

<b>1- INTRODUCCIÓN .....</b>	<b>4</b>
1- PROTOCOLOS DE COMUNICACIONES: MODELO OSI Y TCP/IP .....	5
<i>Protocolos de comunicaciones .....</i>	5
<i>Arquitectura de protocolos .....</i>	7
<i>Unidades de datos de los protocolos .....</i>	10
<i>Funcionamiento de una arquitectura de protocolos.....</i>	13
<i>El modelo OSI .....</i>	15
<i>El modelo TCP/IP .....</i>	17
<i>Tecnologías de red .....</i>	21
2- HISTORIA E IMPLEMENTACIÓN DE TCP/IP .....	22
<i>Nacimiento de TCP/IP: ARPANET .....</i>	22
<i>Características de TCP/IP .....</i>	24
<i>Internet .....</i>	25
<i>Organismos reguladores y normas.....</i>	28
<i>OSI y TCP/IP.....</i>	30
<i>Implementación de TCP/IP .....</i>	31
<i>Encapsulamiento de TCP/IP .....</i>	32
<i>Demultiplexación en TCP/IP.....</i>	33

<b>2- DIRECCIONES INTERNET .....</b>	<b>34</b>
1- DIRECCIONES IP .....	35
<i>Direcciones IP .....</i>	35
<i>Formato de direcciones IP .....</i>	38
<i>Clases de direcciones IP .....</i>	39
<i>Direcciones sin conexión a Internet .....</i>	41
<i>Redes y Subredes de TCP/IP .....</i>	42
<i>Máscaras de subred.....</i>	44
<i>Direcciones especiales reservadas .....</i>	46
2- PROTOCOLO ARP .....	48
<i>Protocolo ARP.....</i>	48
<i>Formato de trama ARP .....</i>	51
<i>Tabla ARP .....</i>	53
<i>Mecanismo ARP Proxy.....</i>	54
3- DIRECCIONES DE MULTIENVÍO .....	56
<i>Direcciones de multienvío .....</i>	56
<i>Grupos de multienvío .....</i>	57
<i>Extensión de IP para manejar el multienvío .....</i>	60
<i>Traducción de direcciones de multienvío IP a direcciones Ethernet .....</i>	61
<i>Protocolo de gestión de grupos de Internet (IGMP) (I) .....</i>	64
<i>Protocolo de gestión de grupos de Internet (IGMP) (II).....</i>	66
<i>Protocolo de gestión de grupos de Internet (IGMP) (III).....</i>	67
<b>3- PROTOCOLO INTERNET (IP) .....</b>	<b>68</b>
1- PROTOCOLO IP .....	69
<i>Características de IP .....</i>	69
<i>Formato de trama.....</i>	71
<i>Campos de trama destino, origen y protocolo.....</i>	72
<i>Campos de trama versión, longitud de cabecera y longitud total del datagrama.....</i>	73
<i>Campo de precedencia y tipo de servicio .....</i>	74
<i>Tiempo de vida y checksum de la cabecera.....</i>	75
<i>Identificación, banderas y desplazamiento del fragmento .....</i>	77
<i>Mecanismo de fragmentación.....</i>	78
<i>Opciones del Datagrama IP.....</i>	81
<i>Campo opciones .....</i>	82
<b>4- PROTOCOLO DE MENSAJES DE CONTROL DE INTERNET (ICMP).....</b>	<b>85</b>
1- PROTOCOLO ICMP .....	86
<i>Protocolo ICMP .....</i>	86
<i>Mensajes de error de ICMP .....</i>	88
<i>Ejemplo de notificación de errores con ICMP .....</i>	89
<i>Mensajes de error de ICMP .....</i>	90
<i>Tratamiento de los mensajes de error ICMP entrantes.....</i>	92
<i>Formato del mensaje de error.....</i>	93
<i>Mensajes de petición ICMP.....</i>	94
<i>Mensajes ICMP de petición y respuesta de eco .....</i>	95
<b>5- PROTOCOLO DE DATAGRAMAS DE USUARIO (UDP).....</b>	<b>97</b>
1- PROTOCOLO UDP .....	98
<i>Protocolo UDP.....</i>	98
<i>Puertos de las aplicaciones: encapsulación y demultiplexación.....</i>	101
<i>Mecanismos del protocolo UDP.....</i>	103
<b>6- SERVICIO DE TRANSPORTE DE FLUJO CONFiable (TCP).....</b>	<b>104</b>
1- PROTOCOLO TCP .....	105
<i>Protocolo TCP.....</i>	105
<i>Formato del segmento .....</i>	107

<i>Puertos de aplicación</i> .....	108
<i>Identificadores de conexión</i> .....	110
<i>Mecanismos de fiabilidad de TCP</i> .....	111
<i>Reconocimientos</i> .....	112
<i>RTT</i> .....	113
<i>Campos</i> .....	114
<i>Sliding Window</i> .....	115
<i>Suma de control</i> .....	121
<i>Datos urgentes</i> .....	122
<i>Mecanismo push</i> .....	124
<i>Banderas (Flags) de control</i> .....	126
<i>Opción de tamaño máximo de segmento</i> .....	127
<i>Establecimiento de la conexión</i> .....	128
<i>Liberación de conexión</i> .....	131
<b>7- PROTOCOLOS DE APLICACIÓN .....</b>	<b>134</b>
<b>1- PROTOCOLO DHCP .....</b>	<b>135</b>
<i>Introducción</i> .....	135
<i>Protocolo de aplicación DHCP</i> .....	137
<i>Escenario típico DHCP</i> .....	138
<b>2- DNS .....</b>	<b>141</b>
<i>DNS</i> .....	141
<i>Organización jerárquica del DNS</i> .....	146
<i>Proceso DNS en estructura jerárquica</i> .....	147
<b>3- TELNET Y FTP .....</b>	<b>149</b>
<i>TELNET</i> .....	149
<i>FTP</i> .....	150
<b>4- PROTOCOLOS DE CORREO EN INTERNET .....</b>	<b>152</b>
<i>SMTP</i> .....	152
<i>ESMTP</i> .....	153
<i>POP</i> .....	154
<i>IMAP</i> .....	155
<b>5- NOTICIAS Y GESTIÓN DE RED .....</b>	<b>156</b>
<i>NNTP</i> .....	156
<i>SNMP</i> .....	157
<b>6- WORLD WIDE WEB .....</b>	<b>158</b>
<i>WWW</i> .....	158
<i>Navegadores de la WWW</i> .....	159
<i>URL</i> .....	160
<i>HTML</i> .....	162
<i>HTTP</i> .....	164
<i>Escenario con HTTP</i> .....	165

## 1- Introducción

La asimilación de este capítulo permitirá:

- Comprender en qué consisten los protocolos de comunicaciones.
- Tener una visión general de la arquitectura de protocolos TCP/IP, así como situarla respecto al modelo de referencia OSI.
- Identificar las unidades de datos de cada uno de los protocolos que constituyen la arquitectura TCP/IP.
- Entender los mecanismos de encapsulación y demultiplexación de las unidades de datos.

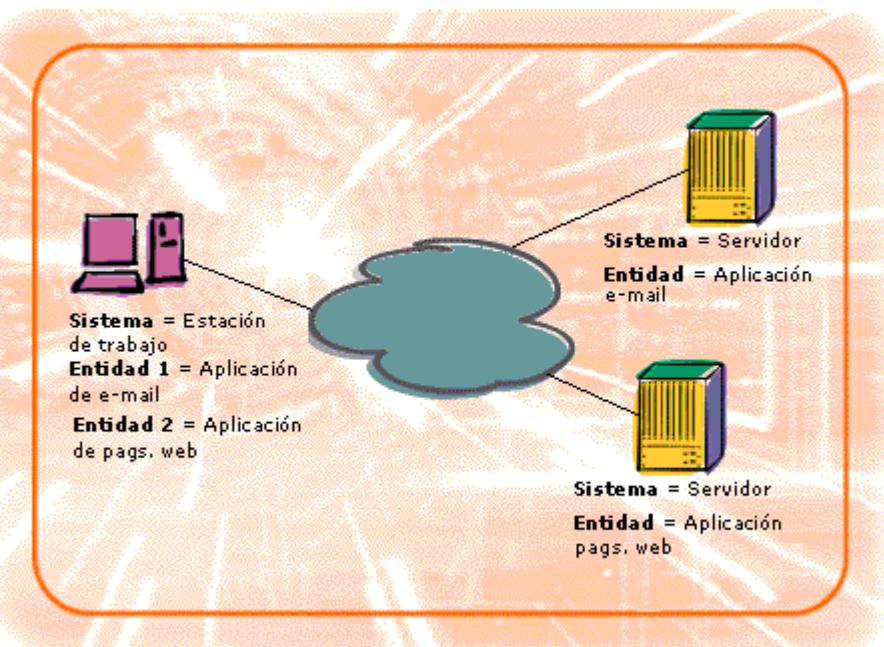
## 1- Protocolos de comunicaciones: modelo OSI y TCP/IP

¿Qué te parece si empezamos por el principio? ¿Qué es un protocolo?

### Protocolos de comunicaciones

Para la comunicación entre dos entidades situadas en sistemas diferentes es necesario la definición y utilización de un **protocolo**.

Los términos “entidad” y “sistema” se están usando en sentido muy general.



Ejemplos de entidades son:

Los programas de aplicación de los usuarios, las utilidades para transferencia de archivos, los sistemas de gestión de bases de datos, así como los gestores de correo electrónico y terminales.

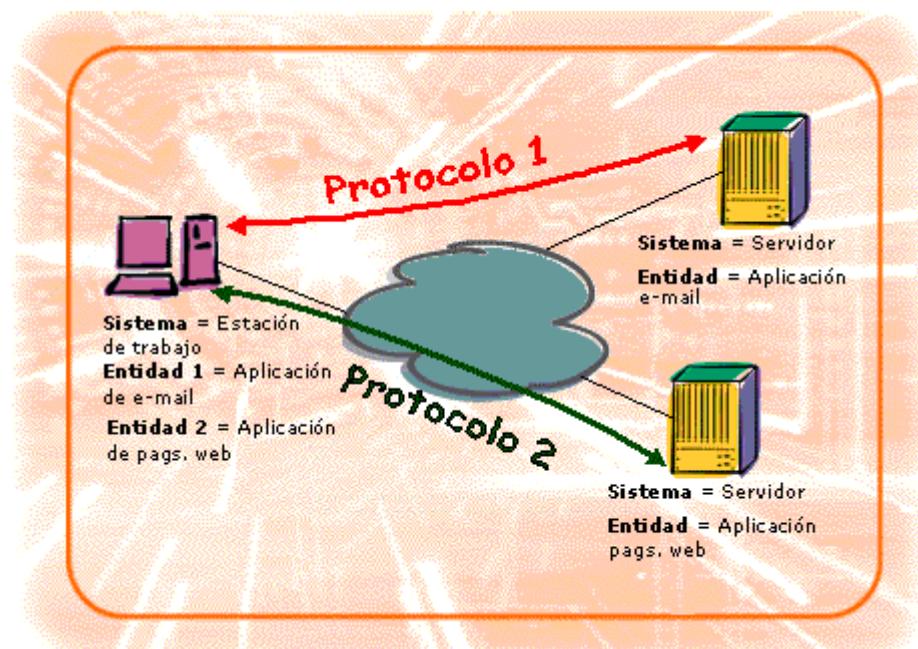
Ejemplos de sistemas son:

Los ordenadores, los terminales y los sensores remotos.

En algunos casos, la entidad y el sistema en que ésta se ubica son coincidentes.

En general, una entidad es cualquier cosa capaz de enviar y recibir información, y un sistema es un objeto físico de naturaleza distinta, que contiene una o más entidades.

Para que dos entidades se comuniquen con éxito, se requiere que "hablen el mismo idioma".



Lo que se comunica, cómo se comunica y cuándo se comunica, debe seguir una serie de convenciones mutuamente aceptadas por las entidades involucradas.

Este conjunto de convenios se denominan **protocolos**, que se pueden definir como el conjunto de reglas que gobiernan el intercambio de datos entre dos entidades.

Los puntos que definen o caracterizan un protocolo son:

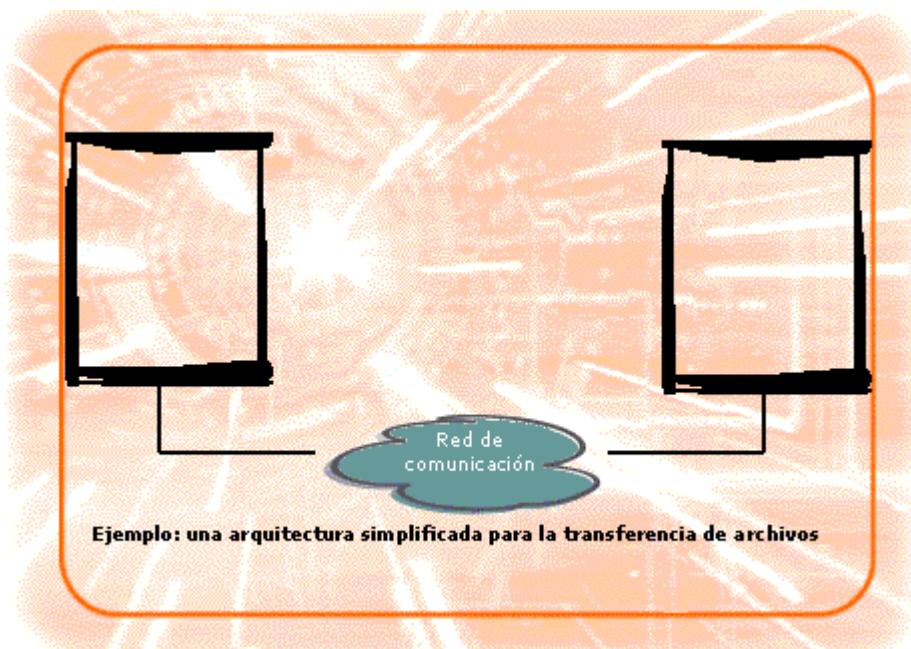
- La sintaxis: incluye aspectos tales como el formato de los datos y los niveles de señal.
- La semántica: incluye la información de control para la coordinación y el manejo de errores.
- La temporización: incluye la sintonización de velocidades y la secuenciación.

## Arquitectura de protocolos

En lugar de implementar toda la lógica para llevar a cabo la comunicación en una única y enorme tarea, es más práctico dividir en subtareas, cada una de las cuales se realice por separado.

Así, en vez de implementar módulos que realicen todas las tareas involucradas en la comunicación, se utilizan estructuras consistentes en conjuntos de módulos capaces de realizar cada uno de ellos una subtarea distinta. Aplicamos el "divide y vencerás".

Sigue leyendo y entenderás que queremos decir.



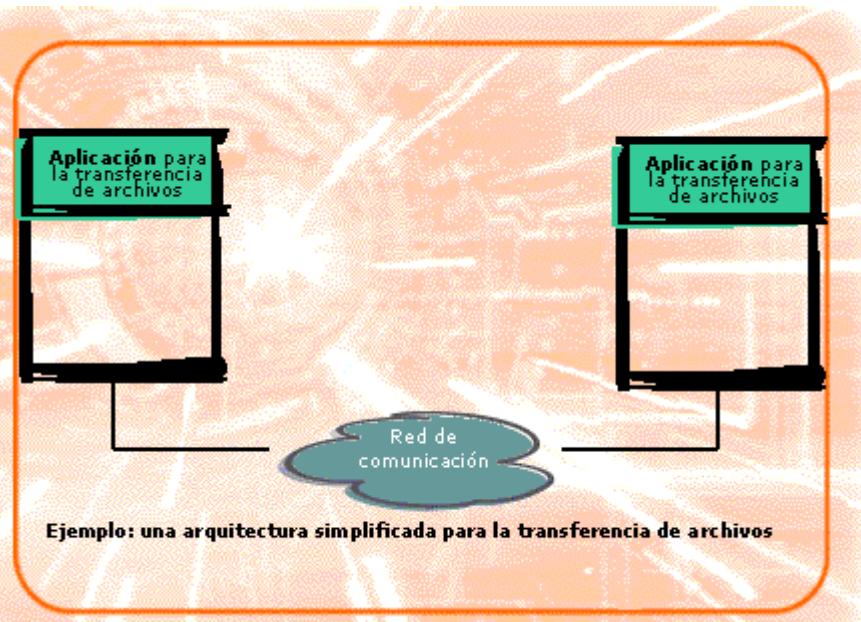
Estas estructuras multimodulares se denominan **arquitectura de protocolos**. De esta forma, cada módulo en el origen deberá intercambiar información con su módulo equivalente en el destino, y viceversa.

Así, cada módulo se convierte en una "entidad" independiente dentro del "sistema", existiendo un protocolo (conjunto de reglas) para la comunicación entre cada par de entidades equivalentes en origen y destino.

A modo de ejemplo, veremos a través de la figura como se podría desarrollar una aplicación de transferencia de archivos en la que se emplean tres módulos:

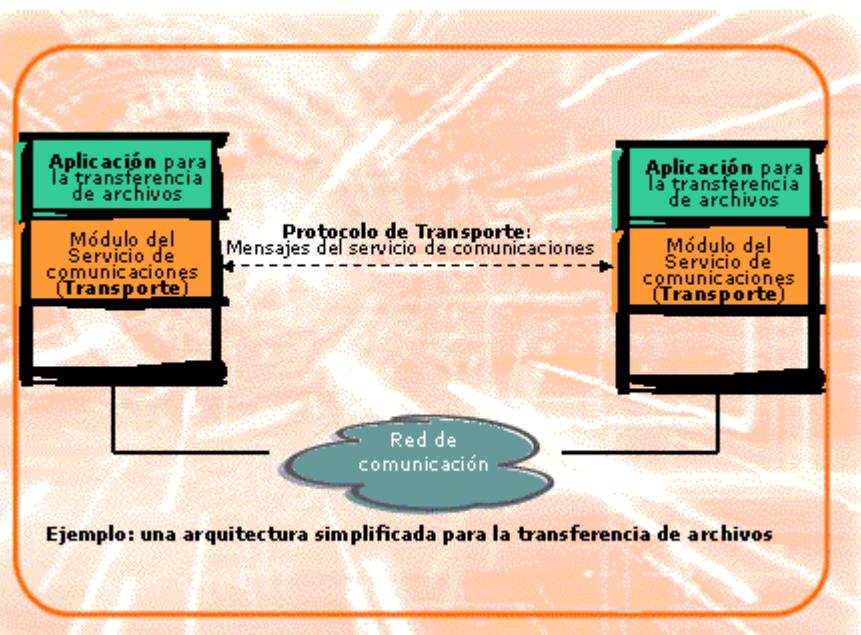
El **módulo de transferencia de archivos** se encargaría de:

- Asegurarse de que la aplicación destino está preparada para aceptar y almacenar el archivo para un usuario determinado.
- Verificar que el formato del archivo es entendido por ambos sistemas, origen y destino, y si no es así, encargarse de su adecuación.



Los módulos de transferencia de archivos en ambos sistemas intercambian archivos y órdenes.

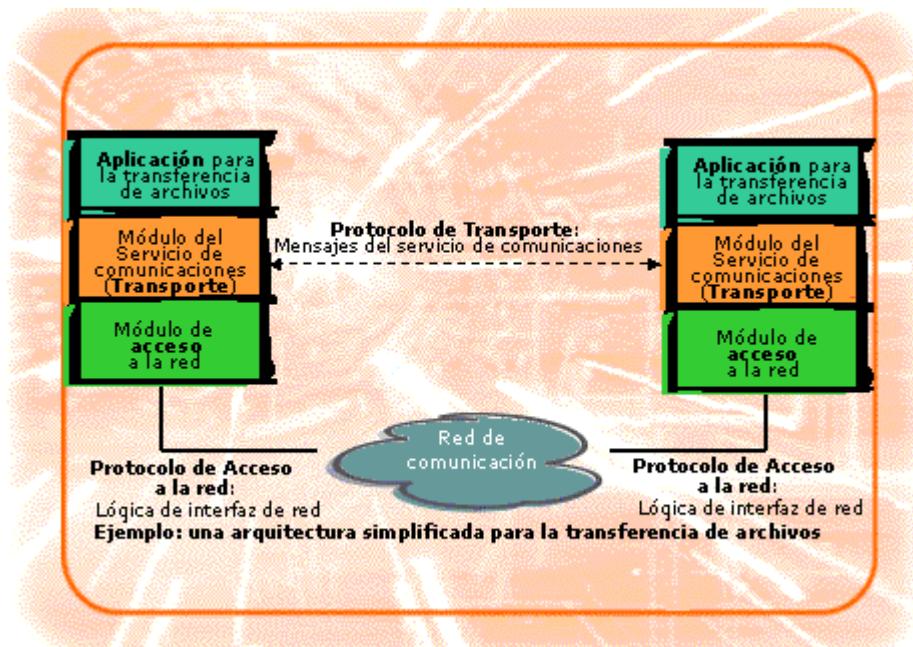
Sin embargo, en lugar de exigir que el módulo de transferencia se encargue de los detalles con los que se realiza el envío de datos y órdenes, dichos módulos delegan en los **módulos de servicio de comunicaciones**.



El módulo de servicio de comunicaciones se encargará de asegurar que el intercambio de órdenes y datos se realice fiablemente; para ello, el módulo de servicio de comunicaciones del sistema fuente debe asegurarse de que el destino está preparado para recibir datos.

