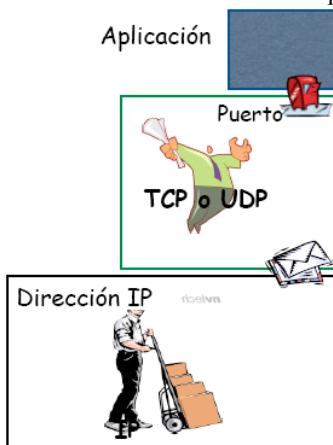
Napster, eMule y similares...

- Transferencia de ficheros P2P
- Búsqueda de ficheros centralizada:

- Peers registran el contenido ofrecido en un servidor central
- Peers preguntan al mismo servidor para buscar ficheros
- Mensajería Instantánea (Instant messaging=IM)
  - Conversación entre dos usuarios es P2P
  - Detección de presencia y localización centralizada:
    - Los usuarios registran su dirección en un servidor central cuando se conectan a la red
    - Contactan con el servidor central para encontrar la dirección actual de sus contactos

### 7.3 Sockets

- Los procesos envían y reciben mensajes a través de un socket
- Delega en el nivel de transporte para que haga llegar los mensajes al otro socket
- Acceso a través de un API
- Puede escoger el protocolo de transporte
- Puede configurar algunos parámetros del mismo
- No controla cómo se comporta



Identificando al proceso

- El emisor de un mensaje debe identificar al host receptor
- Un host (interfaz) tiene una dirección IP única (32 bits)
- Muchos procesos en el mismo host
- Debe identificar al proceso receptor que corre en ese host
- Número de puerto diferente asociado a cada proceso

Ejemplos:

Servidor Web: puerto TCP 80

Servidor e-mail: puerto TCP 25

### 7.4 Servicios que necesitan las aplicaciones

- Retardo** Algunas apps requieren bajo retardo (ej. juegos en red)
- Ancho de banda** Algunas aplicaciones requieren un mínimo de ancho de banda (ej. audioconf). Otras (elásticas) funcionan con cualquier cantidad pero pueden sacar provecho a todo el disponible
- Pérdidas** Algunas aplicaciones soportan pérdidas (ej. audio). Otras requieren 100% de fiabilidad (ej. Transferencia de ficheros)

#### Requisitos de aplicaciones comunes

Aplicación	Pérdidas	Ancho de banda	Retardo
Transf. ficheros	ninguna	elástico	no
e-mail	ninguna	elástico	no
Web	ninguna	elástico	no
audio/vídeo en RT	soporta	audio: 5kbps-1Mbps video: 10kbps-5Mbps	sí, 100's msec
audio/vídeo diferido	soporta	idem	sí, unos segs
juegos interactivos	soporta	desde unos kbps	sí, 100's msec
IM	ninguna	elástico	sí y no

### 7.5 Servicios ofrecidos por protocolos de transporte en Internet

#### TCP:

- orientado a conexión: establecimiento requerido entre ambos procesos
- transporte fiable: sin pérdidas
- control de flujo: el emisor no saturará al receptor

- control de congestión: limita el envío cuando la red está sobrecargada
  - no ofrece: límite al retardo, mínimo ancho de banda garantizado

UDP:

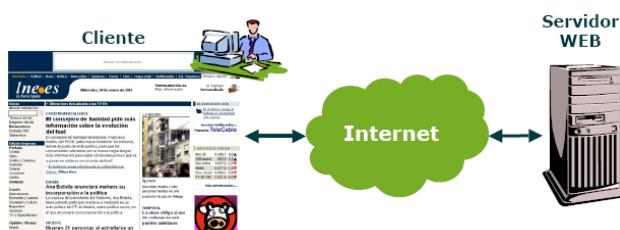
- Transferencia de datos no fiable entre los dos procesos
  - No ofrece: conexión, fiabilidad, control de flujo, control de congestión, límite al retardo ni ancho de banda garantizado

Aplicaciones de Internet: protocolos de aplicación y transporte

Aplicación	Protocolo de nivel de aplicación	Protocolo de nivel de transporte
e-mail	SMTP [RFC 2821]	TCP
acceso remoto	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
transferencia de fichero	FTP [RFC 959]	TCP
streaming	Suele ser propietario (ej. RealNetworks)	TCP o UDP
Telefonía en Internet	Suele ser propietario (ej.. Dialpad)	típicamente UDP

## 7.6 El protocolo HTTP

Protocolo para la transferencia de ficheros de hipertexto. Base para los servicios Web

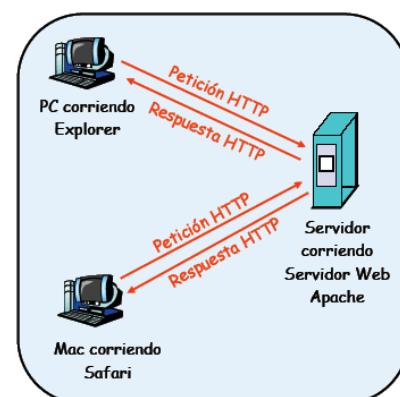


Términos

- Una Página Web está compuesta por objetos
  - Un objeto puede ser un fichero HTML, una imagen JPEG, un applet JAVA, un fichero de sonido...
  - La página Web está compuesta por un fichero HTML base que hace referencia a otros objetos
  - Se hace referencia a cada objeto mediante un URL

Ejemplo de URL:

<http://www.tlm.unavarra.es/~mikel/index.html>



## HTTP: HyperText Transfer Protocol

- Protocolo de nivel de aplicación de la Web
  - Modelo cliente/servidor
  - *cliente*: browser (navegador) que solicita, recibe y muestra objetos de la Web
  - *servidor*: el servidor Web envía objetos en respuesta a peticiones
  - HTTP 1.0: RFC 1945
  - HTTP 1.1: RFC 2668

Usa TCP;

- El cliente inicia una conexión TCP al servidor, puerto 80
  - El servidor acepta la conexión TCP del cliente

- Cada uno tiene un socket conectado con el otro
- Se intercambian mensajes http entre el navegador y el servidor Web
- Se cierra la conexión TCP

HTTP es “sin estado”

- El servidor no mantiene ninguna información de peticiones anteriores del cliente

### Empleo de las conexiones

HTTP no persistente

- En cada conexión TCP se envía como máximo un objeto
- HTTP/1.0

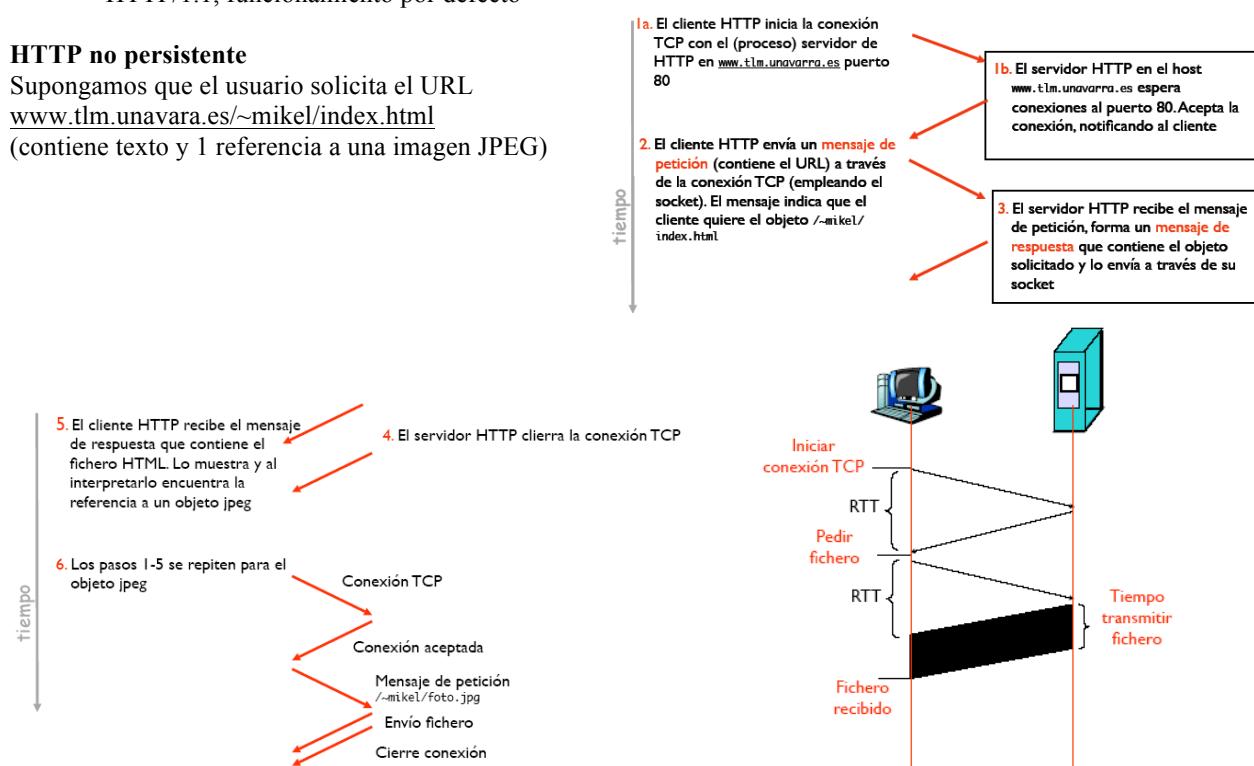
HTTP persistente

- En la misma conexión TCP se pueden enviar varios objetos entre el servidor y el cliente
- HTTP/1.1, funcionamiento por defecto

### HTTP no persistente

Supongamos que el usuario solicita el URL  
[www.tlm.unavarra.es/~mikel/index.html](http://www.tlm.unavarra.es/~mikel/index.html)

(contiene texto y 1 referencia a una imagen JPEG)



### Modelo del tiempo de respuesta

Definición de RTT:

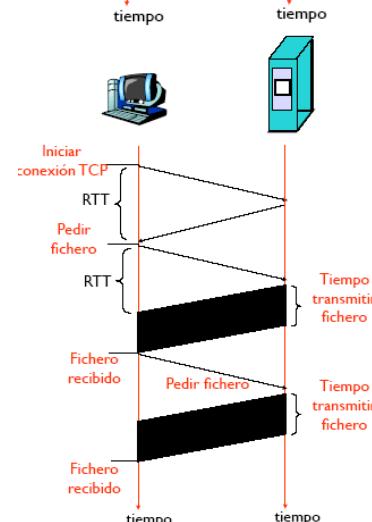
*Round Trip Time* tiempo para que un paquete pequeño viaje de cliente a servidor y vuelta

Tiempo de respuesta:

- Un RTT para iniciar la conexión
  - Un RTT para la petición HTTP y el comienzo de la respuesta
  - Tiempo de transmisión del fichero
- total = 2RTT+tiempo transmisión

Con HTTP no persistente:

- Requiere 2 RTTs por objeto
- OS debe reservar recursos para cada conexión TCP
- Pero el navegador suele abrir varias conexiones TCP en paralelo



HTTP persistente:

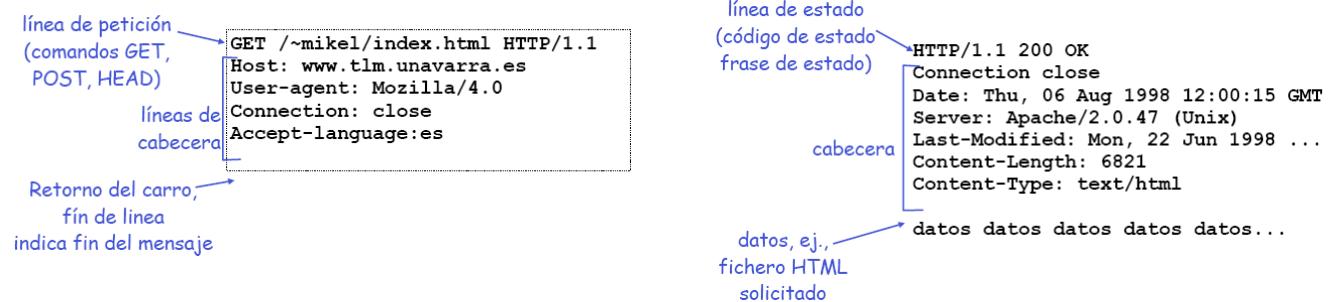
- El servidor deja la conexión abierta tras enviar la respuesta
- Los siguientes mensajes http entre cliente y servidor van por la misma conexión

**HTTP request message**

- Dos tipos de mensajes messages: *request, response*
- » Mensaje HTTP request :

> ASCII (formato legible por humanos)

**HTTP response message**



**Probando HTTP desde el cliente**

► Haga telnet a su servidor Web favorito:

```
$ telnet www.tlm.unavarra.es 80
```

Abre una conexión TCP al puerto 80 (puerto por defecto del servidor HTTP) de www.tlm.unavarra.es  
Lo que se escriba se envía por la conexión TCP

► Escriba una petición GET de HTTP:

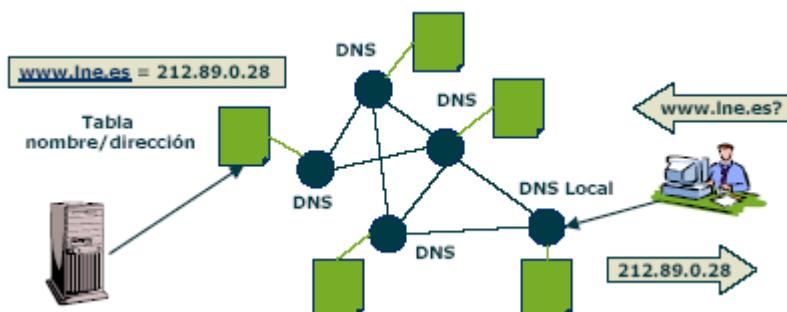
```
GET /~mikel/ HTTP/1.1
Host: www.tlm.unavarra.es
```

Escribiendo esto (y retorno del carro dos veces) se envía un petición HTTP 1.1 mínima pero completa al servidor

► Vea el mensaje de respuesta del servidor

## 7.7 El protocolo DNS--Domain Name System

- Protocolo para la resolución de nombres
- A partir del nombre lógico de una máquina resuelve su dirección IP



**El problema de los nombres**

- Las direcciones IP, que identifican a los interfaces de los hosts, son números de 32 bits
- Sencillas de manejar para las máquinas, complicado para los humanos
- Más sencillo memorizar nombres textuales

- Hace falta “traducir” el nombre textual en la dirección numérica para que se pueda realizar la comunicación. Esto se llama “resolver el nombre”
- La traducción se realiza mediante el Sistema de Nombres de Dominio o DNS (Domain Name System)

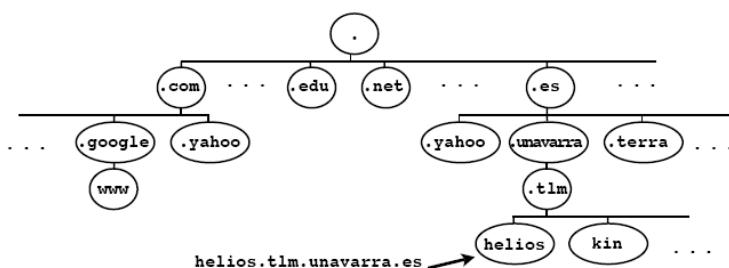
### **Jerarquía de nombres**

Los nombres están formados por segmentos alfanuméricos separados por puntos (no distingue mayúsculas)

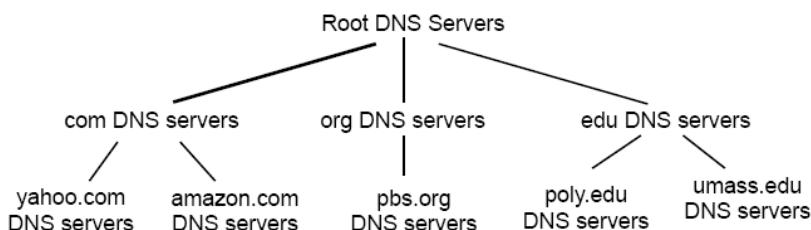
helios.tlm.unavarra.es

www.google.com

### **Estructura jerárquica**



### **Base de datos jerárquica distribuida**



El cliente busca la IP de www.amazon.com, 1<sup>a</sup> aproximación:

- El cliente pregunta a un servidor Root para encontrar el servidor de DNS del dominio com
- El cliente pregunta al servidor del dominio *com* para obtener el servidor del dominio amazon.com
- El cliente pregunta al servidor DNS del dominio amazon.com para obtener la IP de www.amazon.com

### **Implementación**

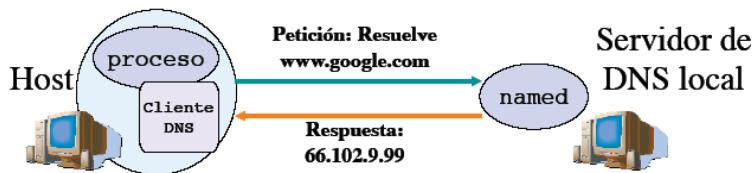
- El servidor es un programa específico pero el cliente es generalmente solo unas funciones en una librería (*resolver*)...
- La aplicación cliente de DNS es la propia aplicación del usuario...
- El software típico que lo implementa es BIND (Berkeley Internet Name Domain) (el programa servidor se llama *named*)...