Por otra parte, sería interesante que la naturaleza del intercambio entre los sistemas fuera independiente de la naturaleza de la red que los interconecta.

Para ello, en lugar de construir los detalles de la interfaz de red en el módulo de servicio de comunicaciones, se utiliza un módulo adicional de acceso a la red que se encarga de activar un camino directo de datos entre los dos sistemas, o bien de proporcionar a la red de comunicación la identificación del sistema destino deseado.



## Unidades de datos de los protocolos

Supongamos una operación sencilla: Una aplicación en el ordenador A quiere transmitir un mensaje a otra aplicación en el ordenador B.

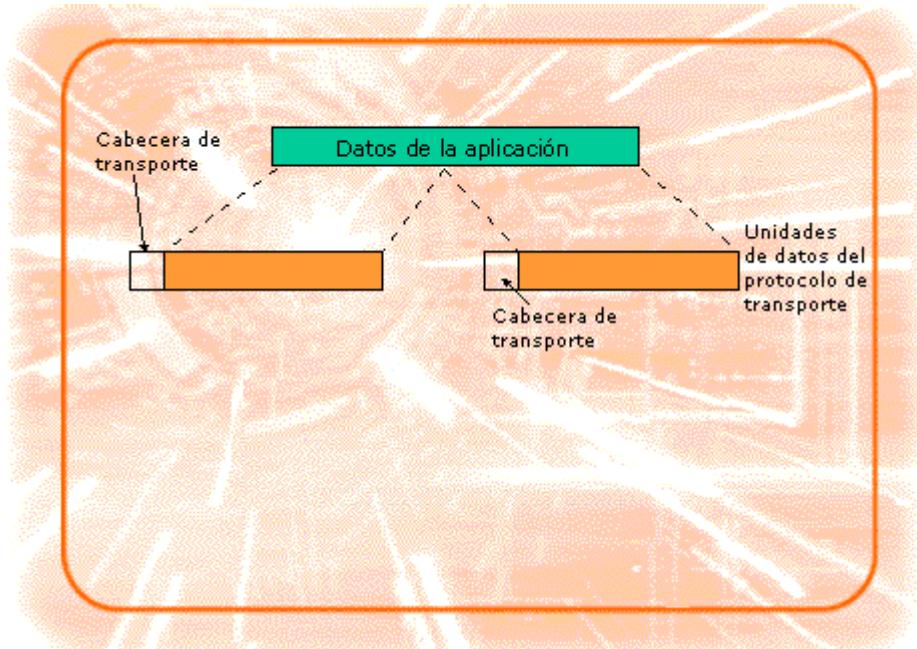


La aplicación en A pasa un mensaje con los datos que quiere enviar a la aplicación B a la capa de transporte de su sistema.

La capa de transporte pasa el mensaje a la capa de acceso, la cual proporciona las instrucciones necesarias a la red para que envíe el mensaje a B.

Supongamos que la aplicación emisora genera un bloque de datos y se lo pasa a la capa de transporte.

Esta última puede romper el bloque en unidades más pequeñas para hacerlas más manejables.



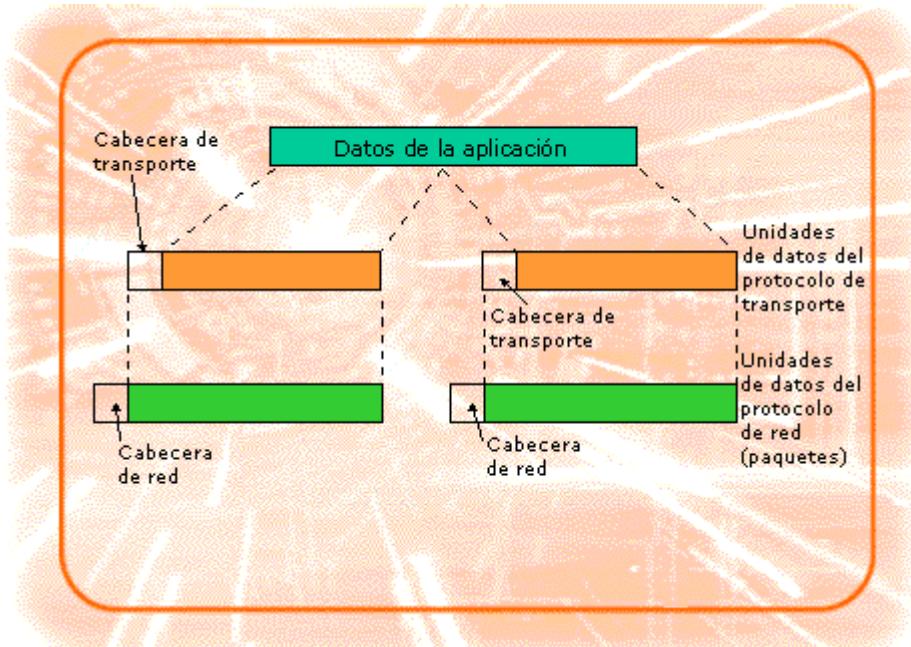
Como se ve en la figura, A cada una de estas pequeñas unidades la capa de transporte añadirá una cabecera, que contendrá información de control según el protocolo.

La unión de los datos generados por la capa superior junto con la información de control de la capa actual es lo que constituye una unidad de datos del protocolo (**PDU**, "Protocol Data Unit").

En el ejemplo se trataría de la unidad de datos del protocolo de transporte.

La cabecera en cada PDU de transporte contiene información de control que será usada por el mismo protocolo de transporte en el ordenador B (aplicación destino, num. de secuencia, código de detección de error, ...).

A continuación la capa de transporte pasa cada una de "sus" PDUs a la capa de red, con la instrucción de que sea transmitida al computador destino.

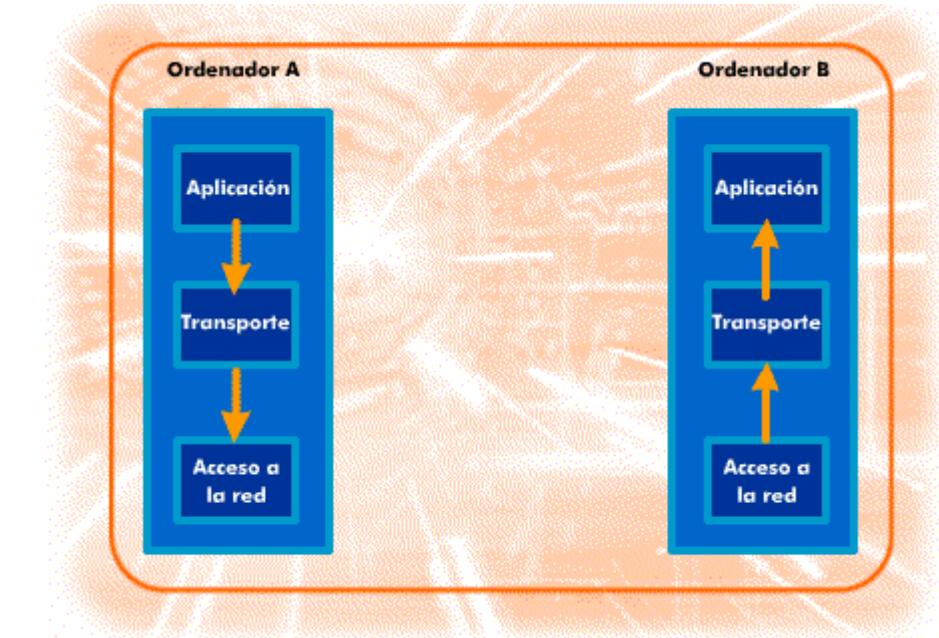


Para satisfacer este requerimiento, el protocolo de acceso a red debe pasar los datos a la red con una petición de transmisión.

Como anteriormente, esta operación requiere el uso de información de control.

En este caso, el protocolo de acceso a la red añade la cabecera de acceso a la red (dirección del ordenador destino, prioridades en la red, ...) a los datos provenientes de la capa de transporte, creando así la PDU de acceso a la red.

## Fucionamiento de una arquitectura de protocolos



El gráfico muestra las interacciones entre módulos necesarias para completar una comunicación.

Supongamos que el módulo de transferencia de archivos en el ordenador A está transfiriendo la información registro a registro al ordenador B (cuando hablamos de registros, hablamos simplemente de bloques de información).

Cada uno de los registros se pasa al módulo de la capa de transporte en una operación que se podría describir en términos informáticos como una llamada a un procedimiento.

Los argumentos de entrada de este procedimiento serían, la dirección del ordenador de destino, la aplicación de destino y el registro que estamos tratando de transmitir.

La capa de transporte añade la identidad de la aplicación origen e información de control adicional, que se agregará al registro para formar la PDU de transporte.

Ésta se pasa a la capa inferior de acceso a la red mediante la llamada a otro procedimiento.

En este caso, los argumentos de entrada del procedimiento serán la dirección del ordenador de destino y la PDU de la capa de transporte.

La capa de acceso a la red utilizará esta información para construir la PDU de red. Esto es lo que se denomina encapsulamiento. La PDU de red encapsula la PDU de transporte, esto es, la PDU de red está constituida por una cabecera y la PDU de transporte.

La unidad de datos del protocolo de transporte constituye el campo de datos de la PDU de red, y su cabecera contendrá información relativa a las direcciones destino y fuente.

Observa en que la cabecera de transporte no es "visible" al nivel de acceso a la red; en otras palabras, a dicho nivel no le concierne el contenido concreto de la PDU de transporte.

La red acepta la PDU de transporte de A y la transmite a B. El módulo de acceso a la red en B recibe la PDU, elimina la cabecera y pasa la PDU de transporte adjunta al módulo de la capa de transporte de B.

La capa de transporte examina la cabecera de la PDU de transporte y, en función del campo en la cabecera que contenga la identidad de la aplicación, entregará el registro correspondiente a la aplicación pertinente, en este caso al módulo de transferencia de archivos de B.

## El modelo OSI

- 
- ◆ Aplicación
  - ◆ Presentación
  - ◆ Sesión
  - ◆ Transporte
  - ◆ Red
  - ◆ Enlace de datos
  - ◆ Física

EL modelo OSI ("Open Systems Interconnection") se desarrolló por la Organización Internacional de estandarización ISO ("International Organization for Standardization") como una **arquitectura para comunicaciones entre ordenadores**, con el objetivo de ser el marco de referencia en el desarrollo de protocolos estándares.

Los diseñadores del modelo OSI consideraron que este modelo y sus protocolos asociados llegarían a dominar las comunicaciones entre ordenadores, reemplazando eventualmente las implementaciones particulares de protocolos, así como a modelos rivales tales como TCP/IP. Sin embargo, esto no ha sido así.

Si bien sigue siendo un modelo teórico de referencia, en la práctica, la arquitectura de siete capas no ha prosperado. Por el contrario, la arquitectura TCP/IP se ha erigido como dominante.

En cualquier caso, y dada su vigencia como modelo teórico, puede resultarte interesante conocer las funciones de cada una de las capas que lo constituyen.

### Aplicación

Proporciona el acceso al entorno OSI para los usuarios.

### Presentación

Proporciona independencia a los procesos de aplicación respecto a las diferencias en la representación de los datos (sintaxis).

## Sesión

Proporciona el control de la comunicación entre las aplicaciones; establece, gestiona y cierra las conexiones (sesiones) entre las aplicaciones cooperadoras.

## Transporte

Proporciona seguridad, transferencia transparente de datos entre puntos finales; proporciona además, procedimientos de recuperación de errores y de control de flujo origen-destino.

## Red

Proporciona independencia a los niveles superiores respecto a las técnicas de conmutación y de transmisión utilizadas para conectar los sistemas; es responsable del establecimiento, mantenimiento y cierre de las conexiones.

## Enlace de datos

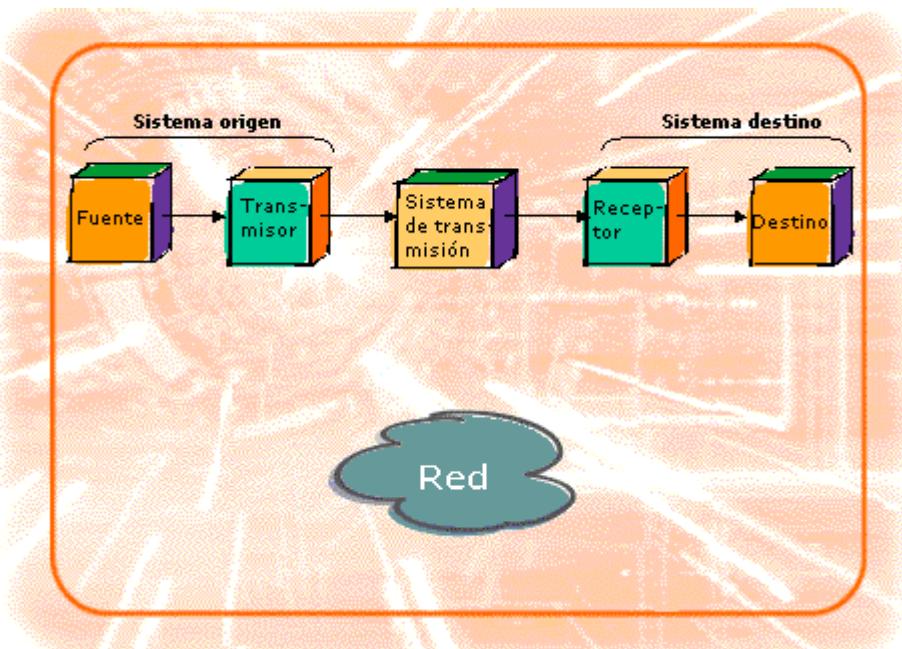
Proporciona un servicio de transferencia de datos seguro a través del enlace físico; envía bloques de datos (tramas) llevando a cabo la sincronización, el control de errores y de flujo necesarios.

## Física

Se encarga de la transmisión de cadenas de bits no estructurados sobre el medio físico; está relacionada con las características mecánicas, eléctricas, funcionales y de procedimiento para acceder al medio físico.

## El modelo TCP/IP

El conjunto de protocolos TCP/IP (habitualmente se suele hablar de la pila de protocolos TCP/IP) es, como ya hemos dicho, la arquitectura dominante para la **interconexión de sistemas**.

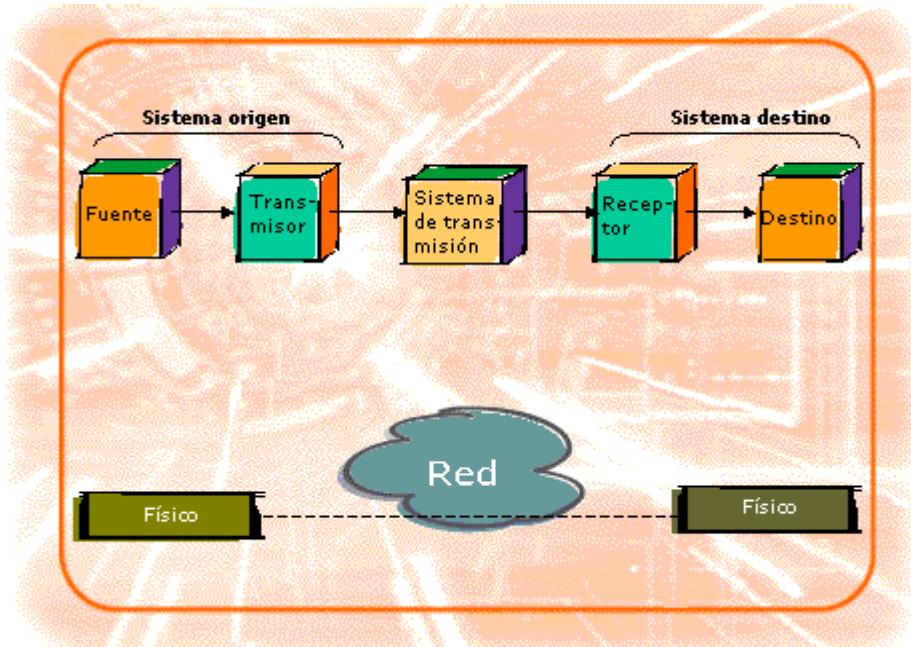


Al contrario que en OSI, no hay un modelo oficial de referencia TCP/IP.

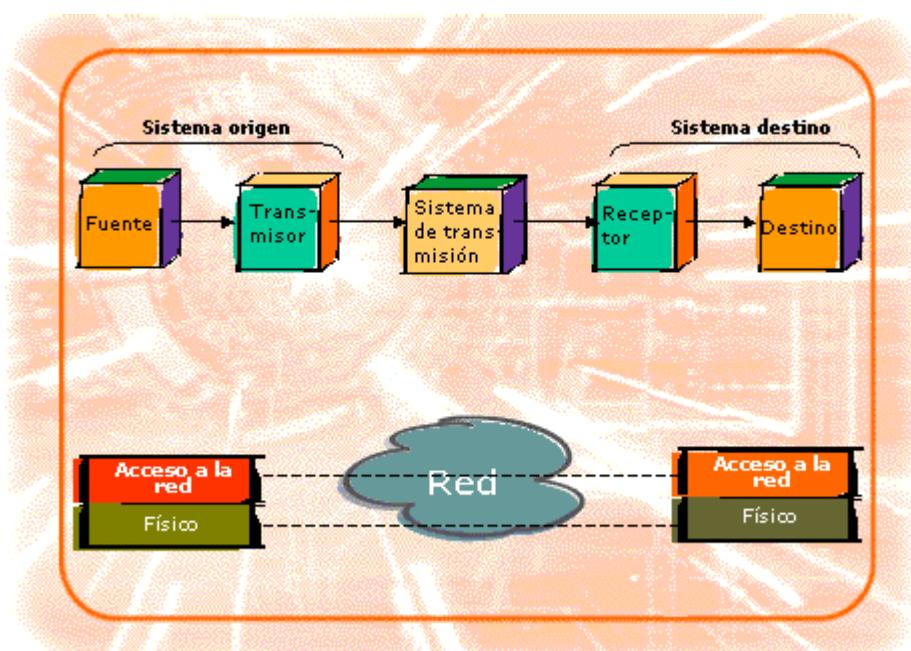
No obstante, basándose en los protocolos estándar que se han desarrollado, todas las tareas involucradas en la comunicación se pueden organizar en cinco capas relativamente independientes:

La **capa física** contempla la interfaz física entre el dispositivo de transmisión de datos (por ejemplo, el ordenador) y el medio de transmisión o la red.

Esta capa está relacionada con la especificación de las características del medio de transmisión, la naturaleza de las señales, la velocidad de datos y cuestiones afines.

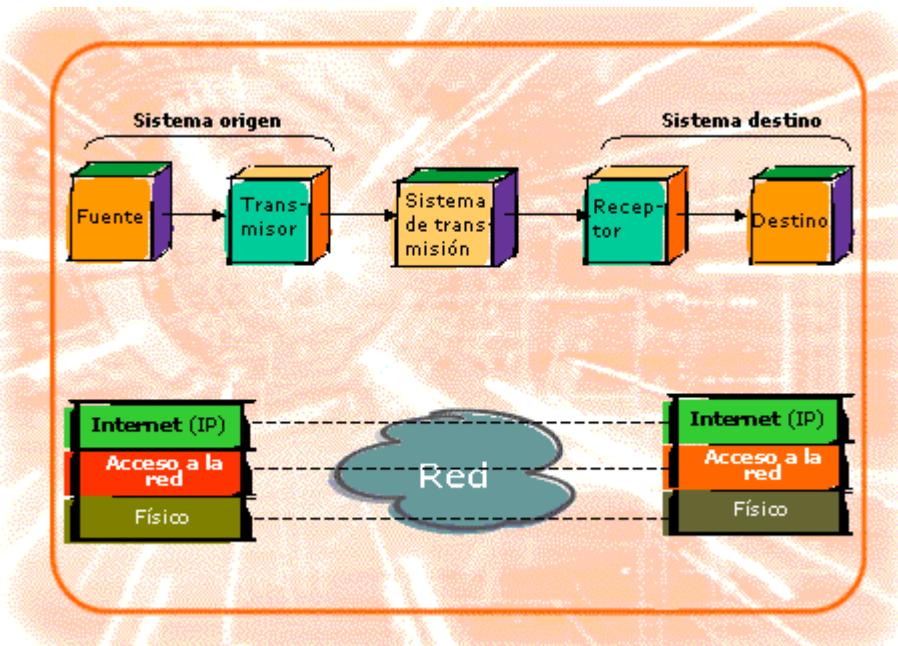


La **capa de acceso a la red** es responsable del intercambio de datos entre el sistema final y la red a la cual se está conectado. El emisor debe proporcionar a la red la dirección del destino, de tal manera que la red pueda encaminar los datos hasta el destino apropiado. El software en particular que se use en esta capa dependerá del tipo de red que se disponga; se han desarrollado diversos estándares para conmutación de circuitos, conmutación de paquetes (por ejemplo X.25), redes de área local (p.e. Ethernet), entre otros. Por tanto, es coherente separar las funciones relacionadas con el acceso a la red en una capa independiente.



La capa de acceso a la red está relacionada con el acceso y el encaminamiento de los datos a través de la red.

En situaciones en las que los dispositivos estén conectados a redes diferentes, se necesitarán una serie de procedimientos para permitir que los datos atraviesen las diferentes redes interconectadas. Ésta es la función de la **capa Internet**.