### **Funcionamiento**

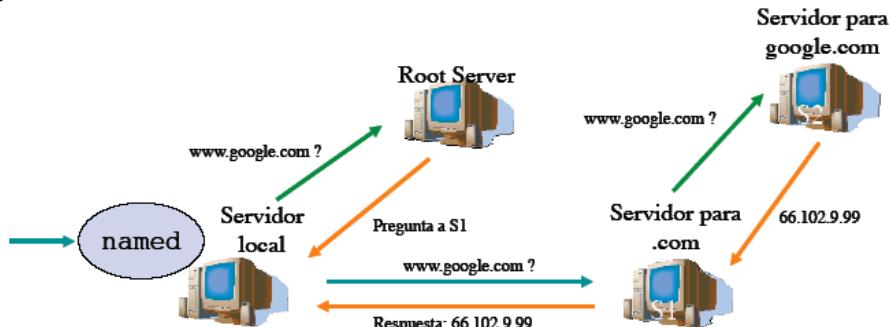
- Mensajes sobre UDP puerto 53
- Cada ISP posee un servidor de nombres local...
- Los hosts tienen configurado a su servidor local

- Cuando un host desea resolver un nombre hace la petición a su servidor local el cual le devuelve la respuesta...



### ¿Cómo conoce la respuesta el servidor local?

- Si es el servidor autoritario (authoritative server) para el dominio en el que está esa máquina él tiene la porción de la base de datos distribuida en la que está el mapeo
  - Si no lo es preguntará a un Root Server
- El servidor local pregunta a un Root Server...
- Éste le devuelve la dirección de un servidor intermedio (petición iterativa)...
- El Servidor local hace una petición recursiva a ese servidor...
- Ese servidor continuará haciendo la petición (recursiva) hasta que llegue un servidor autoritario . . .
  - Todas las peticiones son recursivas menos la petición al Root Server para reducir la carga sobre los Root



### DNS: Root name servers

- 13 en el mundo
- En el fichero de configuración de cada servidor de DNS



- En vez de direcciones binarias de red, los programas normalmente usan strings de ASCII, tal como *ing.puc.cl*. Pero la red entiende solamente las direcciones binarias, así que se necesita una manera para traducir entre las dos.

- Originalmente en la ARPANET se usaba un archivo *hosts.txt* con todos los hosts y sus direcciones IP. Cada noche todos los hosts lo bajaban.
- Claramente este enfoque no puede escalar. Se necesita un sistema que evite conflictos pero no requiera la administración central. Ahora se usa DNS (sistema de nombres de dominios) para manejar los nombres. Usa una base de datos distribuida y una esquema jerárquico de administración de nombres.
- En principio, DNS funciona en la manera siguiente: Para traducir un nombre a una dirección un programa llama a un procedimiento de resolución. Esto manda un paquete de UDP al servidor local de DNS, que busca la dirección de IP usando el nombre. La vuelve al procedimiento de resolución, que la vuelva al programa.

### Espacio de nombres de DNS

- Como en el correo, se parte la Internet en algunos cientos de dominios de primer nivel (*edu, com, cl, de, be*, etc.). Se parte cada dominio en subdominios, que se parten, etc. El sistema es un árbol.
- Hay dos tipos de dominios de primer nivel: genérico y países. Los dominios genéricos son *com* (comercial), *edu* (educación), *gov* (gobierno de EE.UU.), *mil* (fuerzas armadas de EE.UU.), *net* (proveedores de red), y *org* (organizaciones sin fines comerciales).
- Se dividen las partes de un nombre por puntos: *ing.puc.cl*. Los nombres pueden ser absolutos o relativos (por ejemplo, *ing*). Los últimos se tienen que interpretar en algún contexto.
- Los nombres son insensibles a caja. Los componentes pueden tener hasta 63 caracteres, y los nombres completos no pueden exceder 255 caracteres.
- Cada dominio controla la asignación de los dominios bajo él. Un dominio puede crear nuevos subdominios sin la autorización de dominios arriba.
- Los nombres son basados en las organizaciones, no en las redes físicas.

Nombre de Dominio	Significado
<b>COM</b>	<b>Organizaciones comerciales</b>
<b>EDU</b>	<b>Instituciones educativas</b>
<b>GOV</b>	<b>Instituciones gubernamentales</b>
<b>MIL</b>	<b>Grupos militares</b>
<b>NET</b>	<b>Centros de soporte de red</b>
<b>ORG</b>	<b>Organizaciones</b>
<b>ARPA</b>	<b>Dominio de ARPANET (obsoleto)</b>
<b>INT</b>	<b>Organizaciones Internacionales</b>
<b>TV</b>	<b>Televisores - Nuevo</b>