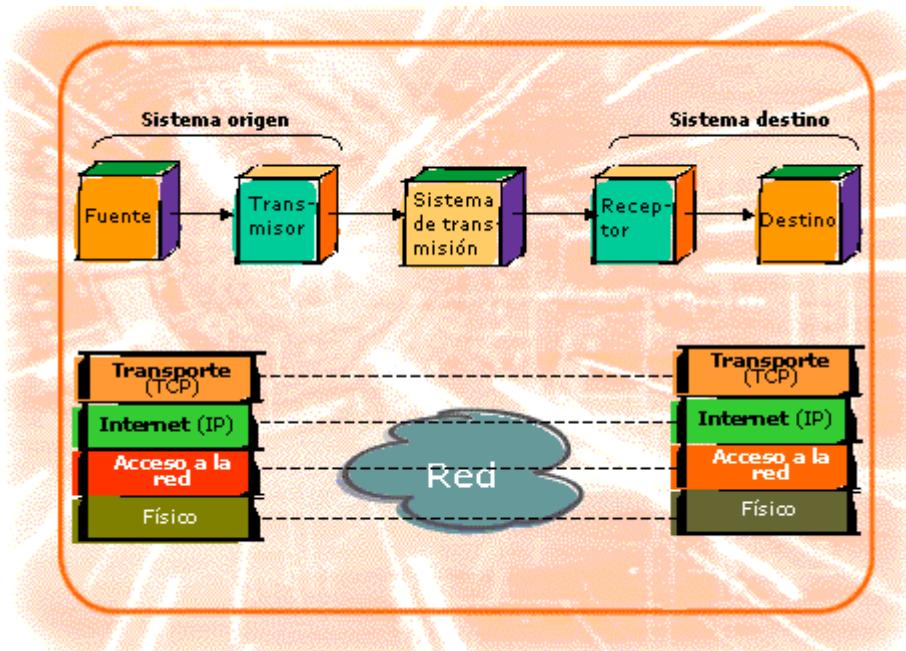


El protocolo internet (**IP** "internet protocol") se utiliza en esta capa para ofrecer un servicio de encaminamiento a través de distintas redes. Este protocolo se implementa tanto en los sistemas finales como en los "routers" intermedios.

Un "**router**" es un dispositivo con capacidad de procesamiento que conecta dos redes y cuya función principal es retransmitir datos de una red a otra, de forma que se establezca un camino que llegue hasta el destino.

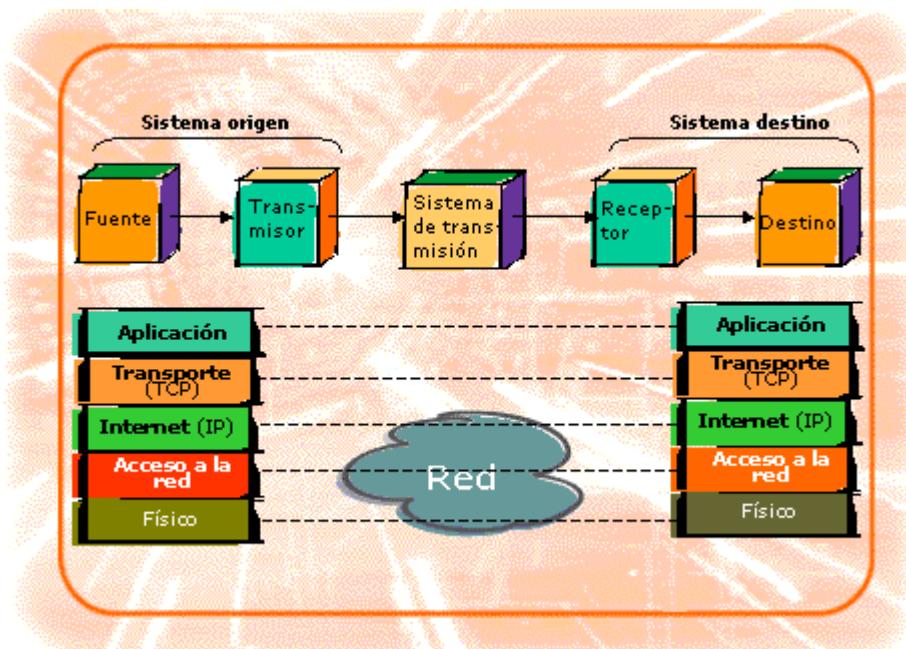
Independientemente de la naturaleza de las aplicaciones que estén intercambiando datos, es usual requerir que los datos se intercambien de forma segura. Esto es, sería deseable asegurar que todos los datos llegan a la aplicación destino y en el mismo orden en el que fueron enviados.

Por tanto, es justificable agrupar todos los mecanismos necesarios para ofrecer seguridad en una capa que sea compartida por todas las posibles aplicaciones, denominada **capa de transporte**. El protocolo **TCP** ("Transmission Control Protocol") es el más utilizado para proporcionar estas funciones.



Finalmente, la **capa de aplicación** contiene toda la lógica necesaria para llevar a cabo las aplicaciones de usuario.

Para cada tipo específico de aplicación, como es por ejemplo la transmisión de un archivo, se necesitará un módulo particular dentro de esta capa.



## Tecnologías de red

- ◆ Líneas punto a punto.
- ◆ Servicios de área extensa de conmutación de circuitos.
- ◆ Servicios de área extensa de conmutación de paquetes.
- ◆ Servicios de conmutación de celdas.
- ◆ **Tecnologías LAN.**

Durante los últimos años ha aparecido un número sin precedentes de tecnologías innovadoras de LAN (Redes de Área Local) y WAN (Redes de Área Amplia) que han sido absorbidas rápidamente por el mercado. Se ha introducido el uso del par trenzado, el coaxial y la fibra a un ritmo que nadie hubiese podido predecir.

Red digital de servicios integrados (RDSI), "Frame Relay", T1, T1 fraccional, T3, SDH, líneas de fibra óptica, Servicio Conmutado de Datos Multimegabit (SMDS), conexiones de cable y ATM, son algunas de las tecnologías de área extensa con las que convivimos. El punto en común entre ellas es que todas prometen conexiones rápidas y baratas.

El conjunto de protocolos TCP/IP es independiente del tipo de tecnología de red que se utilice (hablamos de las capas de enlace de datos y física).

### **Líneas punto a punto**

HDLC, PPP, SLIP.

### **Servicios de área extensa de conmutación de circuitos**

Digitales: RDSI.

### **Servicios de área extensa de conmutación de paquetes**

Modo circuito: X.25, Frame Relay.

### **Servicios de conmutación de celdas**

ATM.

### **Tecnologías LAN**

Ethernet, Token Ring, etc.

## 2- Historia e implementación de TCP/IP

Ahora que tenemos una idea un poco más clara acerca de lo que es un protocolo de comunicaciones, vamos a hablar del que, "de facto" es, el estándar mundial. El protocolo TCP/IP. Comenzaremos con una interesante introducción histórica.

### Nacimiento de TCP/IP: ARPANET



A finales de los años 60, la Agencia de proyectos avanzados de investigación del Departamento de Defensa de los Estados Unidos (ARPA, posteriormente llamada DARPA) comenzó una asociación con universidades de los Estados Unidos y otros organismos de investigación para experimentar sobre nuevas tecnologías de comunicación de datos.

Entre los partícipes, construyeron la Red de la agencia de proyectos avanzados de investigación: **ARPANET**, *Advanced Research Projects Agency Network*, la primera **red de comutación de paquetes**.

En 1969 comenzó a funcionar una versión experimental de ARPANET con cuatro nodos.

El experimento fue un éxito y, a partir de ahí, evolucionó hasta cubrir los Estados Unidos de costa a costa.

En 1975, la Agencia de comunicaciones para la defensa (DCA, *Defense Communications Agency*) asumió la responsabilidad del funcionamiento de la red, que aún era considerada una red de investigación.

Los protocolos iniciales de ARPANET eran lentos y solían sufrir frecuentes problemas.

En el año 1974 se sentaron las bases del Protocolo de Internet (**IP**) y del Protocolo de Control de Transporte (**TCP**), sobre los cuales se construiría la nueva ARPANET.

A principios de los años 80, ARPANET se convirtió a los nuevos protocolos.

En 1983, ARPANET contaba con más de 300 nodos y se había convertido en un valioso recurso para los usuarios.

En 1984, la ARPANET original se dividió en dos partes.

Una se siguió llamando ARPANET y se dedicó a la investigación y desarrollo.

La otra se llamó MILNET y se convirtió en una red militar no clasificada.

## Características de TCP/IP

- ◆ Permite la creación de bancos de redes, creando una red mayor llamada **Internet**.
- ◆ Independencia del hardware de los nodos y de la tecnología de red subyacente, con capacidad de encaminamiento adaptativo transparente al usuario.
- ◆ Disponibilidad de TCP/IP.

La figura muestra algunas de las características que justifican la durabilidad del protocolo TCP/P.

La arquitectura TCP/IP agrupa bancos de redes, creando una red mayor llamada **Internet**.

Para el usuario, Internet aparece, simplemente, como una red única compuesta por todos los equipos de usuario conectados a cualquiera de los nodos que la forman.

Los protocolos TCP/IP se diseñaron para ser independientes del hardware del host o de su sistema operativo, así como de las tecnologías de los medios y enlaces de datos.

Se requería que los protocolos fuesen robustos, sobreviviendo a altas tasas de error en la red, y con capacidad de encaminamiento adaptativo transparente en el caso de que se perdiesen los enlaces.

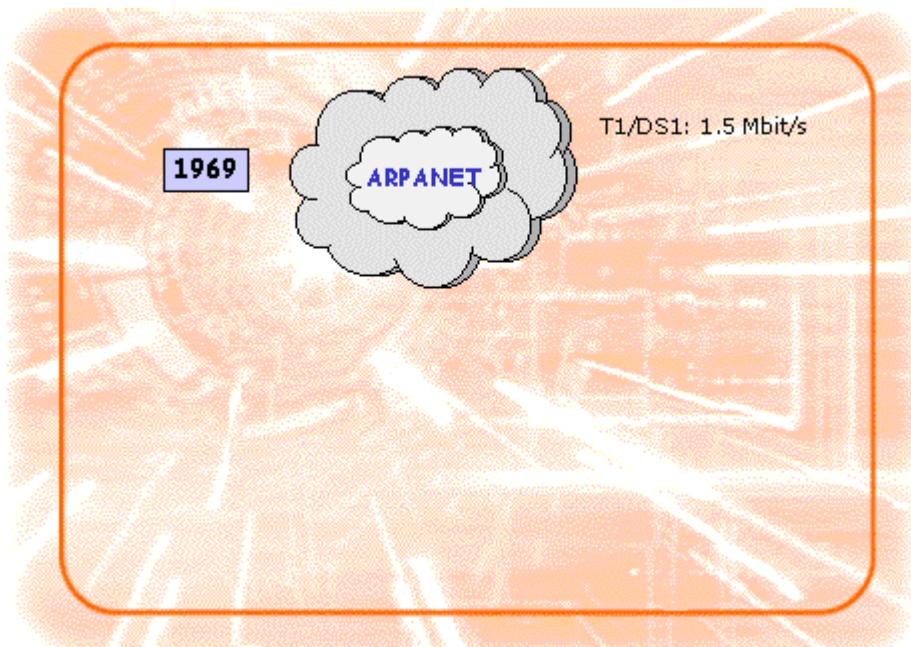
Cuando el Departamento de Defensa de los Estados Unidos y otras agencias gubernativas impusieron como requisito los protocolos TCP/IP en la compra de sus ordenadores, los fabricantes se enfrentaron a la necesidad de implementar TCP/IP para competir en los concursos del gobierno.

En los años 90, TCP/IP llegó al mundo comercial. Es el software de red más disponible universalmente. Ha habido un rápido progreso al integrar TCP/IP junto con los servidores de LAN y los sistemas operativos de sobremesa. Además, existe el soporte para TCP/IP sobre una selección creciente de tecnologías de transmisión.

## Internet

La facilidad para agrupar las redes TCP/IP combinada con una política de puertas abiertas, que permitió a las redes de investigación, académicas y comerciales conectarse a ARPANET, hizo crecer la super red llamada **Internet**.

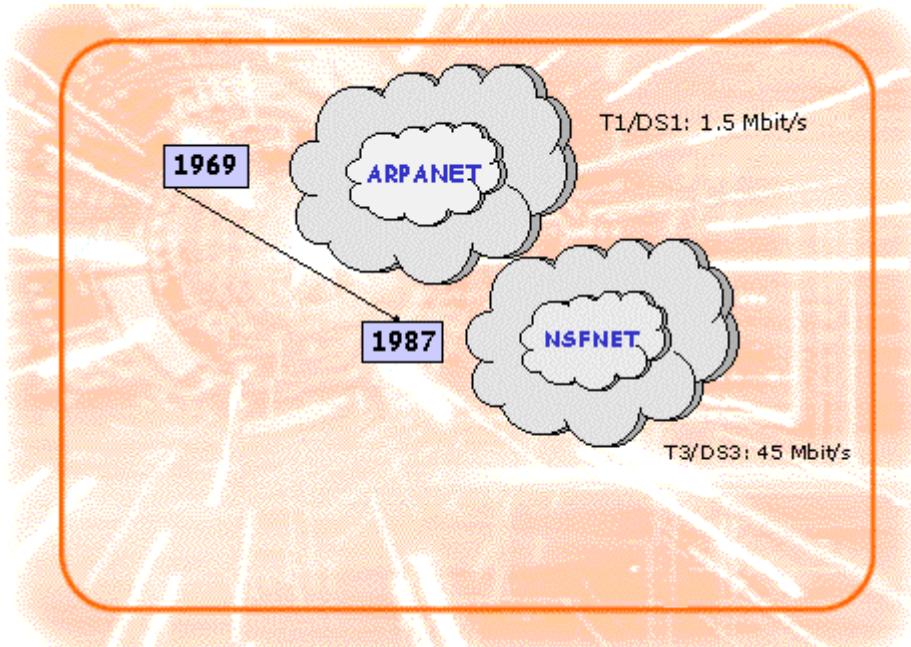
Durante los años 80, ARPANET se mantenía como la red troncal de esta creciente red.



Gracias a las características de los protocolos de TCP/IP, el crecimiento de Internet era continuado y sin pausa.

Internet se convirtió en la mayor red del mundo, contando básicamente con redes del gobierno, de investigación militar y académica y redes comerciales, cada una con cientos de subredes.

En 1985 se incorporó una nueva red troncal, la red de la *National Science Foundation* (NSFNET) que tenía como fin dar soporte a enlaces de alta velocidad para centros de investigación y supercomputación.



Gracias a la ayuda del gobierno y a una infraestructura de proveedores de servicios distribuidos por todos los Estados Unidos tuvo un gran crecimiento.

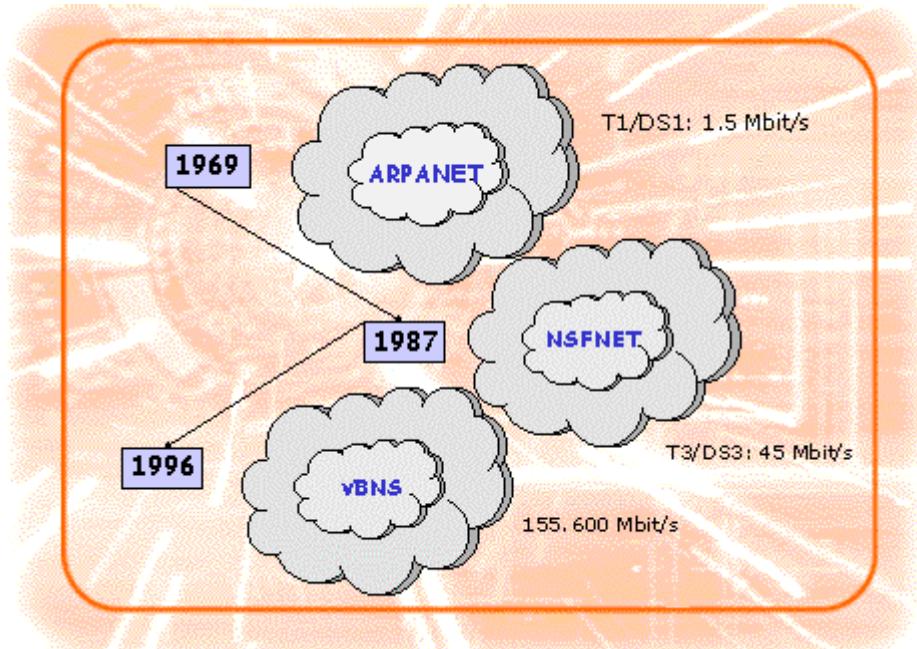
Las universidades y laboratorios de investigación se conectaban al proveedor regional más cercano que, a su vez, se interconectaba a la red troncal.

Internet se extendió por todo el mundo, apareciendo Proveedores de servicios en docenas de países.

En 1994, había millones de ordenadores interconectados e Internet estaba preparada para el mercado comercial.

La *National Science Foundation* (NSF) dió su apoyo y los Proveedores de servicios de los Estados Unidos se conectaron unos a otros en grandes centros de conmutación distribuidos por todo el país.

En 1996 se sustituye la red NSFNET por una nueva red troncal denominada vBNS (*very high speed Backbone Network Service*), que hace uso de las capacidades de conmutación avanzada (ATM) y trasmisión por fibra óptica (SONET) de la red nacional de MCI.



La combinación de SONET y ATM permite altas velocidades y alta capacidad para que señales de voz, datos y vídeo sean combinadas y trasmítidas "bajo demanda".

## Organismos reguladores y normas

- ◆ InterNIC
- ◆ IAB
- ◆ IEFT
- ◆ REFCs

**InterNIC** son, en realidad, dos agencias que dependen del NFS y que se ocupan de:

- Ofrecer servicios de registro de nombres y direcciones de ordenadores interconectados bajo TCP/IP. Hoy día se han establecido centros de registro adicionales en otros países.
- Ofrecer servicios de directorio y base de datos que funcionan como la fuente de las normas de Internet y otros documentos de información. Todos los documentos son gratuitos.

La coordinación en el desarrollo de los nuevos protocolos de TCP/IP, y el mantenimiento de los antiguos, lo lleva una organización llamada **Internet Architecture Board (IAB)**.

El IAB identifica las áreas técnicas que es necesario tratar. Por ejemplo, en los últimos años, el IAB ha encabezado los esfuerzos para desarrollar nuevos protocolos de gestión de red, mejores protocolos de encaminamiento y una nueva generación de la versión IP.

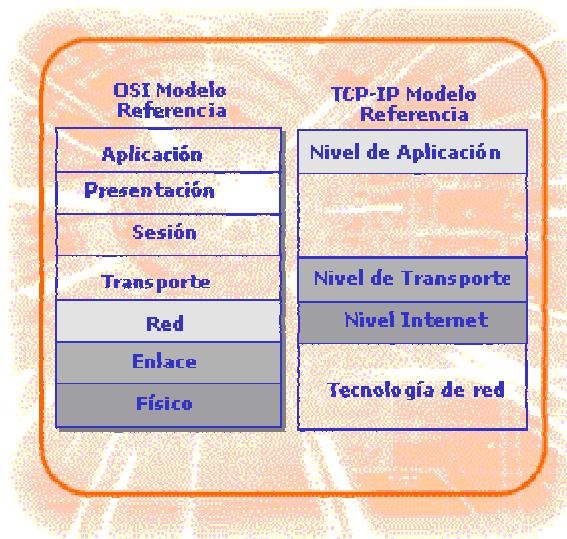
El **Internet Engineering Task Force (IETF)** es un conjunto de grupos de trabajo, coordinados por el IAB, que se ocupa del diseño e implementación de nuevos protocolos.

Los miembros del IETF son voluntarios. Para resolver un problema, se forma un grupo de trabajo cuyos miembros tengan la experiencia técnica necesaria. Los partícipes en un grupo de trabajo usan una metodología que combina la teoría con la implementación inmediata. De hecho, la validez y completitud de una especificación de un protocolo se comprueba creando, al menos, dos implementaciones independientes.

Una nueva especificación de un protocolo se distribuye en un documento que se llama Petición de comentarios (**RFC - Request For Comments**). Los documentos RFC se numeran en secuencia y se pueden obtener a través de los Servicios de directorio y base de datos de InterNIC, el cual mantiene un índice de las RFC y entradas para las RFC obsoletas que contienen los números de documentos que los sustituyen.

No todas las RFC describen protocolos. Algunas sólo organizan y presentan ideas que han evolucionado dentro de la comunidad Internet.

## OSI y TCP/IP



La principal diferencia entre el modelo OSI y la pila de protocolos TCP-IP es el número de niveles: en TCP-IP no existen los niveles de presentación y sesión y no se definen los niveles físico y de enlace. Esto se debe a la independencia de la tecnología de red y a la existencia de numerosas técnicas de encapsulamiento para llevar IP sobre las tecnologías de red existentes.

Los diferentes niveles que constituyen el TCP/IP son:

- Nivel de aplicación.
- Nivel de transporte.
- Nivel Internet.

### **Nivel de Aplicación**

Define los protocolos estándares de aplicación en Internet.