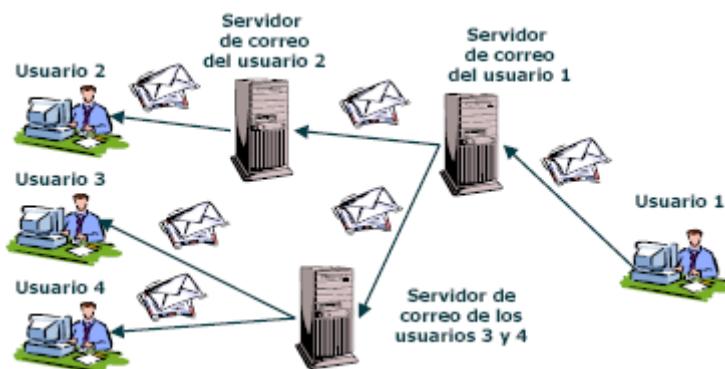
### Registros de recurso

- Cada dominio puede tener un conjunto asociado de *registros de recurso*. El procedimiento de resolución realmente recibe registros de recurso del servidor de DNS.
- Un registro de recurso tiene cinco partes: nombre de dominio, tiempo de vida, tipo, clase, y valor.
- El nombre de dominio es la clave del registro. Normalmente hay muchos registros por dominio y la base de datos guarda información sobre dominios múltiples. La orden de registros no importa.
- El tiempo de vida (en segundos) indica la estabilidad de la información en el registro. Se usa para controlar la expiración de copias de la información.
- Tipos y valores:
  - SOA (Start of Authority). Los parámetros para esta zona.
  - A (Address). La dirección de IP por el host.
  - MX (Mail Exchange). La prioridad y el nombre del dominio que puede aceptar correo electrónico dirigido a esto.

- NS (Name Server). El nombre de un servidor de DNS para este dominio.
- CNAME (Canonical Name). Un alias para un nombre (por ejemplo, `www.ing.puc.cl`).
- PTR (Pointer). Otro tipo de alias.
- HINFO (Host Info). Una descripción de la máquina y su sistema operativo.
- TXT (Text). Otra información opcional.
- La clase es siempre IN para información de Internet.
- Ejemplo: En `nslookup` prueba "`ls -d ing.puc.cl`".

## 7.8 El protocolo SMTP

Protocolo para la gestión de correo electrónico



Tres elementos principales:

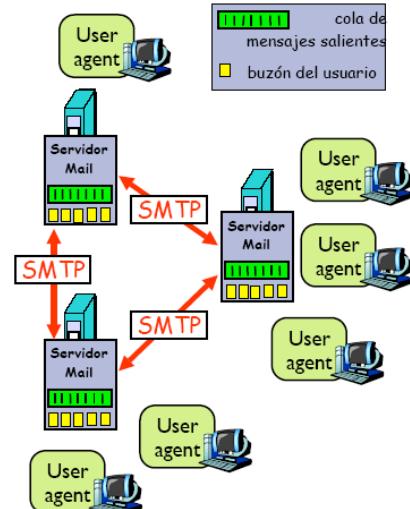
- Agentes de usuario (*user agents*)
- mail servers
- Simple Mail Transfer Protocol:

### User Agent

- alias “programa de correo”
- Componer, editar, leer mensajes de correo
- ej., Eudora, Outlook, elm, Netscape Messenger
- Mensajes salientes y entrantes en el servidor

### Servidores de Mail:

- mailbox contiene los mensajes entrantes para el usuario
- cola de mensajes salientes (a enviar)
- Protocolo SMTP entre servidores de correo para enviar mensajes
  - > cliente: el servidor de correo que envía
  - > “servidor”: el servidor de correo que recibe



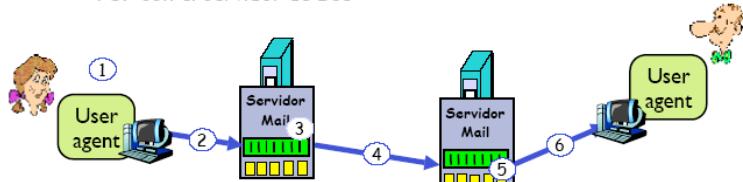
### Electronic Mail: SMTP [RFC 2821]

- Emplea TCP para entregar de forma fiable los mensajes entre el cliente y el servidor
- Puerto 25
- Transferencia directa: del servidor del emisor al servidor del receptor
- Tres fases en la transferencia
  - > *handshaking* (el saludo)
  - > transferencia de mensajes
  - > cierre
- Interacción mediante comandos y respuestas
- comandos: texto ASCII
- respuestas: código de estado y frase de estado

- Los mensajes deben estar en ASCII de 7 bits

**Ejemplo: Alicia envía mensaje a Bob**

- 1) Alicia emplea un UA para crear el mensaje para bob@someschool.edu
- 2) El programa envía el mensaje a su servidor de correo y lo coloca en una cola de mensajes
- 3) El Servidor de Mail, como cliente, abre una conexión TCP con el Servidor de Bob
- 4) Envía el mensaje de Alicia empleando SMTP sobre esa conexión TCP
- 5) El servidor de mail de Bob coloca el mensaje en su buzón
- 6) Bob lanza su UA para leer el mensaje



**Ejemplo de SMTP**

S: 220 hamburger.edu  
 C: HELO crepes.fr  
 S: 250 Hello crepes.fr, pleased to meet you  
 C: MAIL FROM: <alice@crepes.fr>  
 S: 250 alice@crepes.fr... Sender ok  
 C: RCPT TO: <bob@hamburger.edu>  
 S: 250 bob@hamburger.edu ... Recipient ok  
 C: DATA  
 S: 354 Enter mail, end with "." on a line by itself  
 C: Do you like ketchup?  
 C: How about pickles?  
 C: .  
 S: 250 Message accepted for delivery  
 C: QUIT  
 S: 221 hamburger.edu closing connection