### **Nivel de Transporte**

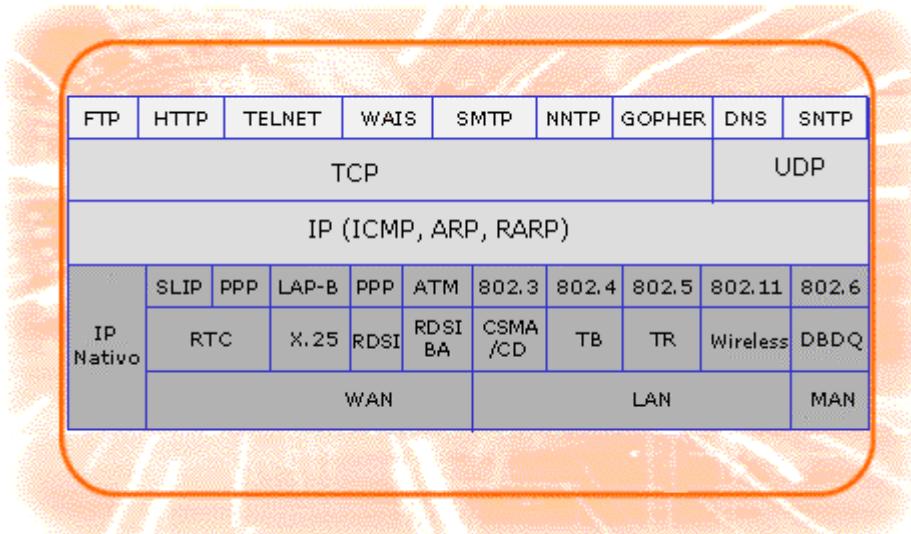
Define conexiones extremo a extremo y se encarga del control de flujo. En este nivel se han definido dos protocolos:

- **TCP** (Protocolo de control de transferencia). Orientado a conexión y confiable.
- **UDP** (Protocolo de datagrama de usuario). Mantiene la filosofía de IP (no orientado a conexión).

### **Nivel Internet**

Definido por el protocolo IP (Protocolo Internet). Es el responsable del servicio de entrega de paquetes, es no confiable porque la entrega no está garantizada, es sin conexión porque cada paquete es tratado independientemente y es "de mejor esfuerzo" porque la red no descarta paquetes caprichosamente, solamente cuando los recursos están agotados o la red falla.

## Implementación de TCP/IP



Como hemos dicho, la arquitectura TCP/IP no especifica el nivel inferior, esto es así porque el protocolo IP es implementable sobre diversas redes físicas. Eso sí, la red física que se use determinará la forma en que se encapsulen los datos.

IP puede usarse sobre:

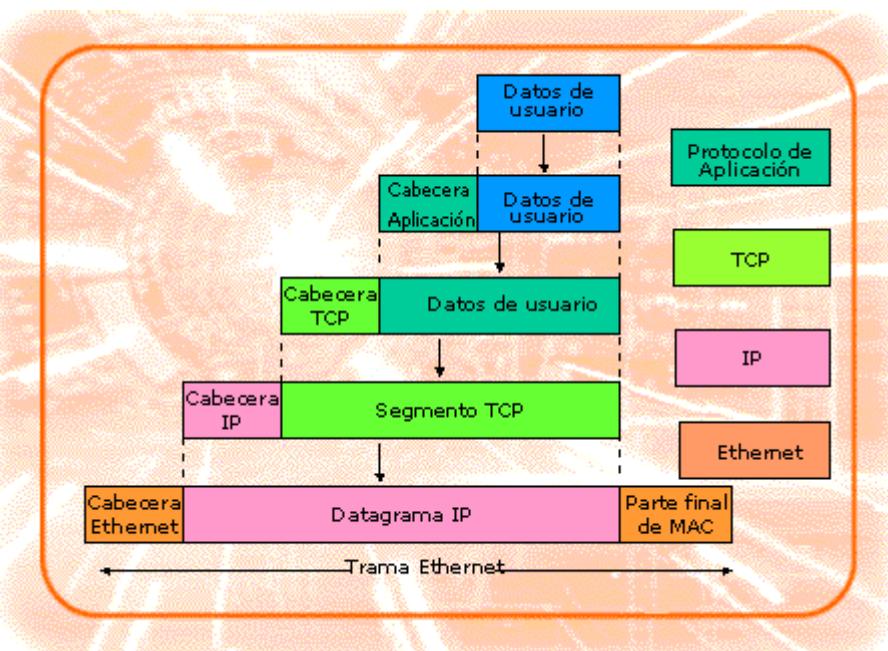
- Redes nativas IP (líneas dedicadas).
- RTC (Redes de conmutación de circuitos).
- Redes de conmutación de paquetes (X.25 y Frame Relay).
- RDSI.
- Redes ATM.
- Redes de área local y metropolitana (IEEE 802.x).

En algunos casos el encapsulamiento es especialmente complejo.

Por ejemplo, IP encapsulado en una trama Ethernet, puede ser nuevamente encapsulado y transmitido a través de redes ATM, o ATM, sobre interfaz ADSL, etc.

## Encapsulamiento de TCP/IP

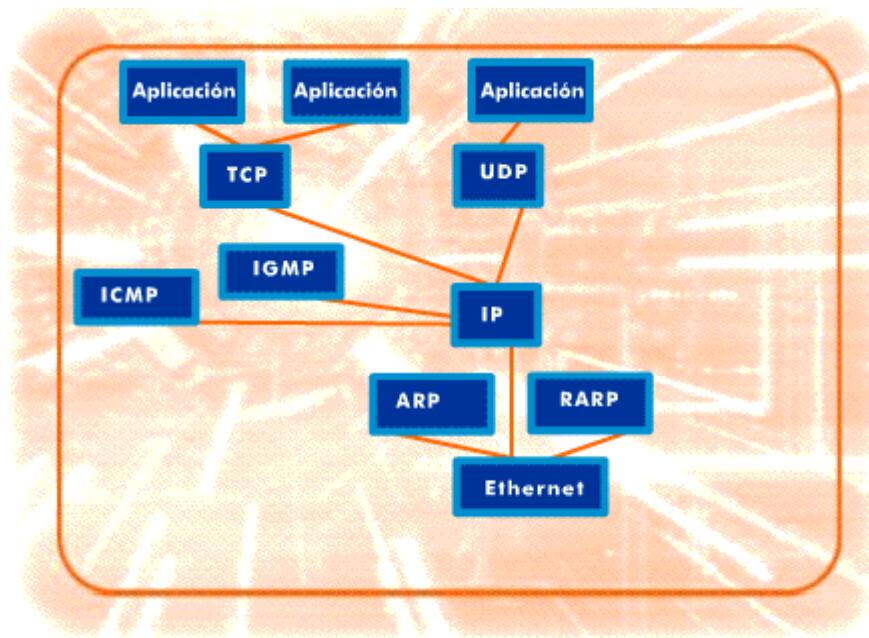
Cuando una aplicación envía información de un equipo a otro, los datos se encapsulan en uno o más segmentos TCP (en función de la longitud del mensaje). Hablar de segmentos TCP equivale a hablar de las PDUs del nivel de transporte.



Los segmentos, a su vez, se encapsulan en datagramas IP, que son transportados por el protocolo de red subyacente (ej. Ethernet). Hablar de datagramas IP equivale a hablar de las PDUs del nivel de Internet.

El ejemplo de la figura correspondería a la utilización del TCP/IP sobre una red Ethernet.

## Demultiplexación en TCP/IP



Cuando un nodo conectado a una red (Ethernet) recibe una trama, extrae de ella la cabecera Ethernet. Esta cabecera guarda la dirección física (MAC) del equipo destino, así como información acerca del protocolo que transporta, por lo general IP.

De la cabecera del datagrama IP se extrae información acerca de la dirección IP y del protocolo contenido en el datagrama (TCP, UDP, ICMP, etc.).

De las cabeceras TCP y UDP se extrae información acerca del puerto lo que indica el protocolo del nivel de aplicación destinatario del mensaje.

## 2- Direcciones Internet

Todos los nodos de una red necesitan una dirección. ¿Cómo se asigna? Puede que no sea un problema en una LAN autónoma con un puñado de equipos de usuario (host), pero cuando se trata de cientos de miles de equipos, empezar con un buen plan de direcciones puede ahorrar muchos problemas cuando se añaden, eliminan o reubican host, encaminadores o redes.

Los administradores de Internet tenían que tratar con la gestión de direcciones de una red cuyo tamaño se duplicaba, más o menos, cada año. Utilizaron una estrategia práctica, delegar.

El esquema de Internet de TCP/IP para la gestión de direcciones:

- Permite delegar la asignación de direcciones a un responsable de toda o parte de una red particular.
- Asigna direcciones que reflejan la topología lógica de la red de la organización.

## 1- Direcciones IP

Te explicaremos como funciona el protocolo por el que "adquieren su nombre" las máquinas en Internet.

### Direcciones IP

El protocolo IP utiliza direcciones IP para identificar los host y encaminar los datos hacia ellos.

**Todos los host deben** tener una dirección IP única para las comunicaciones.

El "nombre" de un host se traduce a su dirección IP consultando el nombre en una base de datos de pares nombre-dirección. Por esta razón, cuando navegamos en Internet usamos nombres del tipo "www...." en vez de direcciones IP.

- Identificación de los nodos mediante números.

Cuando se diseñaron las direcciones IP, nadie había soñado que llegase a haber millones de ordenadores en el mundo, y que muchos de ellos quisieran o necesitasen una dirección IP. Los diseñadores pensaron que tenían que satisfacer las necesidades de una modesta comunidad de universidades, grupos de investigación y organizaciones gubernativas y militares. Eligieron un diseño que les parecía razonable por entonces.

Una dirección IP es un número binario de 32 bits (cuatro octetos), por lo que se pueden direccionar como máximo 4.294.967.296 dispositivos distintos.

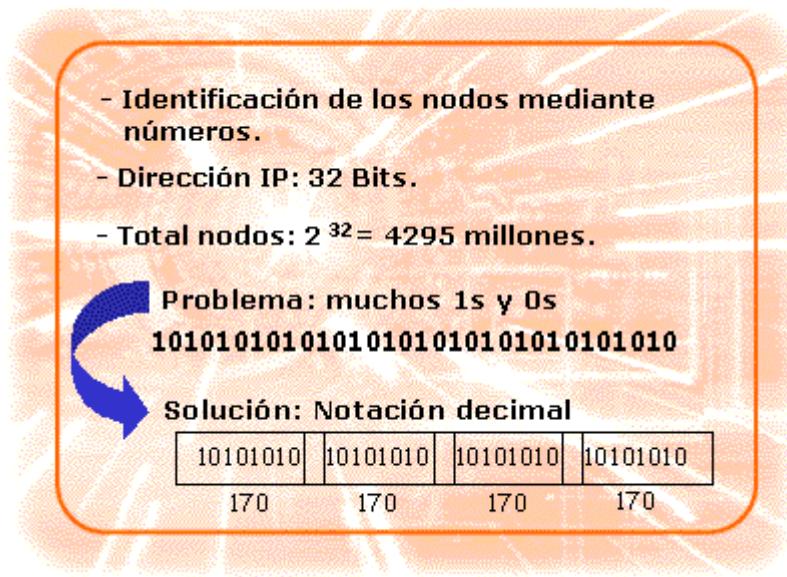
- Identificación de los nodos mediante números.
- Dirección IP: 32 Bits.

Claramente, la dirección se eligió para que encajase convenientemente en un registro de 32 bits de un ordenador.

El espacio de direcciones resultado, es decir, el conjunto de todos los números de direcciones posibles contiene  $2^{32}$  (4.294.967.296) números.

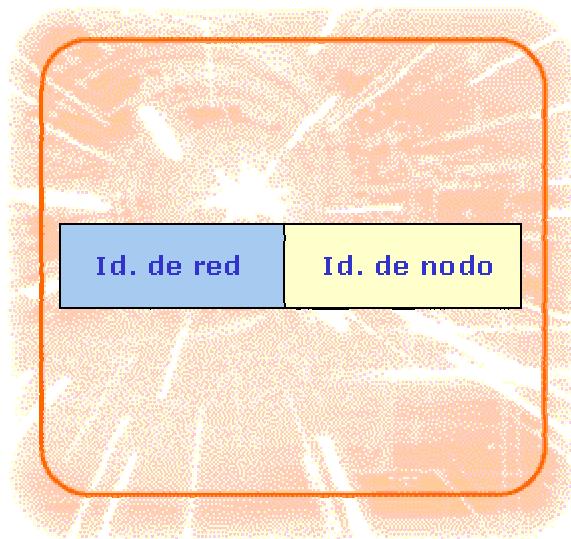
- Identificación de los nodos mediante números.
- Dirección IP: 32 Bits.
- Total nodos:  $2^{32} = 4295$  millones.

La notación "punto" se inventó para leer y escribir fácilmente las direcciones IP. Cada octeto de una dirección se convierte a su número decimal, y los números se separan por puntos. Por ejemplo, la dirección www.yahoo.com es un número binario de 32 bits (1101100010000000100101000110100) que en notación punto es: 216.32.74.52.



Hay que tener en cuenta que el mayor número que puede aparecer en una posición dada es 255, que corresponde al número binario 11111111.

## Formato de direcciones IP



Como se muestra en la figura, una dirección IP tiene un formato de dos partes que son la **dirección de red** y la **dirección local**.

La dirección de red identifica a la red a la que está conectado el nodo.

La dirección local identifica a un nodo particular dentro de una red de una organización.

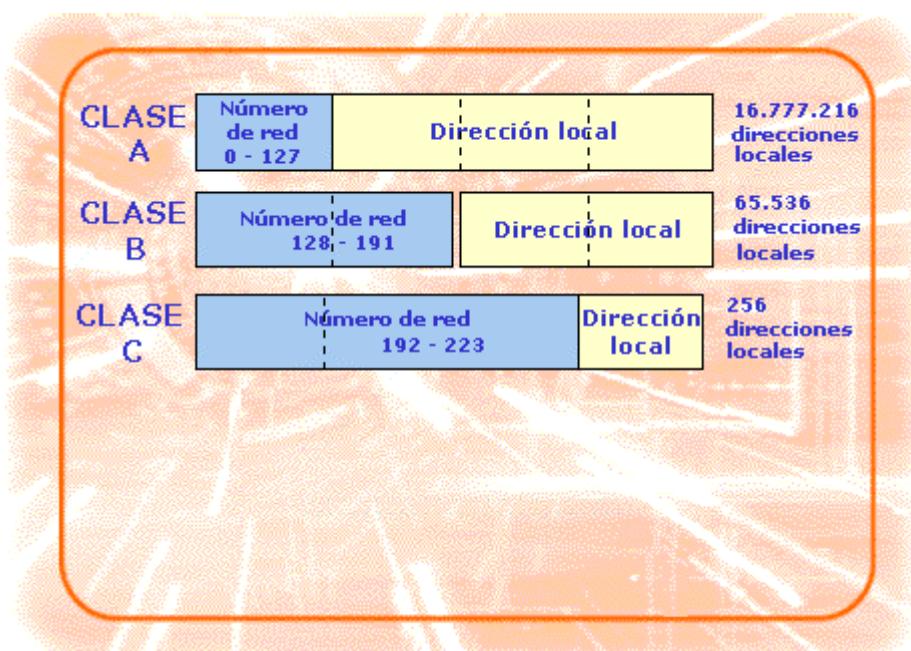
Todos los ordenadores deben tener una dirección IP única en el rango de los sistemas con los que se comunican. Es decir, dentro de una red no puede haber dos sistemas con la misma IP.

## Clases de direcciones IP

Toda organización que planee conectarse a la Internet debe conseguir un bloque de direcciones IP únicas.

Las direcciones se consiguen de la autoridad de registro apropiada.

Por conveniencia, las NIC de registro delegan grandes bloques de su espacio de direcciones IP a los proveedores de servicio. De esta forma, las organizaciones pueden obtener sus direcciones de sus proveedores de servicios en lugar de un NIC de registro.



Durante muchos años, sólo había tres tamaños de bloques de direcciones, grande, mediano y pequeño. Existían tres formatos diferentes de direcciones de red para cada uno de los tamaños de bloques. Los formatos de direcciones eran:

- Clase A, para redes muy grandes.
- Clase B, para redes de tamaño medio.
- Clase C, para redes pequeñas.

En los inicios de la Internet, a las organizaciones con redes muy grandes, como la Marina de los Estados Unidos o Digital Equipment Corporation, se les concedía direcciones de **Clase A**.

La parte de red de una dirección de Clase A tiene una longitud de un octeto. Los tres octetos restantes pertenecen a la parte local y se utilizan para asignar números a nodos.

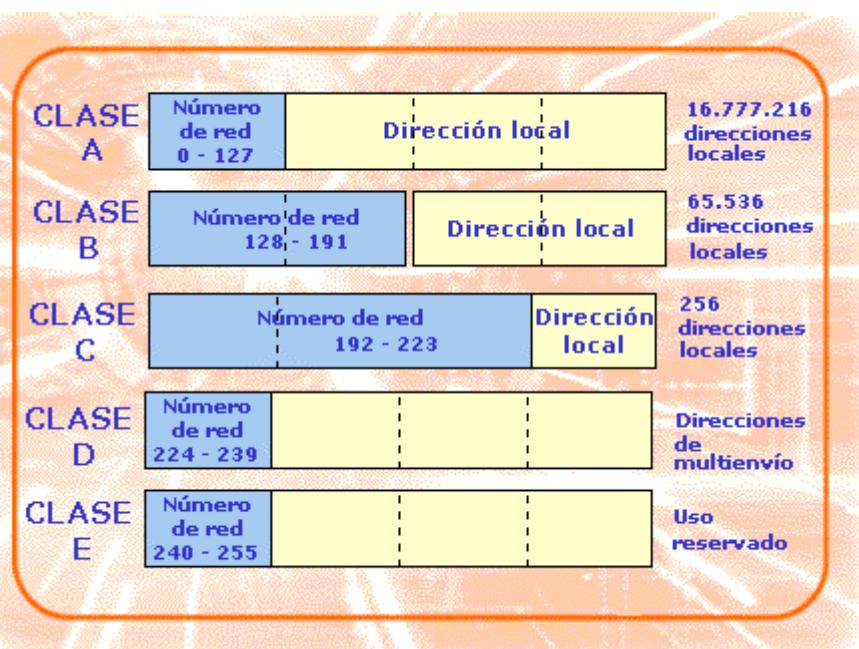
Existen muy pocas direcciones de Clase A, y la mayoría de las organizaciones de gran tamaño han tenido que conformarse con un bloque de direcciones de **Clase B**.

La parte de dirección de red de una Clase B es de dos octetos. Los dos octetos restantes de una dirección de Clase B pertenecen a la parte local y se utilizan para asignar números a los nodos.

Las organizaciones pequeñas reciben una o más direcciones de **Clase C**. Su parte de red son 3 octetos, con lo que, solamente, queda un octeto para la parte de nodo.

Es sencillo adivinar la clase de una dirección IP. Basta con mirar el primer número de la dirección en formato de puntos. Los intervalos de números para cada una de las clases se pueden ver en el gráfico.

Además de las Clases A, B y C, existen dos formatos especiales de direcciones, la Clase D y la Clase E.



Las direcciones de **Clase D** se utilizan para *multienvío* de IP. El multienvío permite distribuir un mismo mensaje a un grupo de ordenadores dispersos por una red. Las direcciones de multienvío permiten, por ejemplo, realizar aplicaciones de conferencia.

Las direcciones de **Clase E** se han reservado para uso experimental.

## Direcciones sin conexión a Internet

Se han reservado varios bloques de direcciones para su uso en redes que no se van a conectar a Internet y que no van a necesitar conectividad con otra organización. Estas direcciones son:

- 10.0.0.0 - 10.255.255.255.
- 172.16.0.0 - 172.31.255.255.
- 192.168.0.0 - 192.168.255.255.

Hay que tener en cuenta que puede haber *muchas* organizaciones que utilicen estos números. Si una compañía se fusiona con otra en algún momento, o decide comunicarse con los clientes o los proveedores mediante TCP/IP, puede haber conflictos de direcciones.

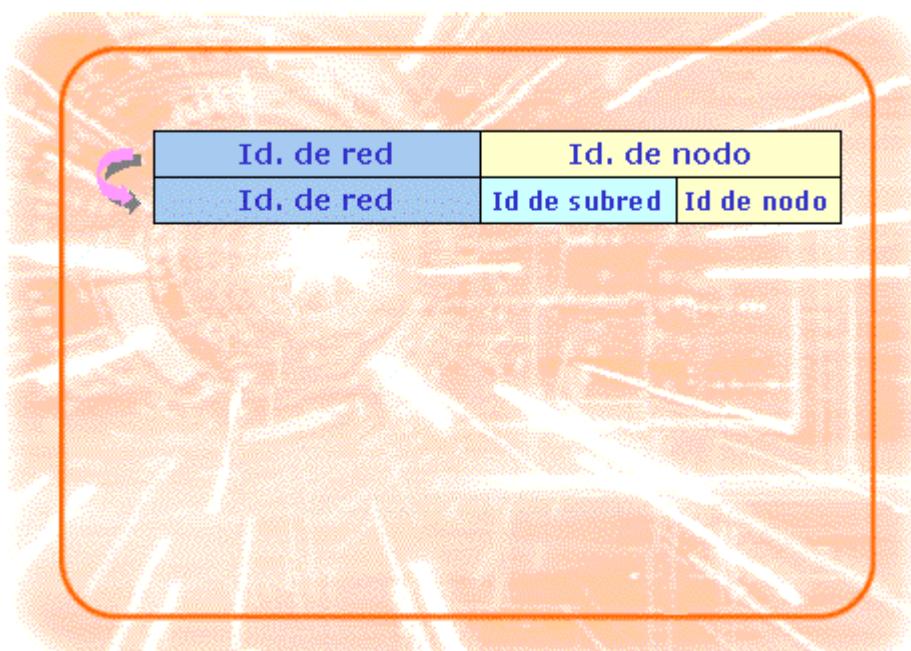
Sin embargo, puede registrar una red de Clase C y usarla para las comunicaciones externas. Se puede obtener software de envío que mande la información entre ciertos ordenadores y el mundo exterior a través de una red registrada de Clase C.