[S]ervidor [C]lient

**Probando SMTP**

Escriba:

```
$ telnet servername 25
> Pruebe los comandos HELO, MAIL FROM, RCPT TO, DATA, QUIT
> Con esos comandos puede enviar un email sin emplear un programa de email
```

```
$ telnet si.unavarra.es 25
Trying 130.206.166.108...
Connected to si.unavarra.es.
Escape character is '^].
220 unavarra.es ESMTP Sendmail 8.9.3/8.9.1 (IRIS 3.0); Sun, 7 Aug 2005
14:06:28 +0200 (MET DST)
HELO mikel.tlm.unavarra.es
250 unavarra.es Hello s169m177.unavarra.es [130.206.169.177], pleased to
meet you
MAIL FROM: mikel.izal@unavarra.es
250 mikel.izal@unavarra.es... Sender ok
RCPT TO: mikel.izal@gmail.com
250 mikel.izal@gmail.com... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
Hola esto es para probar que va
.
250 OAA13441 Message accepted for delivery
QUIT
221 unavarra.es closing connection
Connection closed by foreign host.
```

c:\>telnet mail.khainata.net 25

```
Símbolo del sistema
220 khainata.net <IMail 8.20 4143-13> NT-ESMTP Server X1
HELO
250 hello khainata.net
MAIL FROM:Jorge.Orellana@redes.net
250 ok
RCPT TO:luriarte@arcobol.com
250 ok its for <luriarte@arcobol.com>
DATA
354 ok, send it; end with <CRLF>.<CRLF>
Hola, es una prueba de envío de mails por SMTP
.
250 Message queued
QUIT
221 Goodbye

Se ha perdido la conexión con el host.

C:\Documents and Settings\jorellana.IBCH>
```

SMTP emplea conexiones persistentes

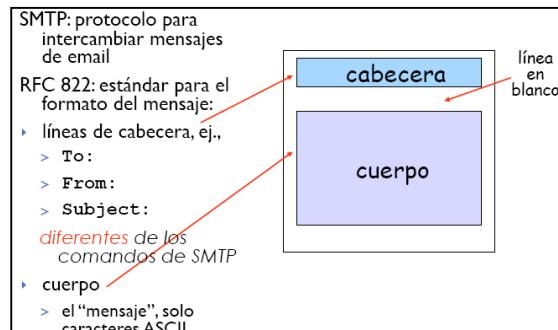
SMTP requiere que el mensaje (cabecera y contenido) esté en ASCII de 7 bits

El servidor de SMTP reconoce el fin del mensaje al ver una linea que sólo contenga .

CRLF.CRLF “\r\n.\r\n”

#### Comparación con HTTP:

- HTTP: pull
  - SMTP: push
  - Ambos usan comandos y respuestas en ASCII
- Formato del mensaje de email



#### Formato del mensaje: multimedia extensions

- MIME: MultiMedia Mail Extension, RFC 2045, 2056
- Permite mandar contenido que no sea texto ASCII
- Líneas adicionales en la cabecera del mensaje para declarar el tipo del contenido

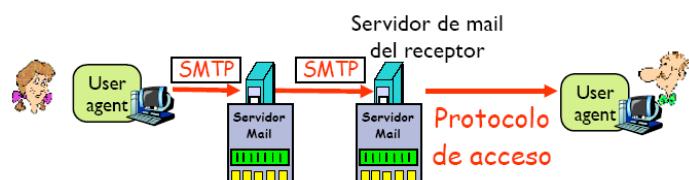
versión de MIME  
método empleado para codificar los datos  
tipo, subtipo, parametros de los datos multimedia  
.....  
datos codificados

```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

base64 encoded data .....
.....base64 encoded data
```

#### Protocolos de acceso al Mail

- SMTP: entrega/almacena en el servidor del receptor
- Protocolo de acceso al Mail: obtención de



mensajes del servidor

- > POP: Post Office Protocol [RFC 1939]
  - + Autorización (agente <-->servidor) y descarga
- > IMAP: Internet Mail Access Protocol [RFC 1730]
  - + más funcionalidades (más complejo)
  - + manipulación de mensajes almacenados en el servidor
- > HTTP: Hotmail , Yahoo! Mail, etc.

- En la Internet se usa SMTP (Simple Mail Transfer Protocol), un protocolo sencillo, sobre TCP a puerta 25 del destino.
- Para mandar el correo electrónico a otras redes se usan gateways de email.
- Para la entrega final, si el usuario no trabaja en la máquina de destino, se pueden usar
  - POP3 (Post Office Protocol). Baja mensajes.
  - IMAP (Interactive Mail Access Protocol). Es más sofisticado. El usuario puede dejar los mensajes en el servidor, especificar mensajes usando sus características (por ejemplo, emisor), etc.
  - DSMP (Distributed Mail System Protocol). No asume que el correo está en un solo servidor.
- Frecuentemente los usuarios pueden instalar *filtros* para su correo. Chequean el correo usando reglas varias.

### Protocolo POP3

#### Autorización

- Comandos del cliente:
  - > user: declara el nombre de usuario
  - > pass: clave
- Respuestas del servidor:
  - > +OK
  - > -ERR

#### Fase de transacción, cliente:

- list: lista números de mensajes
- retr: descarga mensaje por número
- dele: borrar
- quit

```
S: +OK POP3 server ready
C: user bob
S: +OK
C: pass hungry
S: +OK user successfully logged on
```

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <contenido mensaje 1>
S: .
C: dele 1
C: retr 2
S: <contenido mensaje 2>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

#### Más sobre POP3

- El ejemplo anterior era “descargar y borrar”
- Bob no puede volver a leer los mensajes si cambia de cliente
- “Descargar y mantener”: copia el mensaje pero no lo borra. Permite descargarlos en otro cliente
- POP3 es sin estado entre sesiones

#### IMAP

- Mantiene todos los mensajes en un lugar: el servidor
- Permite al usuario organizar los mensajes en carpetas
- IMAP mantiene el estado entre sesiones:
- Nombres de carpetas y relación entre ID de mensaje y carpeta en la que está

### 7.9. El protocolo Telnet (Login remoto)

- Protocolo para el trabajo mediante terminal remota
- Permite trabajar desde una localización remota con la consola de un computador



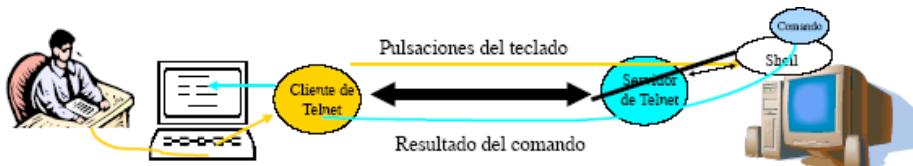
Permite el uso interactivo de otra computadora (UNIX) de forma remota como desde un terminal

#### Funcionamiento:

- El usuario ejecuta un cliente de Telnet especificando una máquina servidor...

- Se crea una conexión TCP con el servidor (puerto del servidor de Telnet=23)...
- El servidor crea un proceso Shell que queda conectado a la conexión TCP...
- Las pulsaciones del teclado del usuario se transmiten por la conexión a la shell...
- La shell ejecuta los comandos que escribe el usuario...
- El resultado que el comando mandaría a la pantalla vuelve por la conexión TCP y sale en la pantalla del cliente...

Otros servicios similares: rlogin, rsh, ssh



### Ejemplo

```

$ telnet tlm22.net.tlm.unavarra.es
Trying 10.1.1.22...
Connected to tlm22.net.tlm.unavarra.es.
Escape character is '^]'.
Fedora Core release 2 (Tettnang)
Kernel 2.6.5-1.358 on an i686
login: mikel
Password:
Last login: Fri May  6 20:33:54 from bender.net.tlm.unavarra.es
[mikel@tlm22 mikel]$ ls -l
total 106132
drwxr-xr-x  2 mikel staff      4096 Mar  5 00:17 a
drwxr-xr-x  15 mikel staff      4096 Oct 25 2004 apache
-rw-r--r--   1 mikel staff  41685513 Nov 12 2004 borrar
drwxr-xr-x   4 mikel staff      4096 Dec  2 2004 demo-italia
-rw-r--r--   1 mikel staff  116627 Dec  2 2004 demo-italia.tar.gz
drwxr-xr-x   3 mikel staff      4096 Jun 27 09:43 Desktop
drwx-----   4 mikel staff      4096 May 27 15:38 evolution
drwxr-xr-x   2 mikel staff      4096 Oct  6 2004 prueba_c
-rw-r--r--   1 mikel staff  209211 May  6 10:24 prueba.ps
drwxr-xr-x   2 mikel staff      4096 May 11 21:34 py23
[mikel@tlm22 mikel]$
    
```

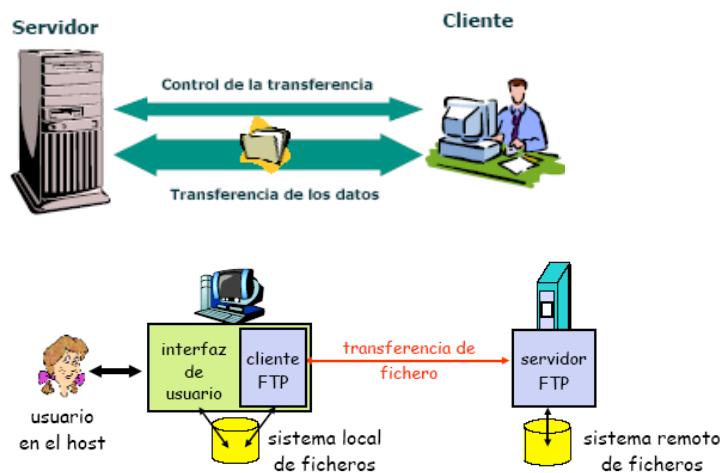
```

SunOS 5.8

login: optimix
Password:
Login incorrect
login: optimix
Password:
Login incorrect
login: root
Password:
Last login: Tue Dec 13 19:26:12 from 10.0.0.2
Sun Microsystems Inc. SunOS 5.8      Generic Patch  February 2004
Sourcing //PROFILE-EIS.....
root@sirio.pdar.net # ls
bin          export      mnt      platform     u01      var
cdrom        home       net      proc         u02      vol
dev          kernel     nsmail   sbin         u03      xfn
devices      lib        opt      tmp          u04
etc          lost+found oracle   TT_DB       usr
root@sirio.pdar.net #
    
```

### 7.10. El Protocolo FTP: File Transfer Protocol

Protocolo utilizado para la transferencia de archivos entre máquinas remotas



- Transferencia de fichero hacia/desde host remoto
- modelo cliente-servidor
  - > cliente: extremo que inicia la transferencia (bien sea desde o hacia el extremo remoto)
  - > servidor: host remoto
- FTP: RFC 959
- Servidor FTP: TCP puerto 21

### FTP: conexiones de datos y control

- ▶ El cliente FTP contacta con el servidor en el puerto 21 empleando TCP
  - ▶ El cliente se autentifica a través de esta conexión de control
  - ▶ El cliente puede explorar los directorios remotos enviando comandos por la conexión de control
  - ▶ Cuando el servidor recibe un comando para una transferencia de fichero abre una conexión TCP con el cliente
  - ▶ Tras transferir el fichero cierra esa conexión de datos
- ▶ El servidor abre una segunda conexión TCP para transferir el fichero
  - ▶ Conexión de control “out of band”
  - ▶ El servidor FTP mantiene el “estado”: directorio actual, autenticación

### Comandos y respuestas FTP

<u>Comandos de ejemplo:</u>	<u>Códigos de respuesta:</u>
Enviados como texto ASCII por el canal de control	Código de estado y frase (como en HTTP)
<ul style="list-style-type: none"> <li>▶ <b>USER username</b></li> <li>▶ <b>PASS password</b></li> <li>▶ <b>LIST</b> devuelve una lista de los ficheros en el directorio actual</li> <li>▶ <b>RETR filename</b> Obtiene el fichero</li> <li>▶ <b>STOR filename</b> Almacena el fichero en el host remoto</li> </ul>	<ul style="list-style-type: none"> <li>▶ <b>331 Username OK, password required</b></li> <li>▶ <b>125 data connection already open; transfer starting</b></li> <li>▶ <b>425 Can't open data connection</b></li> <li>▶ <b>452 Error writing file</b></li> </ul>