En la RFC 1918, *Address Allocation for Private Internets* (Asignación de direcciones en Internet privadas), se tratan las ventajas e inconvenientes de utilizar estas direcciones reservadas.

## Redes y Subredes de TCP/IP

Una organización que tenga direcciones de red de Clase A o Clase B es muy probable que tenga una red de cierta complejidad constituida por muchas LAN (Redes de Área Local) y varios enlaces a WAN (Redes de Área Amplia).

Tiene sentido, entonces, hacer la asignación de direcciones de forma que coincida con la estructura de la red, es decir, en función de las subredes existentes.



Para ello, la parte local de la dirección se divide en una **parte de subred** (situada inmediatamente después de los bits usados para identificar el tipo de red) y una **parte de nodo** (que como se ha dicho se usa para hacer referencia a una máquina concreta). El formato se muestra en el gráfico.

El tamaño de la parte de subred de una dirección y la asignación de números a subredes es responsabilidad de la organización que "posee" esa parte del espacio de direcciones.

Las organizaciones con direcciones de Clase B, como por ejemplo 128.21, suelen utilizar el tercer octeto para identificar subredes. Por ejemplo:

- 128.121.1
- 128.121.2
- 128.121.3

El cuarto octeto se utilizará para identificar los host particulares de una subred.

	<b>Id. de red</b>	<b>Id. de nodo</b>
	<b>Id. de red</b>	<b>Id de subred</b>
<b>CLASE B</b>	<b>Dirección de red</b>	<b>Dir de Subred</b>
	128.121	.50
		.145

Analicemos el caso de las organizaciones con direcciones de Clase C. Como sabemos, sólo tienen un octeto de espacio de direcciones (el resto de los bits es usado en la identificación de red).

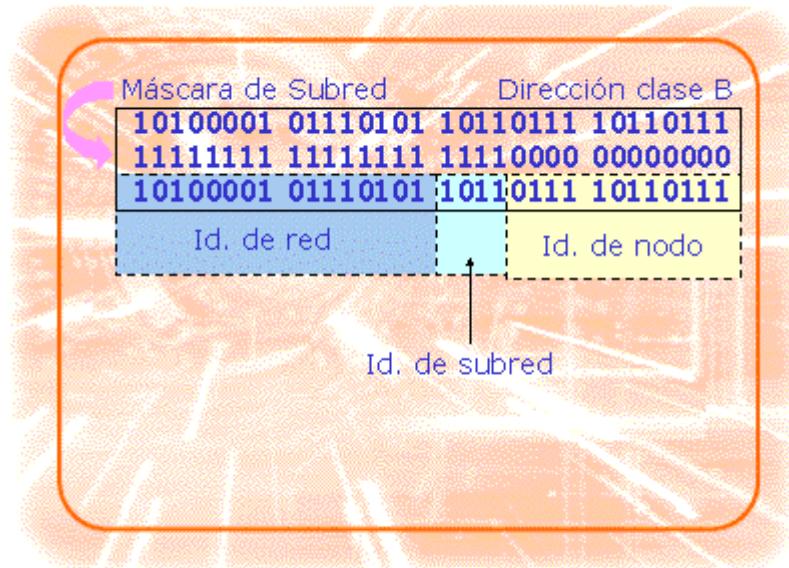
En este caso el administrador de red, podría elegir no realizar subredes o quizás usar 4 bits para el direccionamiento de subredes y los cuatro restantes para direcciones de host, como se muestra en el gráfico.

	<b>Id. de red</b>	<b>Id. de nodo</b>
	<b>Id. de red</b>	<b>Id de subred</b>
<b>CLASE B</b>	<b>Dirección de red</b>	<b>Dir de Subred</b>
	128.121	.50
		.145
<b>CLASE C</b>	<b>Dirección de red</b>	
	193.32.44	.61
	0011	1101

## Máscaras de subred

El tráfico de datos se encamina por tanto hacia un host específico en función de los campos de red y subred de una dirección IP.

Sabemos que la parte de red de una dirección de Clase A, B o C tiene un tamaño fijo.



Pero las organizaciones pueden decidir sus propios tamaños de subred, por lo que ¿cómo pueden reconocer los encaminadores estos campos?

La respuesta reside en que los sistemas se configuran de modo que reconozcan el tamaño de la parte de subred de la dirección IP.

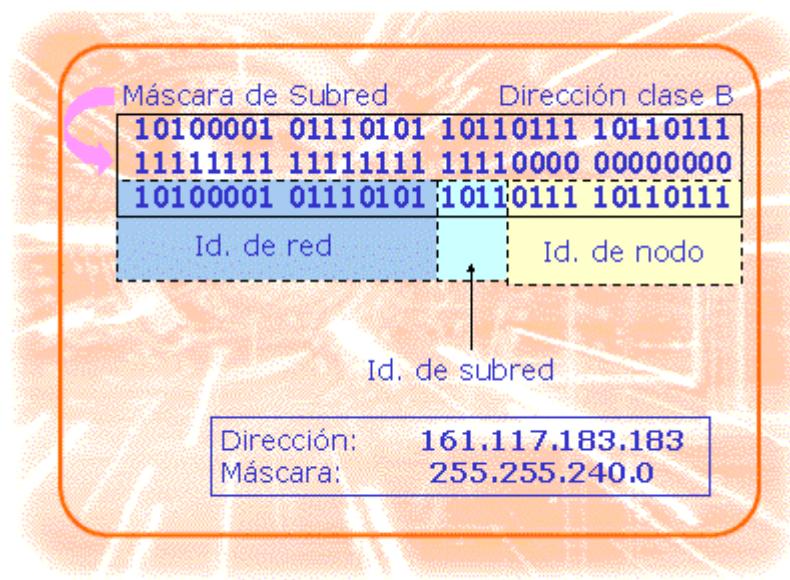
El tamaño del campo de subred se almacena en un parámetro de configuración denominado **máscara de subred**.

La máscara de subred es una secuencia de 32 bits. El administrador de red pondrá los bits que correspondan a los campos identificadores de red y subred a 1 y los usados para identificar al host específico a 0.

Por ejemplo, si se utilizan los cuatro primeros bits del tercer byte de las direcciones que empiezan por 161.117 para identificar las subredes, la máscara en binario sería:

11111111 11111111 11110000 00000000,

que expresada en decimal es: 255.255.240.0.



## Direcciones especiales reservadas

- ◆ **Identificación de redes y subredes**
- ◆ **Difusión en la subred local**
- ◆ **Difusión dirigida a una subred**
- ◆ **Difusión a redes**
- ◆ **Dirección interna**

No se puede asignar cualquier número a una subred o a un host. Por ejemplo, algunas direcciones se utilizan para difusión, mientras que otras están reservadas para su uso en las tablas de encaminamiento.

Una buena regla es la que dice: *aléjate de usar un bloque de ceros o un bloque de unos ni en el campo de subred ni en el campo de host*. Y también, no hay ningún número de red que sean todos ceros o todos unos.

Es apropiado usar el formato de dirección con puntos para referirse a una red. Consiste en completar la parte local de la dirección con ceros. Así, 5.0.0.0 identificaría una red de Clase A, 132.18.0.0 una de Clase B y 201.49.16.0 una de Clase C.

Con este formato de notación podemos identificar de forma inmediata las subredes. Por ejemplo, si sabemos que la red 131.18.0.0 usa máscara de red de 8 bits, 131.18.5.0 y 131.18.6.0 sería la forma en que haríamos referencia a dos de sus subredes. Este tipo de notación es el usado habitualmente en las tablas de encaminamiento.

Lógicamente, las direcciones con este formato no se pueden asignar a un host específico ni a un encaminador.

Un datagrama se puede difundir hacia los sistemas de un determinado ámbito. Existen varios patrones de direcciones IP utilizados para difusión.

La dirección 255.255.255.255, es decir, una IP con 32 unos, difunde un datagrama a todos los sistemas en el enlace local. Este tipo de difusión se usa, por ejemplo, con los protocolos BOOTP y DHCP, que un host utiliza para obtener su dirección IP y otros datos de inicialización desde un servidor de arranque. Un cliente envía una solicitud de arranque a 255.255.255.255 y usa la dirección reservada 0.0.0.0 como su dirección IP.

Una difusión se expande por la LAN envolviendo el datagrama IP en una trama cuya cabecera tiene la dirección de difusión física "a-todos" como dirección de destino.

La difusión se puede dirigir hacia la subred a la que pertenece el equipo origen o hacia una subred remota. Por ejemplo, si 131.18.7.0 es una subred de una Clase B, un equipo origen ubicado en ella, puede utilizar la dirección **130.18.7.255** para difundir un mensaje a todos los demás nodos de dicha red. Si la subred de destino es remota, el resultado de enviar un datagrama a la dirección de difusión es que se transmitirá una copia del datagrama al encaminador conectado a la subred 131.18.7.0. Suponiendo que fuera una LAN, el encaminador debería usar una dirección de difusión física en el campo de destino de la trama MAC para dirigir el mensaje a todos los host de dicha subred. Esto implica que no se podría asignar a ningún sistema la dirección reservada 130.18.7.255.

Se puede enviar un datagrama a todos los host de una red remota dada. Basta poner el campo local de la dirección a uno. Por ejemplo, si un administrador quisiera enviar un aviso a todos los nodos de la red Ethernet de Clase C 201.49.16.0. La dirección Ip que usaría para la difusión sería **201.49.16.255**. Lógicamente, no se puede asignar a ningún host la dirección 201.49.16.255. Del mismo modo, la dirección 131.18.255.255 se podría usar para enviar un mensaje a todos los nodos de la red de Clase B 131.18.0.0. Por lo dicho, tampoco se puede asignar el número 255 como número de subred, ya que los encaminadores no sabrían si se quiere hacer una difusión hacia esa subred concreta o hacia la red completa. Para evitarlo, nunca se asigna a una subred un número que sea todo unos.

En el extremo opuesto de la difusión están los mensajes que nunca abandonan el host local. Esto es habitual, por ejemplo, en los host que contienen procesos clientes y servidores. Los clientes y servidores locales se comunican por IP sin que los datos salgan del host. Para ello, utilizan una dirección especial que se llama dirección interna (o dirección de loopback). Por convenio, cualquier dirección que empieza por 127 se reserva para este propósito. En la práctica, sólo se usa la dirección 127.0.0.1. Ten en cuenta que se ha reservado un espacio de direcciones de Clase A de  $2^{24}$  números con este propósito.

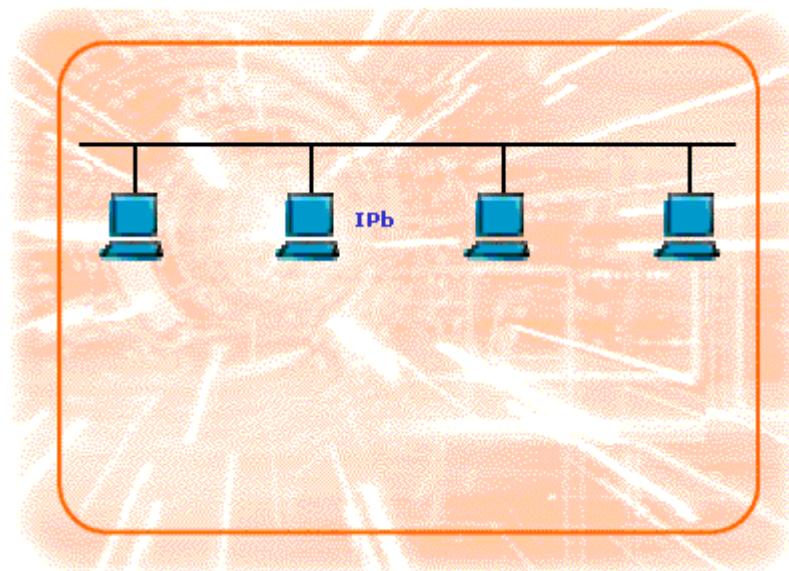
## 2- Protocolo ARP

ARP es algo así como un traductor. Te explicaremos en detalle en que consiste este protocolo.

### Protocolo ARP

#### Protocolo ARP ("Address Resolution Protocol")

Para enviar un paquete de una máquina a otra, el software de red debe transformar la dirección IP en una dirección física (MAC).



#### Solución:

- Si la dirección MAC es menor que la dirección IP, se puede establecer una transformación directa.
- Utilizar el protocolo **ARP**, el cual realiza la transformación (traducción) dinámica de las direcciones y permite que un nodo encuentre la dirección física de otro, dentro de la misma red, con sólo proporcionar la dirección IP destino.

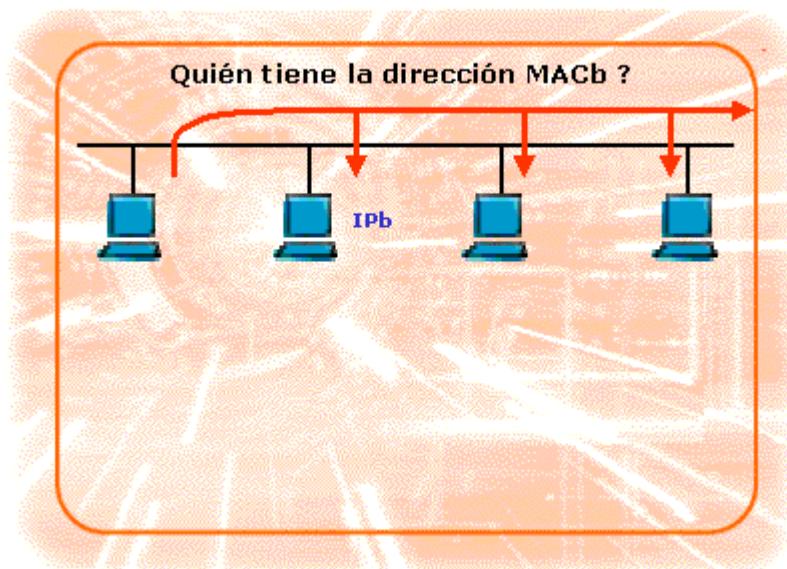
## Características

- Los mensajes ARP van contenidos directamente en una trama física (Ethernet, Token Ring, etc.).
- El mensaje ARP depende de la tecnología de red subyacente.
- Se utiliza broadcast para localizar la máquina destino.
- Se generan tablas temporales (tablas ARP, de las que más adelante hablaremos), con los datos registrados cara evitar sobrecargas en la red.
- El campo long. de la trama Ethernet es igual a 0806<sub>16</sub>.

## Funcionamiento

Los hosts de la red local utilizan ARP para descubrir información sobre su propia dirección física.

Cuando un host quiere empezar a comunicarse con un socio local busca la dirección IP del otro en su tabla ARP, que normalmente se mantiene en memoria.

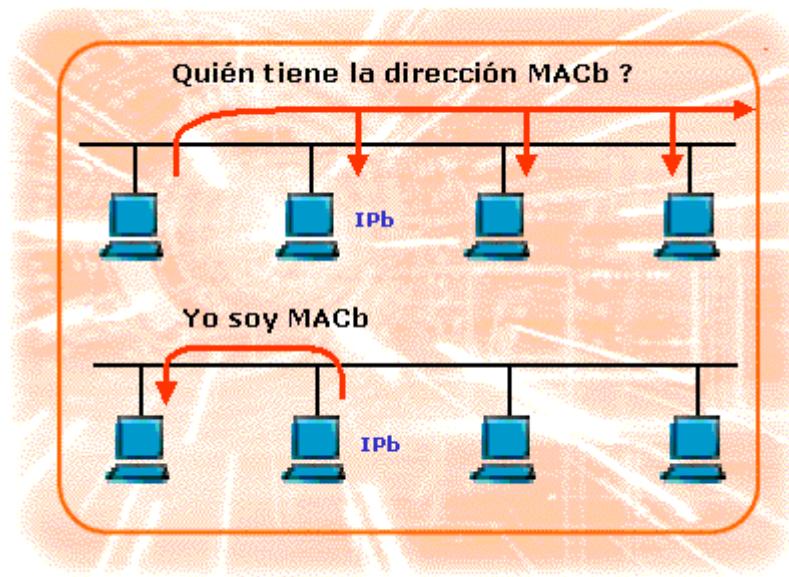


Si no existe una entrada para esa dirección IP, el host difunde una solicitud de ARP que contiene la dirección IP de destino, de acuerdo con la figura.

El host de destino reconoce su dirección IP y lee la consulta.

Lo primero que hace el host destino es actualizar su propia tabla ARP con la dirección física del origen.

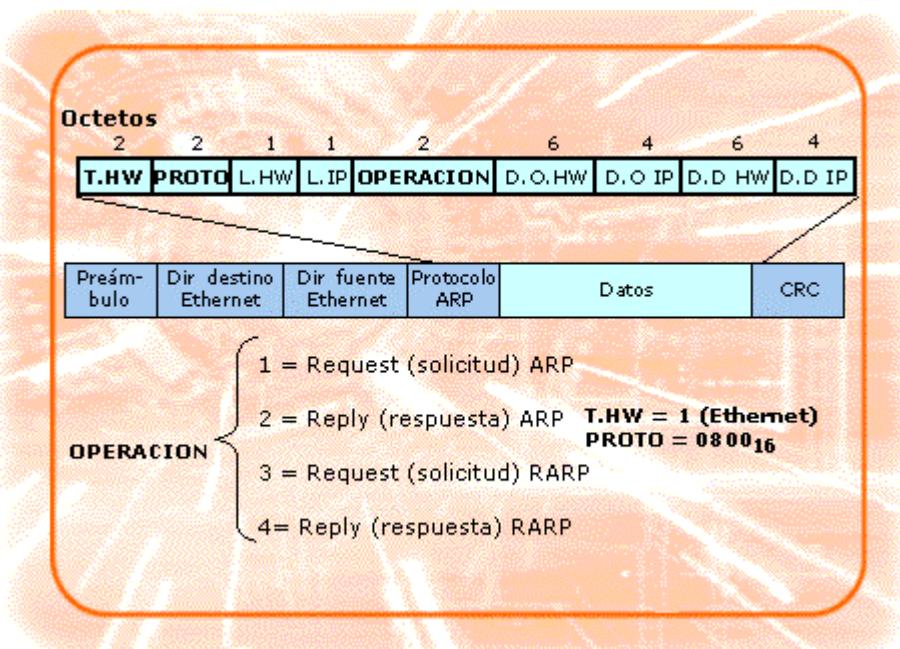
Es lógico, ya que, probablemente, el destino pronto empezará una conversación con el origen.



El host destino envía de vuelta una respuesta que contiene su propia dirección de la interfaz hardware.

Cuando el origen recibe la respuesta, actualiza su tabla ARP y ya está listo para transmitir datos por la LAN.

## Formato de trama ARP



ARP se usó inicialmente en LANs Ethernet, pero su diseño es general, por lo que se puede generalizar su uso a otros tipos de redes, como Token-Ring, FDDI y redes de área extensa SMDS (Servicio de datos multimegabit). Se ha diseñado una variante de ARP para su uso con circuitos virtuales de área extensa (como por ejemplo Frame Relay).

La figura muestra el formato de trama ARP para redes Ethernet.

### **Tipo de Hardware**

Especifica la tecnología de red con la que estamos trabajando (= 0x01, Ethernet).

### **Tipo de Protocolo**

Especifica el tipo de protocolo de nivel de red para el que se están resolviendo las direcciones físicas (= 0x0800, Protocolo IP).

### **Longitud Hardware**

Especifica la longitud de la dirección física que se va a resolver (= 0x06, longitud de la dirección Ethernet). Este campo permite a ARP la resolución de diferentes tipos de direcciones físicas, no sólo Ethernet (Token ring, por ejemplo).

### **Longitud IP**

Especifica la longitud de la dirección del protocolo de nivel 3 que hay que traducir (= 0x04, Protocolo IPv4). Este campo permite a ARP ser utilizado por otros protocolos de nivel 3 (IPv6, IPX, etc.).

## Operación

1 = Solicitud de resolución de una dirección de nivel 3.

2 = Respuesta con la resolución solicitada.

## Dirección Origen Hardware

Especifica la dirección física del emisor de la trama ARP.

## Dirección Origen IP

Especifica la dirección de nivel 3 del emisor de la trama ARP.

## Dirección Destino Hardware

Especifica la dirección física de destino de la trama ARP (dirección de difusión física si se trata de una Operación de Solicitud).

## Dirección Destino IP

Especifica la dirección de nivel 3 del destino de la trama ARP.

## Tabla ARP

Siempre que un host recibe una respuesta ARP, guarda la dirección IP del transmisor, así como la dirección física correspondiente, en su memoria intermedia, para utilizarla en búsquedas posteriores.

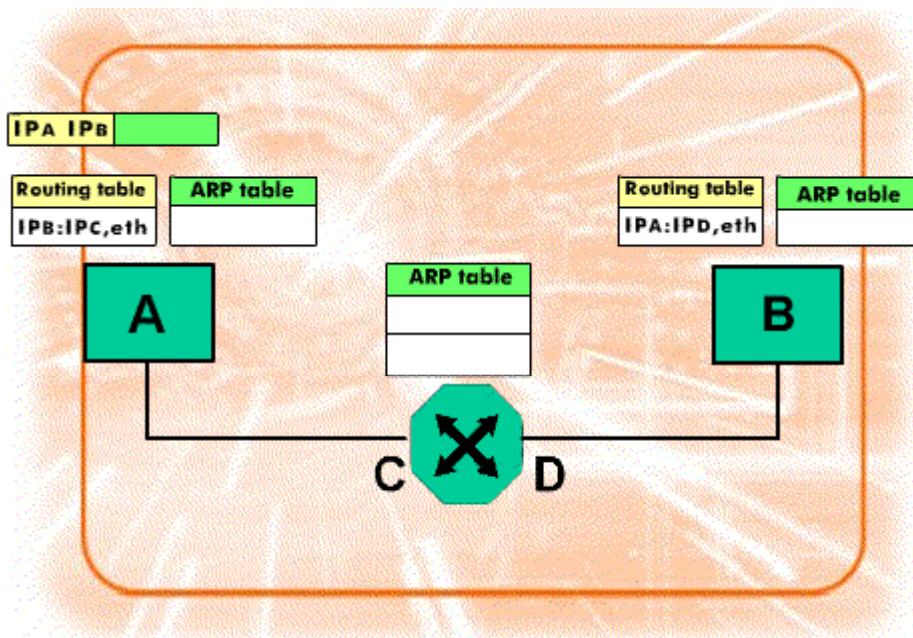
Antes de trasmisir un paquete, lo primero que hace el host es buscar, en su memoria intermedia, la asignación que necesita antes de enviar una solicitud ARP. Si encuentra la asignación deseada en su memoria intermedia, ARP no necesita transmitir una difusión a la red. La tabla ARP es dinámica; pasado un tiempo sin que una entrada de la tabla haya sido utilizada, dicha entrada es borrada de la tabla. Así, los host se protegen ante posibles cambios en las direcciones físicas del resto de host de la LAN.

Por ejemplo, uno de los equipos de la LAN podría reemplazar su interfaz de red (la tarjeta de red) por otro, lo que conllevaría un cambio en su dirección física (aunque no en su dirección IP).

Este host debería informar a los demás host de la LAN, de que su dirección física ha cambiado cara a que actualizaran su tabla ARP. Pero si no lo hace, pasado un tiempo, como los host van borrando las entradas no utilizadas de sus tablas, se encargarán ellos mismos de la actualización de sus tablas al difundir nuevas peticiones ARP.

Habitualmente, los sistemas operativos proporcionan comandos para mostrar y manipular la tabla ARP asociada al propio ordenador.

## Mecanismo ARP Proxy



### Mecanismo ARP Proxy (destino en red diferente)

Supongamos que tenemos un "host" A que quiere enviar paquetes IP a otro "host" B que no se encuentra en su red local. El hecho de que los hosts estén ubicados en subredes distintas implica que se hará uso del encaminador CD (que podría ser por ejemplo un router).

Supongamos también que, inicialmente, las tablas ARP de todos los sistemas están vacías.

A chequea su tabla de enrutamiento y "descubre" que B está conectado al otro lado de la interfaz Ethernet Ipc.

Ipc es la dirección a través de la que los equipos de la subred en que está ubicado A pueden acceder al encaminador CD.

Para enviar datos (datagramas IP) al "siguiente salto", en este caso la dirección Ipc, el "host" A debe conocer la dirección MAC del interfaz C. Para ello, manda un ARP-request.

El encaminador CD reconoce su dirección IP y contesta con un ARP-reply. Además, CD aprovecha para actualizar su tabla ARP con la información de A.

Una vez recibida la información de CD, el "host" A la coloca en su tabla ARP y comienza a enviar datagramas IP, de A a B, usando como paso previo al encaminador CD.

Cuando CD intenta enviar los datagramas que está recibiendo hacia B se da cuenta de que no conoce la dirección MAC de su interfaz (aunque, eso sí, sabe que lo tiene a su lado derecho). Lo que hace entonces es difundir hacia la red derecha un ARP-request solicitando la dirección MAC de B.

El "host" B reconoce su dirección IP y contesta con un ARP-reply. Además aprovecha para actualizar su tabla ARP con la información proveniente de CD.

Con la dirección MAC de B añadida a su tabla ARP, el encaminador CD ya puede enviar los datagramas IP hacia el equipo destino.

Supongamos que B decide enviar una respuesta a A. Para ello, mira en su tabla de enrutamiento y descubre que, si quiere alcanzar a A, tiene que hacerlo a través del encaminador con dirección IPd.

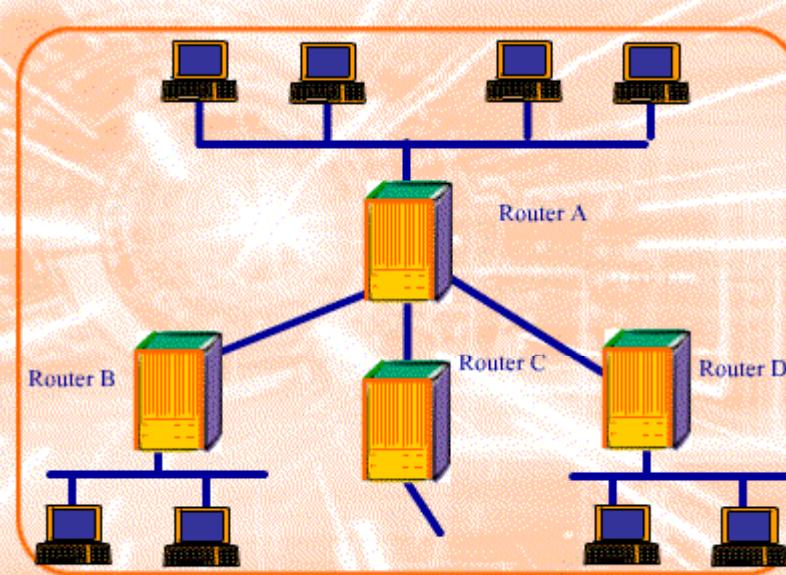
Entonces B chequea su tabla ARP y descubre que hay una entrada en dicha tabla para IPd, por lo que puede empezar a enviar datagramas IP hacia el interfaz D del encaminador CD.

El encaminador CD (normalmente un router) también encuentra la dirección MAC de A en su tabla ARP, con lo que puede enviar directamente los datagramas hacia su destino, el hosts A.

### 3- Direcciones de multienvío

¿Qué dirección de destino usaríamos si quisieramos enviar un mensaje a todos los usuarios?.

#### Direcciones de multienvío



La difusión de IP sirve poder enviar datagramas hacia todos los sistemas de una red o subred. Una forma de envío múltiple más restringido es el **multienvio IP (o multidifusión IP)**, que **sirve para propagar datagramas hacia grupos concretos de sistemas**, como se muestra en el gráfico.

El multienvio de IP puede ser una herramienta de red muy útil. Por ejemplo, se puede utilizar un único mensaje para actualizar simultáneamente datos de configuración de un grupo homogéneo de host o para solicitar a un grupo de host su estado.

El multienvio también es la base de muchas aplicaciones nuevas que permiten a los usuarios "conectarse" a conferencias.

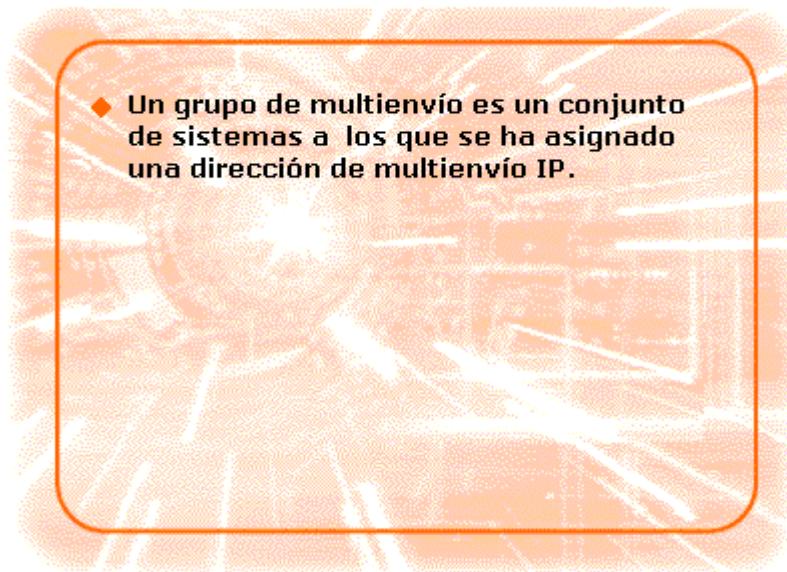
Para el multienvio **se utilizan las direcciones de Clase D**.

Se ha definido una norma del protocolo de multienvio, pero el número de host y encaminadores que admiten la norma actualmente es limitado. Sin embargo, su uso se extenderá en los próximos años, por lo que tiene interés ver alguna de sus características.

## Grupos de multienvío

Un grupo de multienvío es un conjunto de sistemas a los que se ha asignado una dirección de multienvío IP.

Los miembros del grupo siguen manteniendo su propia dirección IP, pero tienen la capacidad de recoger los datos enviados a su dirección de multienvío.



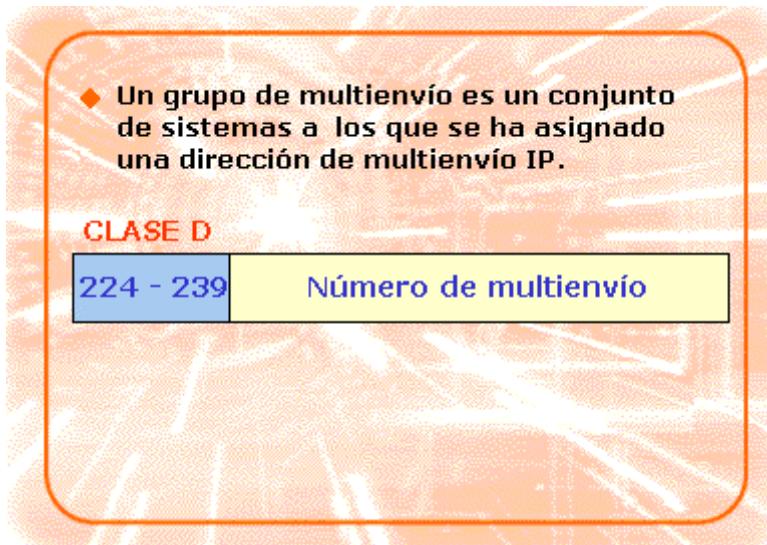
La pertenencia a un grupo de multienvío IP es un proceso dinámico.

Un host puede unirse o abandonar un grupo en cualquier momento.

Además, un host puede ser miembro de un número indeterminado de grupos de multienvío.

Las direcciones de Clase D que se usan para multienvío comienzan con los números en el intervalo 224 a 239.

Algunas direcciones de multienvío IP son permanentes (bien conocidas) y su lista está en la RFC Assigned Numbers de Internet.



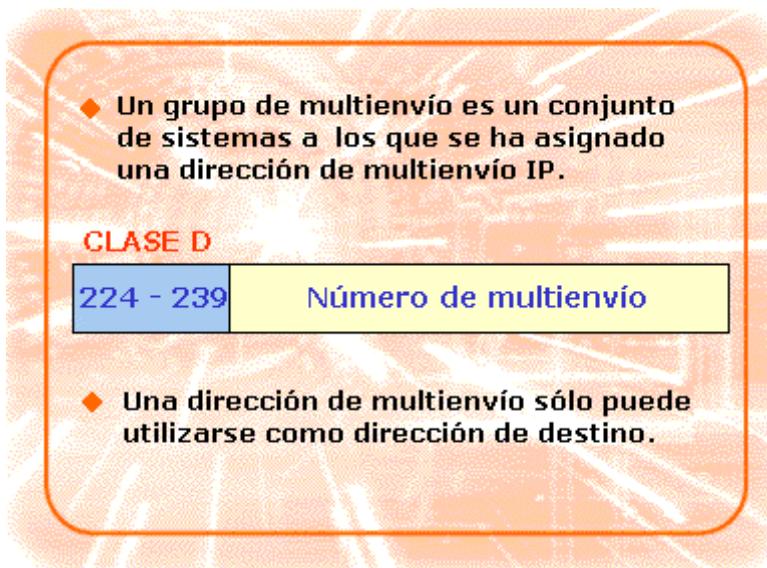
Algunas direcciones permanentes de multienvío IP definidas son:

- 224.0.0.1 Todos los miembros de una subred local.
- 224.0.0.2 Todos los encaminadores de una subred local.
- 224.0.0.5 Todos los encaminadores que admitan el protocolo OSPF.

Las direcciones de multienvío también se asignan de una manera ad hoc a los grupos temporales que se forman y disuelven según se necesitan, por ejemplo, para una conferencia con audio y vídeo.

Las direcciones de multienvío IP sólo pueden emplearse como direcciones de destino.

Éstas nunca podrán aparecer en el campo de dirección fuente de un datagrama.



Además, no se generan mensajes de error ICMP relacionados con datagramas de multienvío (no te preocupes, más adelante te hablaremos en detalle del protocolo ICMP, baste decir ahora que sirve para notificar errores en la transmisión).



## Extensión de IP para manejar el multienvío

Las modificaciones que permiten a un host enviar multidifusión IP no son complejas:

- El software IP debe permitir a un programa de aplicación especificar una dirección de multienvío como una dirección IP de destino,
- y el software de interfaz de red (Ethernet, por ejemplo) debe ser capaz de transformar una dirección de multidifusión IP en la correspondiente dirección de multidifusión hardware (o utilizar la difusión, si el hardware no soporta el multienvío).

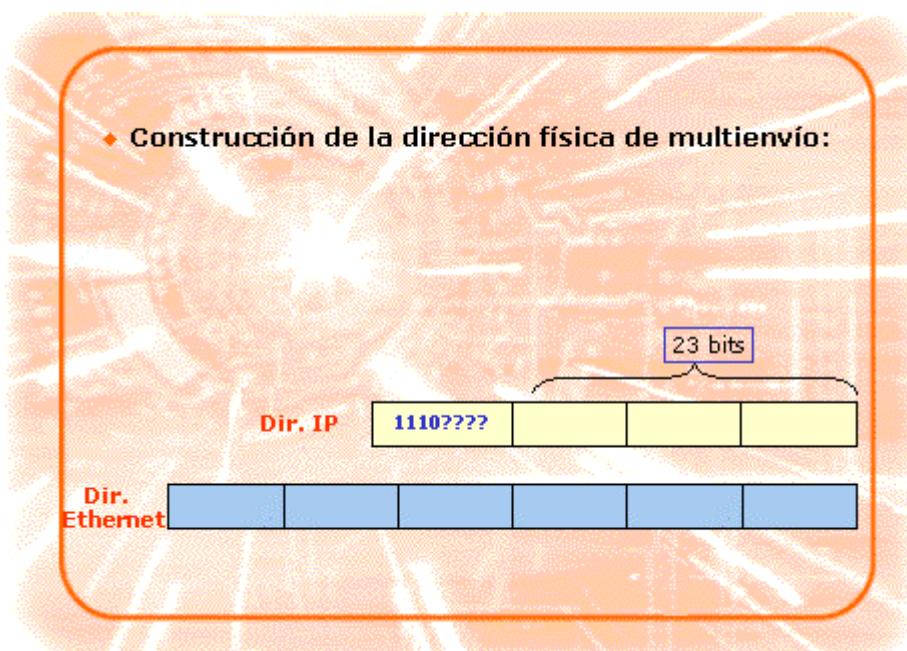
Ampliar el software del host para recibir datagramas de multienvío IP es más complejo:

- El software de IP en el host debe tener una interfaz que permita a un programa de aplicación declarar si desea unirse o abandonar un grupo de multidifusión.
- Si diversos programas de aplicación se unen al mismo grupo, el software de IP debe recordar cada uno de ellos para transferir una copia de los datagramas que llegan destinados a ese grupo.
- Si todos los programas de aplicación abandonan un grupo, el host debe recordar que no quedan participantes en el grupo.
- El host tiene que ejecutar un protocolo que informe a los encaminadores de multienvío locales del estado de los miembros de un grupo.

## Traducción de direcciones de multienvío IP a direcciones Ethernet

Opcionalmente, se puede asignar una o más direcciones de multienvío a las interfaces físicas de las LAN Ethernet (o FFDI). Se trata de asociaciones lógicas y se puede seleccionar cualquier valor conveniente. De esta forma, resulta sencillo traducir direcciones IP de multienvío a direcciones físicas de multienvío.

En las LAN Ethernet y FFDI se puede usar la siguiente regla de traducción:

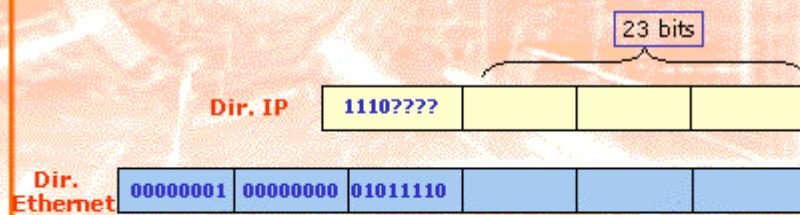


Los tres primeros octetos de la dirección física de multienvío deberían ser:

01-00-5E (...)

◆ Construcción de la dirección física de multienvío:

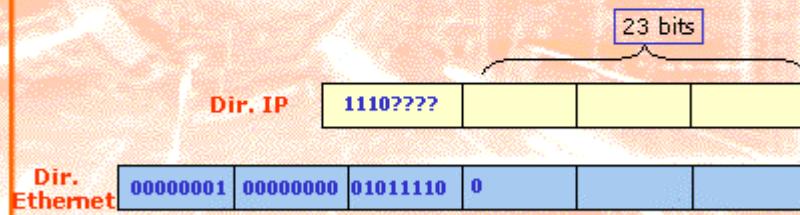
- Tres primeros octetos: 01-00-5E



(...) el bit siguiente debería ser un 0, (...)

◆ Construcción de la dirección física de multienvío:

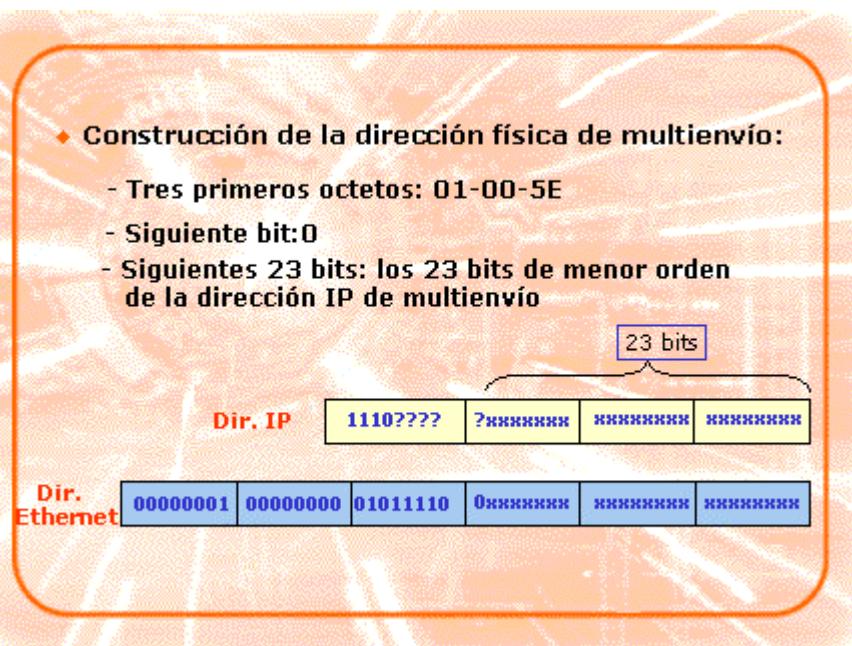
- Tres primeros octetos: 01-00-5E
- Siguiente bit: 0



(...) y los 23 bits finales deberían ser los 23 bits de menor orden de la dirección de multienvío de IP.