## Ejemplo de FTP

```
$ ftp tlm22.net.tlm.unavarra.es
Connected to tlm22.net.tlm.unavarra.es (10.1.1.22).
220 (vsFTPd 1.2.1)
Name (tlm22.net.tlm.unavarra.es:mikel): mikel
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (10,1,1,22,215,255)
150 Here comes the directory listing.
drwxr-xr-x 3 1003 1000 4096 Jun 27 07:43 Desktop
-rw-r--r-- 1 1003 1000 209211 May 06 08:24 prueba.ps
drwxr-xr-x 2 1003 1000 4096 Oct 06 2004 prueba_c
drwxr-xr-x 2 1003 1000 4096 May 11 19:34 py23
-rw-r--r-- 1 1003 1000 20083157 May 11 19:34 py23.tgz
226 Directory send OK.
ftp> get py23.tgz
local: py23.tgz remote: py23.tgz
227 Entering Passive Mode (10,1,1,22,95,165)
150 Opening BINARY mode data connection for py23.tgz (20083157
bytes).
226 File send OK.
20083157 bytes received in 1.86 secs (1.1e+04 Kbytes/sec)
ftp>
```

```
ex C:\WINDOWS\system32\cmd.exe - ftp sirio.pdar.net
C:\>Documents and Settings\Administrador>ftp sirio.pdar.net
Conectado a sirio.pdar.net.
220 sirio.pdar.net FTP server <SunOS 5.8> ready.
Usuario <sirio.pdar.net:<none>>: oracle
331 Password required for oracle.
Contraseña:
230 User oracle logged in.
ftp> ls
200 PORT command successful.
150 ASCII data connection for /bin/ls <10.0.0.2,2914> <0 bytes>.
afiedt.buf
alterarbd.txt
archivos.txt
exp_full.sh.txt
expdat.dmp
nsmail
optimix1795.dmp
oracle8i.cron.txt
tuning_stats.sql
tuning_stats.txt
226 ASCII Transfer complete.
ftp: 150 bytes recibidos en 0,00 segundos 150000,00 a KB/s.
ftp> get archivos.txt
200 PORT command successful.
150 ASCII data connection for archivos.txt <10.0.0.2,2824> <66339 bytes>.
226 ASCII Transfer complete.
ftp: 66426 bytes recibidos en 0,20 segundos 327,22 a KB/s.
ftp>
```

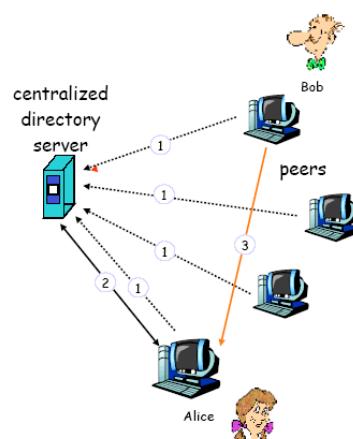
## 7.11. P2P: directorio centralizado

Diseño original de "Napster"

- 1) Cuando un peer se conecta, informa al servidor central:

  - > Dirección IP
  - > contenido

- 2) Alice hace una búsqueda de "Hey Jude"
- 3) Alice pide el fichero a Bob



## Ventajas e inconvenientes

### Ventajas

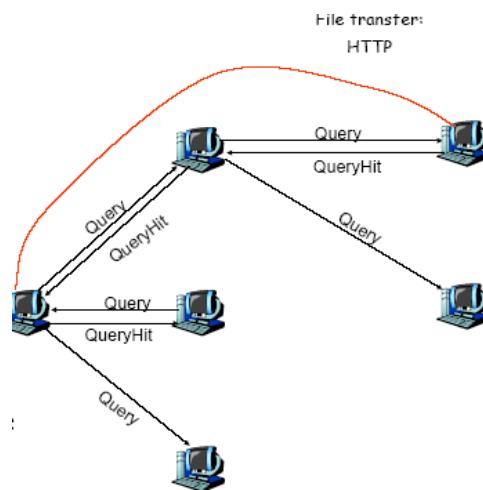
Todos los peers son servidores  
Altamente escalable

### Inconvenientes

Un punto de fallo central  
Impone un límite de prestaciones  
Infracción de copyrights!

### Gnutella

Completamente distribuido  
Dominio público  
Overlay network  
> Grafo  
> Cada conexión un enlace  
Petición de búsqueda enviada sobre las conexiones TCP  
peers reenvían la petición  
Respuesta enviada por el camino inverso  
Escalabilidad: limitar el alcance de la inundación



## 7.12. USENET

- USENET es independiente de la Internet. Tiene miles de *newsgroups* (grupos de noticias). La cantidad de mensajes excede 500 MB por día. Con 56 kbps se necesitan 20 horas para bajar todos.
- Para obtener las noticias, un sitio necesita un *newsfeed* (fuente de noticias). Típicamente los sitios no reciben todos los grupos.
- USENET usa el protocolo NNTP (Network News Transfer Protocol). Está en la puerta 119. Puede *empujar* los mensajes o *tirarlos*.
- **Comandos de NNTP:**
  - LIST. Darme la lista de grupos.
  - NEWGROUPS fecha. Darme la lista de grupos nuevos.
  - GROUP grupo. Darme la lista de los mensajes en el grupo.
  - NEWNEWS grupos fecha. Darme la lista de los mensajes nuevos en los grupos.
  - ARTICLE id. Darme el mensaje específico.
  - POST. Tengo un mensaje.
  - IHAVE id. Tengo el mensaje específico; ¿lo quieres?
  - QUIT. Terminar la sesión.
- Un problema con el protocolo es que es parar-y-esperar.