Esto significa, por ejemplo, que las tres direcciones de multienvío IP:

- 224.17.17.17      11100000 00010001 00010001 00010001
- 224.145.17.17     11100000 10010001 00010001 00010001
- 225.145.17.17     11100001 10010001 00010001 00010001



se traducirán, todas ellas, a la misma dirección física de multienvío:

00000001 00000000 01011110 00010001 00010001 00010001

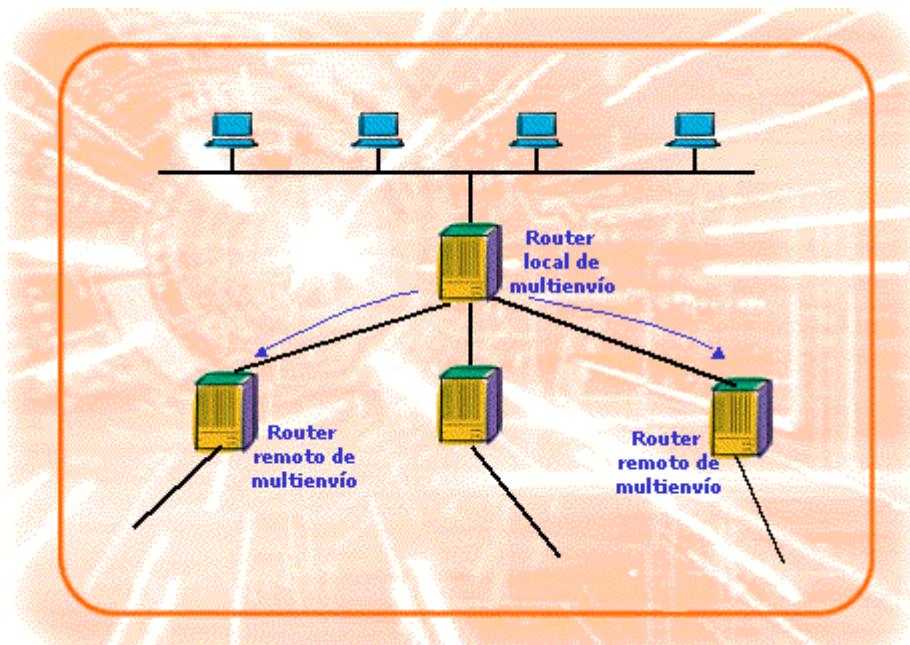
Las interfaces de los sistemas que pertenezcan a cualquiera de los tres grupos capturarían los multienvíos de todos los grupos. Sin embargo, la capa IP de los host descartará cualquier multienvío extraño.

Una buena forma de evitar este proceso es elegir direcciones de multienvío que tengan ceros en las posiciones. De esta forma, siguen quedando  $2^{23}$ , más de 8 millones, de direcciones de multienvío.

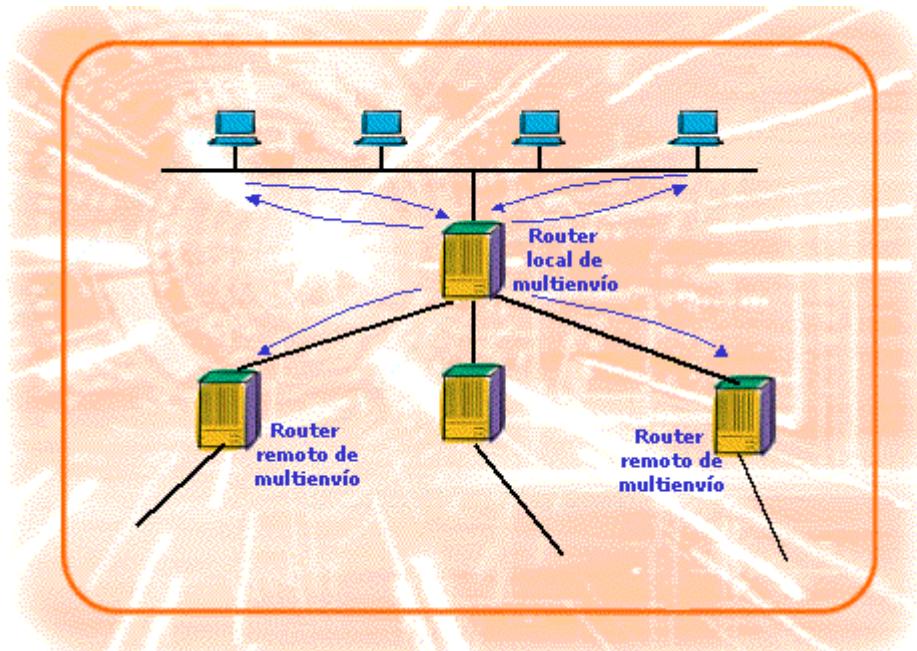
## Protocolo de gestión de grupos de Internet (IGMP) (I)

El multienvío no está restringido a una red local.

Los encaminadores (generalmente routers), con software de multienvío, son capaces de propagar los datagramas IP de multienvío a otros sistemas en Internet.

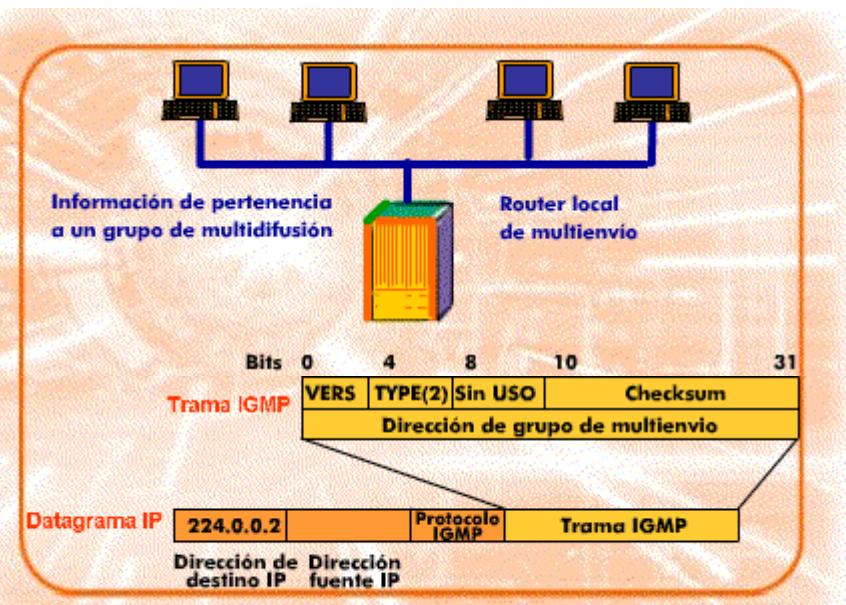


Para poder hacerlo eficientemente, el encaminador necesita conocer si existen host en las redes conectadas localmente que pertenecen a un grupo concreto de multienvío.



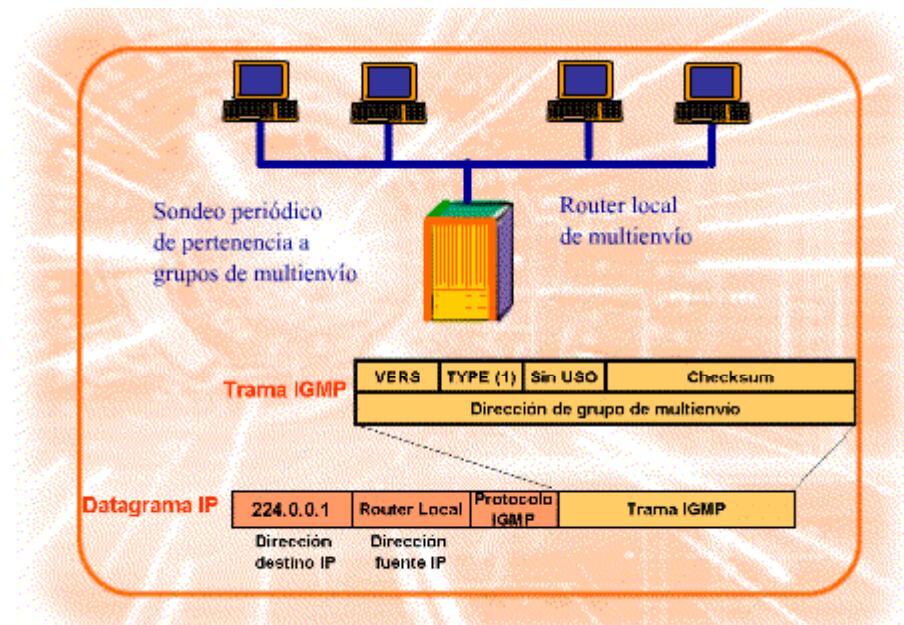
También necesita intercambiar información con otros encaminadores para poder descubrir si existen miembros del grupo que se encuentren en redes remotas a las que se deben reenviar los datagramas de multienvío.

## Protocolo de gestión de grupos de Internet (IGMP) (II)



Los host utilizan el *Protocolo de gestión de grupos de Internet* (IGMP - *Internet Group Management Protocol*) para informar de su pertenencia a un grupo a los encaminadores vecinos que admiten encaminamiento con multienvío. Esta información, es decir, los mensajes IGMP, se envían desde los hosts hacia los encaminadores, encapsulados en datagramas IP que tienen como dirección de destino 224.0.0.2 (es decir, "a todos los encaminadores de la subred local").

## Protocolo de gestión de grupos de Internet (IGMP) (III)



Para asegurarse de que la información de pertenencia es completa, el protocolo IGMP permite que los encaminadores sondeen periódicamente a los host, pidiéndoles un informe de los grupos actuales a los que pertenecen. Estas peticiones se envían encapsuladas en datagramas IP con dirección de destino 224.0.0.1 (es decir, "a todos los host").

### 3- Protocolo Internet (IP)

Como ya hemos mencionado, una **internet** es un conjunto de redes interconectadas con encaminadores, y el **Protocolo de Internet (IP)** es un protocolo de la capa de red que encamina los datos por una internet. Los investigadores y diseñadores que crearon IP respondían a los requisitos del Departamento de Defensa (DOD) de Estados Unidos de **crear un protocolo que pudiese:**

- Utilizarse en host y encaminadores de distintos fabricantes.
- Seguir el crecimiento de distintos tipos de redes.
- Permitir que la red crezca sin interrumpir el servicio.
- Admitir sesiones de nivel superior y servicios orientados a mensajes.

La arquitectura de la capa de red de IP se diseñó para cubrir estas necesidades.

Resultó que IP también daba a los creadores de redes exactamente lo que necesitaban para integrar las **redes de área local (LAN)** que se habían extendido por sus organizaciones como islas. Más aún, las nuevas islas se podían conectar sin tener que interrumpir las que ya existían.

Estas características hicieron que IP se convirtiese en el protocolo de red elegido para las agencias gubernamentales, universidades y empresas.

## 1- Protocolo IP

Aprende como funciona el protocolo que se encarga del direccionamiento en la red de redes.

### Características de IP

- ◆ **Datagramas IP**
- ◆ **Filosofía best effort**
- ◆ **Encaminamiento adaptativo**
- ◆ **MTU, fragmentación y reensamblado (I)**
- ◆ **MTU, fragmentación y reensamblado (II)**

IP dispone de ciertas características que contribuyen a dotarle de flexibilidad y capacidad para adaptarse a muchos entornos diferentes.

El protocolo IP ofrece los mecanismos necesarios para transportar unidades, denominadas **datagramas IP** por una Internet. Un datagrama IP está constituido por una cabecera IP y un trozo de datos a entregar.

IP sigue la filosofía del **Best Effort (mejor esfuerzo)**. Esto significa que IP no garantiza que el datagrama se entregue a su destino, pero sí que se hará lo mejor que se pueda. Un datagrama se puede destruir en el camino debido a: errores en los bits durante su transmisión por el medio, que un encaminador congestionado lo descarte debido a falta de espacio en el búfer, que no haya temporalmente camino hasta el destino.

Todas las funciones que aseguran la fiabilidad se han concentrado en la capa TCP, IP se desentiende de este asunto. La recuperación de datos destruidos es responsabilidad de TCP.

Normalmente, **el encaminamiento de los datagramas es adaptativo**. Es decir, en todo momento se realiza la mejor elección para el siguiente salto comprobando la tabla de encaminamiento del nodo actual. Conviene reseñar que las entradas de la tabla de encaminamiento pueden cambiar en cualquier momento en función de las condiciones de la red.

Antes de transmitir un datagrama por un salto de red, debe encapsularse por las cabeceras de la capa 2, en función de la tecnología de red subyacente. Pero, cada tecnología de LAN y WAN impone límites diferentes al tamaño de las tramas. Un datagrama debe "caber" en una trama, por lo que el tamaño máximo de la misma restringe el tamaño de los datagramas que se pueden enviar por un determinado medio.

El mayor tamaño de un datagrama por un medio específico se denomina Unidad máxima de transmisión (*Maximum Transmission Unit*), o **MTU** (802.3 tiene una MTU de 1492 octetos).

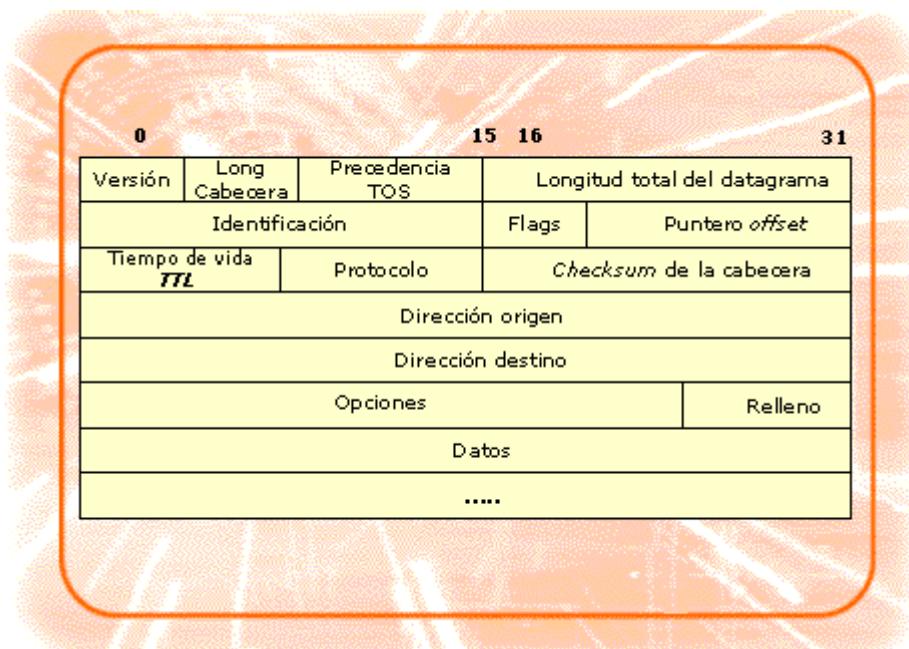
En una Internet grande, un host origen puede que no conozca los límites de tamaño con que se va a encontrar un datagrama en el camino. ¿Qué ocurre si el host origen ha enviado un datagrama que es demasiado grande para un nodo intermedio?

Cuando llega el datagrama al encaminador conectado a la red intermedia, IP resuelve el problema de tamaño dividiendo el datagrama en varios datagramas menores llamados fragmentos. Es responsabilidad del IP del host de destino recoger los fragmentos y reconstruir el datagrama original.

## Formato de trama

Como ya se ha mencionado, IP proporciona un servicio no orientado a conexión (servicio de datagrama).

El gráfico muestra una visión general de la estructura de un datagrama IP.

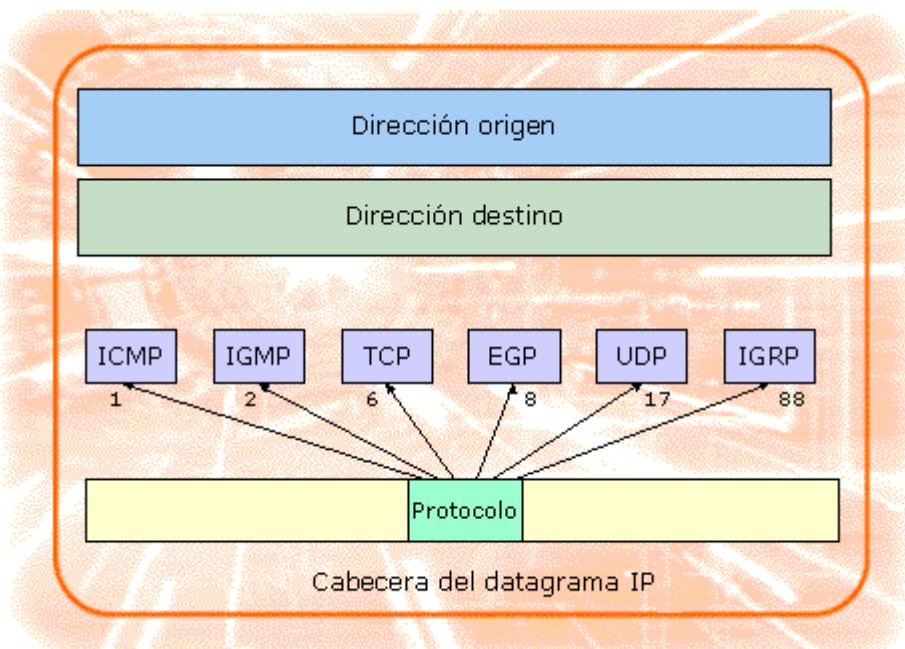


Consta de la "cabecera-IP" y datos provenientes de niveles superiores.

La cabecera IP consta de, al menos, 20 octetos (desde el campo *Versión* hasta *Dirección Destino*).

El estándar de red para la ordenación de los bytes especifica que, en los campos codificados en binario, los bits más significativos se envían primero.

## Campos de trama destino, origen y protocolo



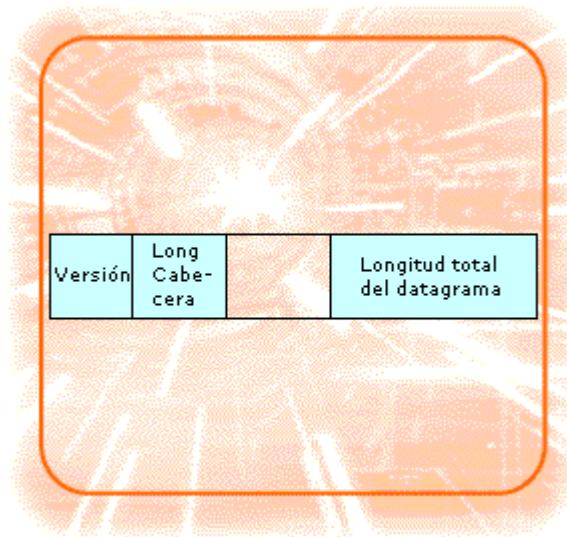
Los campos más importantes de la cabecera son los de Dirección IP de destino, Dirección IP de origen y Protocolo.

La **dirección IP de destino** permite a IP encaminar el datagrama. Cuando el datagrama llega a su host de destino, el campo **Protocolo** permite identificar el servicio concreto al que va dirigido el datagrama, como por ejemplo TCP o UDP.

Existen otros protocolos además de TCP o UDP que envían y reciben datagramas.

La Autoridad de asignación de números de Internet (IANA - *Internet Assigned Numbers Authority*) es la responsable de coordinar la asignación de valores a los parámetros de TCP/IP, entre ellos, los valores que se pueden usar en el campo Protocolo de IP.

## Campos de trama versión, longitud de cabecera y longitud total del datagrama



El campo **versión** indica el formato de la cabecera IP. La versión actual es **4 (0100)** en binario). El propósito de este campo es permitir la evolución del protocolo. En la actualidad, IPv6 ya está definido, pero tendrá que pasar algún tiempo hasta que este nuevo protocolo sea usado.

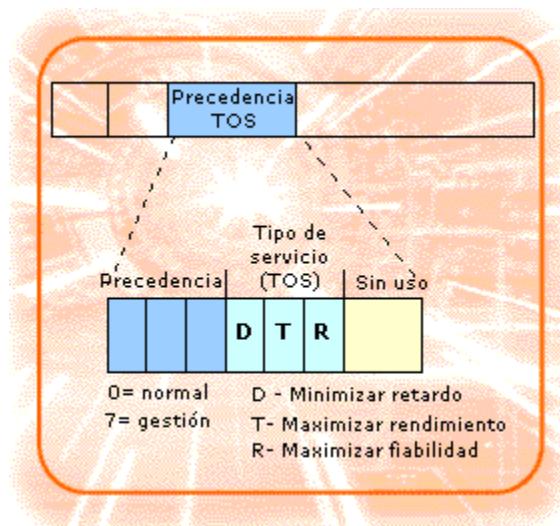
El campo **cabecera** (IHL, "Internet Header Length") especifica la **longitud de la cabecera** IP en unidades de palabras de 32 bits. El tamaño del campo es 4 bits, lo cual limita el rango hasta 1532 bits o 60 octetos. Fíjate que el valor mínimo para una cabecera correcta es 5, o 1010 en binario.

El campo **longitud total de datagrama** especifica la longitud total del datagrama, medido en octetos, incluyendo la cabecera y los datos.

Uno de los propósitos de Internet es ocultar las topologías de red subyacentes. Cuando un datagrama viaja de encaminador en encaminador, puede atravesar diferentes redes. Estas redes pueden soportar diferentes tamaños máximos de datagramas, por lo que se debe escoger un tamaño adecuado de datagrama y prever algún método que permita la fragmentación en paquetes de menor tamaño.

Internet no limita los datagramas a un tamaño específico, pero sugiere que ambos, redes y encaminadores, deberían estar preparados para manejar datagramas a partir de 576 octetos (por ejemplo, un bloque de datos de 512 octetos y 64 octetos de cabecera).

## Campo de precedencia y tipo de servicio



### Precedencia

Los bits que configuran la Precedencia fueron pensados para dar prioridad a determinados datagramas, pero han sido poco usados en aplicaciones no gubernamentales o militares. En la actualidad la situación está cambiando.

La norma de IP no obliga sobre las acciones concretas que se pueden seguir de los valores de los bits de precedencia.

### Tipo de Servicio

Los bits Tipo de Servicio (TOS - Type of Service) determinan la calidad de servicio, lo que podría afectar al manejo de los datagramas, siendo éste el caso por ejemplo de un encaminador que se queda sin memoria y tiene que descartar algunos datagramas.

Así, un encaminador puede tener en cuenta que un datagrama con el bit de fiabilidad a uno, sea menos elegible para descartarlo que uno que tiene el bit de fiabilidad a cero.

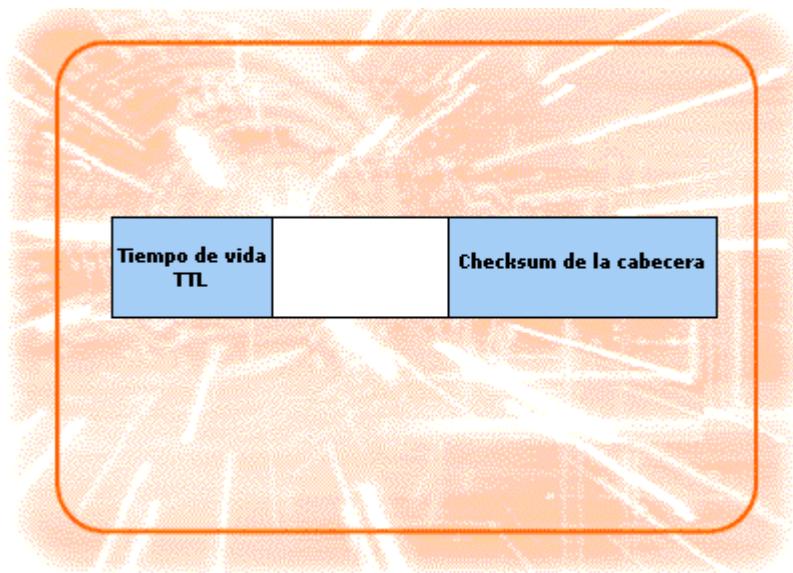
Sólo uno de los 3 bits, D (Minimizar retardo), T (Máximo rendimiento) o R (Máxima fiabilidad) puede estar puesto a 1, es decir, los bits son mutuamente excluyentes. Sólo se puede asignar un valor de TOS en cualquiera de los datagramas IP. La norma Assigned Numbers recomienda ciertos valores para determinadas aplicaciones.

Por ejemplo, minimizar el retraso para Telnet, maximizar el rendimiento cuando se copia un archivo y maximizar la fiabilidad cuando se envían mensajes de gestión de red.

Algunos encaminadores ignoran completamente el campo TOS, mientras que otros, pueden utilizar el campo para tomar decisiones de encaminamiento o para decidir qué tipo de tráfico hay que proteger cuando se va agotando la memoria.

Se cree que el Tipo de servicio tendrá un mayor protagonismo en el futuro.

## Tiempo de vida y checksum de la cabecera



### Tiempo de vida

Cuando ocurre un cambio en la topología de una Internet de IP, al igual que cuando un enlace queda fuera de servicio, o cuando arranca un nuevo encaminador, algunos datagramas pueden estar vagando durante un corto período de tiempo hasta que se elijan nuevas rutas.

Los problemas más serios pueden provenir de errores humanos, cuando se introduce manualmente la información de encaminamiento. Un error puede provocar que los datagramas se " pierdan" o se queden dando vueltas durante mucho tiempo.

El campo Tiempo de vida (TTL – *Time-To-Live*) limita el tiempo que se permite que un datagrama permanezca en una Internet.

En el host de origen se establece el TTL y cada encaminador que maneja el datagrama decremente su valor. Si un datagrama, que aún no ha llegado a su destino, tiene un TTL a cero, se descarta.

Aunque formalmente se define como un tiempo en segundos, realmente el TTL se implanta como un simple contador de saltos que se decrementa, normalmente en uno, en cada encaminador.

Opcionalmente, se puede usar un decremento mayor si un datagrama atraviesa un enlace muy lento o ha permanecido en una cola de espera durante mucho tiempo.

El valor inicial por defecto recomendado para el TTL es, aproximadamente, el doble del camino más largo de la Internet.

La longitud del camino más largo, a veces, se denomina diámetro de la Internet.

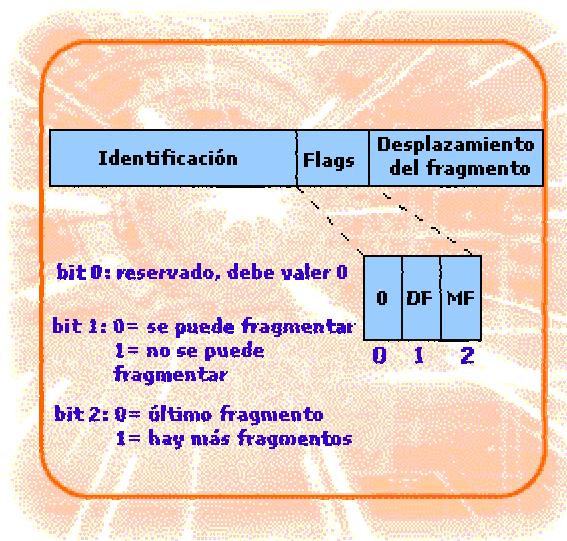
### Checksum de la cabecera

Este campo, de 16 bits, contiene una suma de control que se calcula con los campos de la cabecera IP. El cálculo consiste en tomar el complemento a uno de 16 bits de la suma con complemento a uno de todas las palabras de 16 bits de la cabecera. Antes del cálculo, el campo suma de control se pone a 0.

La suma de control hay que actualizarla según se reenvía el datagrama ya que el Tiempo de vida cambia en cada encaminador.

El resto de los valores también pueden cambiar, debido a la fragmentación o debido a los valores que se escriben en los campos opcionales.

## Identificación, banderas y desplazamiento del fragmento



### Campo Identificación

Contiene un número de 16 bits. Este número permite al host de destino reconocer los fragmentos que pertenecen a un mismo datagrama, es decir, todas las tramas que lleven el mismo número de identificación pertenecen al mismo datagrama.

### Campo Flags (Banderas)

Este campo tiene 3 bits. El bit 0 está reservado y debe ponerse a cero. El origen puede fijar el siguiente bit a uno para evitar que el datagrama se fragmente. Si el datagrama no se puede enviar sin fragmentación y este bit está a uno, habría que descartarlo y se enviaría de vuelta, al origen, un mensaje de error.

El bit 2 se pone a cero si es el último trozo del datagrama, o el único. El bit 2 se pone a uno para indicar que este datagrama es un fragmento y que le siguen más fragmentos.

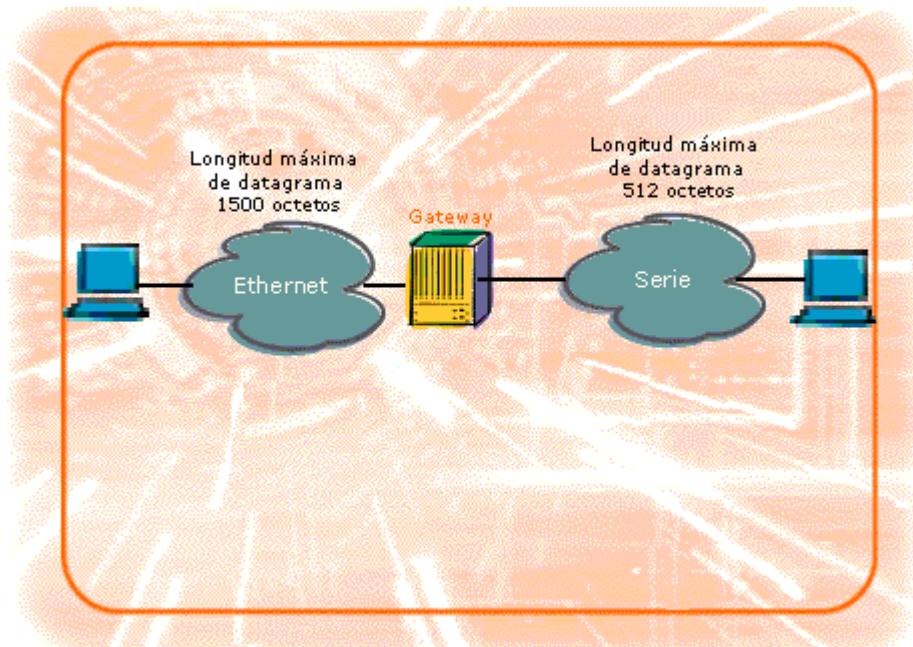
### Campo Desplazamiento del fragmento

Se usa para indicar la posición de un fragmento en relación al comienzo del datagrama original. A un trozo de datos de ocho octetos se le denomina **bloque del fragmento**. El número indicado en el campo **Desplazamiento del fragmento** indica el desplazamiento medido en bloques de fragmento (es decir, el desplazamiento real, dividido por ocho). El campo **Desplazamiento del fragmento** ocupa 13 bits, por lo que los desplazamientos pueden estar entre 0 y 65536 octetos del datagrama completo.

## Mecanismo de fragmentación

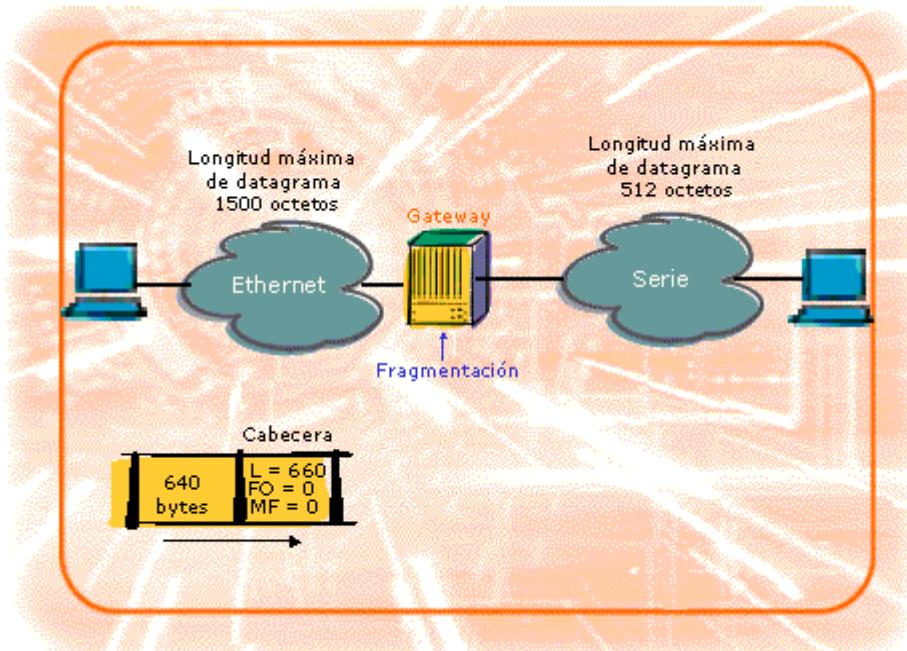
Los campos Identificación, Banderas y Desplazamiento de fragmento, permiten que los datagramas se puedan fragmentar y reensamblar.

Cuando IP necesita transmitir un datagrama con un tamaño mayor que la MTU del siguiente enlace...



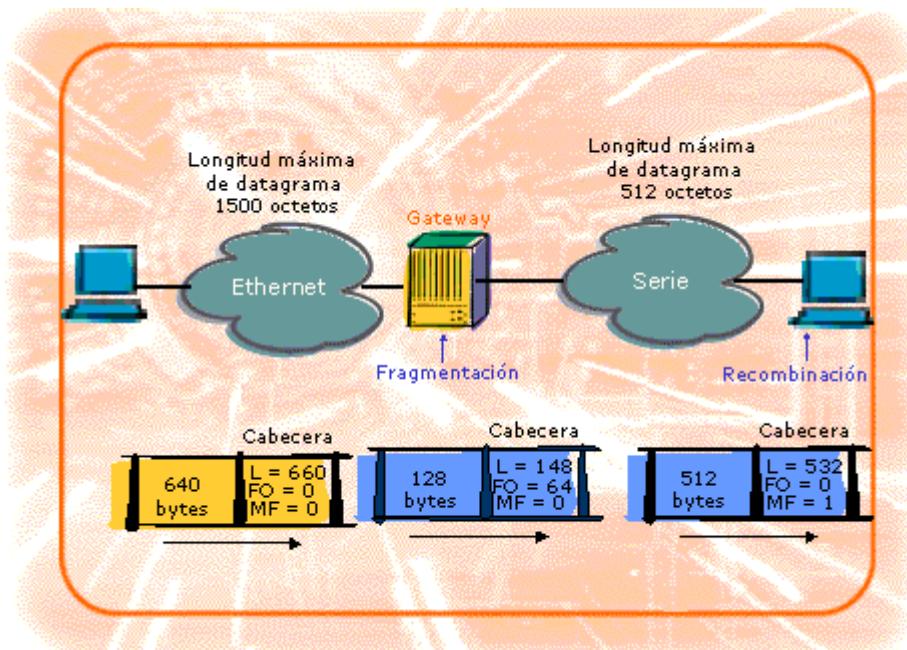
1. El primer paso es comprobar el campo **Banderas** (flags).

Como sabemos existe un bit "No fragmentar" en el campo Banderas, si este bit está a 1, no se puede hacer nada, hay que descartar el datagrama.



2. Si la bandera "No fragmentar" está a 0, los datos se dividen en trozos, según la MTU del siguiente salto.

Cada trozo debe alinearse en un múltiplo de 8 octetos.



3. A cada trozo se le dota de una cabecera IP similar a la cabecera del datagrama original.

En particular, cada trozo tendrá los mismos valores en los campos origen, destino, protocolo e identificación. Sin embargo, hay que poner los siguientes campos a cada trozo por separado:

- **Longitud total del datagrama.** Determina el tamaño específico de este trozo.
- **Bit 2 del campo Flags.** Debe ponerse a 1 en todos, excepto en el último trozo.
- **Desplazamiento del fragmento** se pone para indicar la posición de un fragmento específico en relación al comienzo del datagrama original. La posición inicial es cero.
- Hay que calcular el valor del campo Checksum de la cabecera para cada uno de los fragmentos.

## Opciones del Datagrama IP

- **Ruteo estricto** (**encaminamiento estricto**).
- **Registro de ruta.**
- **Sello de tiempo** (**timestamp**).

Existen hasta 40 octetos útiles para llevar información extra en la cabecera de los datagramas IP, cada uno de los cuales puede tomar una o más opciones. Las opciones son determinadas por las aplicaciones de origen. En cualquier caso el uso de estos octetos extra es poco frecuente.

Entre las más útiles, destacamos las que vemos en la imagen.

**Ruteo estricto:** El principio del ruteo o encaminamiento estricto consiste en proporcionar al emisor una forma en la que pueda determinar la ruta que seguirán los datagramas que envíe. Por ejemplo, para probar la eficiencia de una red en particular N, el administrador puede utilizar la opción de *Ruteo estricto* para forzar a los datagramas a viajar por la red N, incluso aunque los encaminadores por los que pasen seleccionen otra ruta diferente.

**Registro de ruta:** Permite a la fuente crear una lista de direcciones IP y ajustarla para que cada encaminador que maneje el datagrama añada su propia dirección IP a la lista.

Por lo general, una máquina que reciba un datagrama, ignora la ruta registrada. Para usar la opción de *Registro de ruta* se requiere que dos máquinas estén de acuerdo para cooperar; un ordenador no recibirá rutas registradas de los datagramas entrantes ni activará la opción de *Registro de ruta* en los datagramas de salida de forma automática. La fuente debe aceptar la habilitación de la opción de *Registro de ruta* y el destino debe aceptar el procesamiento de la lista resultante.

**Sello de tiempo:** Esta opción funciona de manera muy parecida a la de *Registro de ruta*: contiene una lista inicial vacía y cada encaminador, a lo largo de la ruta, desde la fuente hasta el destino, escribe sus datos en la lista (dirección IP y sello de tiempo). Cada máquina reportará una hora de acuerdo a su reloj local, y los relojes pueden diferir. Así el sello de tiempo deberá considerarse como una estimación, independientemente de la representación.

## Campo opciones

La longitud del campo *Opciones* varía dependiendo de la opción seleccionada. Algunas tienen un octeto de longitud, es decir, su código de opción consiste en un único octeto, mientras que en otras la longitud es variable.

### Campo CÓDIGO de opciones

Cuando un datagrama incluye opciones, éstas aparecen contiguas, sin separadores especiales entre ellas.

Cada opción consiste en:

- Un sólo octeto de *código de opción*,
- que puede llevar, a continuación, uno o varios octetos, en función de la opción concreta de que se trate.

El octeto de código de opción se divide en tres campos:

- Una bandera de un bit, llamada **COPIA**: cuando está puesto a 1, especifica que la opción debe copiarse en todos los posibles fragmentos de ese datagrama.
- Cuando está puesto a 0, significa que la opción sólo se debe copiar dentro del primer fragmento, y no en todos los demás.

### Campo CÓDIGO de opciones

0	Copia
---	-------

- Un campo de dos bits, llamado **CLASE DE OPCIÓN**, que especifica si se trata de un datagrama de *Control de red o de Depuración y medición*.

### Campo CÓDIGO de opciones

0	1	2
Copia	Clase de opción	

- Un campo de 5 bits, llamado **NÚMERO DE OPCIÓN**, que establece la opción específica para una *Clase* determinada.

### Campo CÓDIGO de opciones

0	1	2	3	4	5	6	7
Copia	Clase de opción		Número de opción				

En la figura se muestran las combinaciones más importantes de los campos **Clase de opción** y **Número de opción**.

### Campo CÓDIGO de opciones

0	1	2	3	4	5	6	7
Copia	Clase de opción		Número de opción				
Clase de opción	Número de opción	Longitud	Descripción				
0	7	variable	<b>Registro de ruta.</b> Se utiliza para registrar el trayecto de una ruta				
0	9	variable	<b>Ruteo estricto.</b> Se utiliza para establecer la ruta de un datagrama en un trayecto específico				
2	4	variable	<b>Sello de tiempo.</b> Se utiliza para registrar sellos de hora a lo largo de una ruta				

## 4- Protocolo de Mensajes de Control de Internet (ICMP)

IP tiene un diseño simple y elegante. En condiciones normales, IP hace un uso muy eficiente de los recursos de memoria y transmisión. Pero, ¿qué ocurre cuando las cosas no van bien? Si un encaminador deja de funcionar y se estropea la red, ¿qué aviso se da de que los datagramas están dando vueltas hasta que expira su tiempo de vida? ¿Qué aviso se da a las aplicaciones para que no insistan enviando información hacia un destino inalcanzable?

El **protocolo de Internet de mensajes de control (ICMP - Internet Control Message Protocol)** ofrece remedio a estos problemas.

ICMP también desempeña un papel fundamental de asistente en la red, ayudando a los host con su encaminamiento de IP y permitiendo que los administradores de red comprueben el estado de los nodos de la red.

## 1- Protocolo ICMP

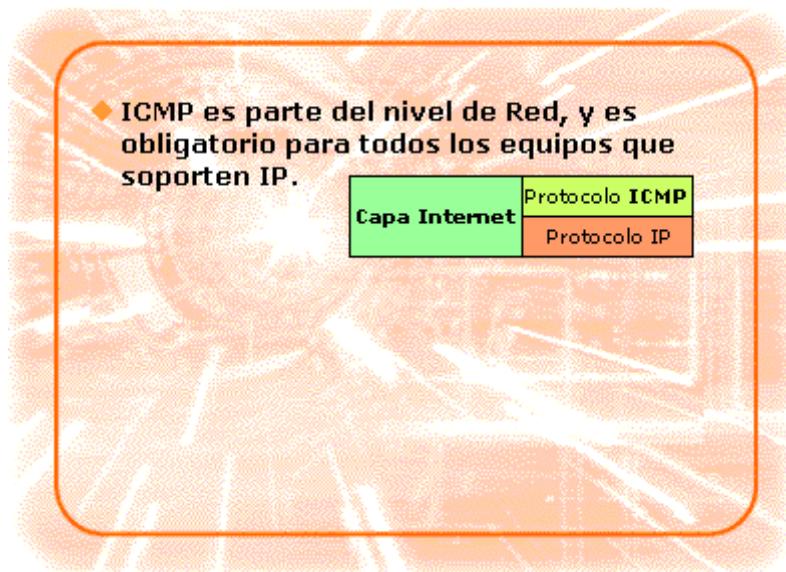
Descubre uno de los protocolos que se encarga de la gestión de los errores en las redes IP.

### Protocolo ICMP

Las funciones de ICMP son una parte esencial de IP.

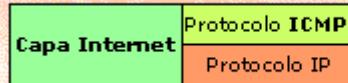
Todos los host y encaminadores deben ser capaces de generar mensajes ICMP y procesar los que reciban.

Si se usa adecuadamente, ICMP puede contribuir a que la red funcione mejor.

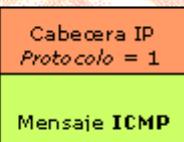


Los mensajes ICMP se transmiten como datagramas IP, con una cabecera normal de IP, con el campo de *Protocolo* con el valor 1.

- ♦ ICMP es parte del nivel de Red, y es obligatorio para todos los equipos que soporten IP.



- ♦ Los mensajes ICMP se transmiten en datagramas IP con el campo *Protocolo* a 1.



## Mensajes de error de ICMP

- ◆ Los datagramas IP pueden descartarse en ciertas situaciones.
- ◆ Los mensajes ICMP informan al origen del datagrama del error.
- ◆ La notificación de los errores no depende de la existencia de un centro de administración de red.
- ◆ ICMP no ofrece funcionalidad de gestión de red: los mensajes se envían a todos los orígenes de los datagramas erróneos.

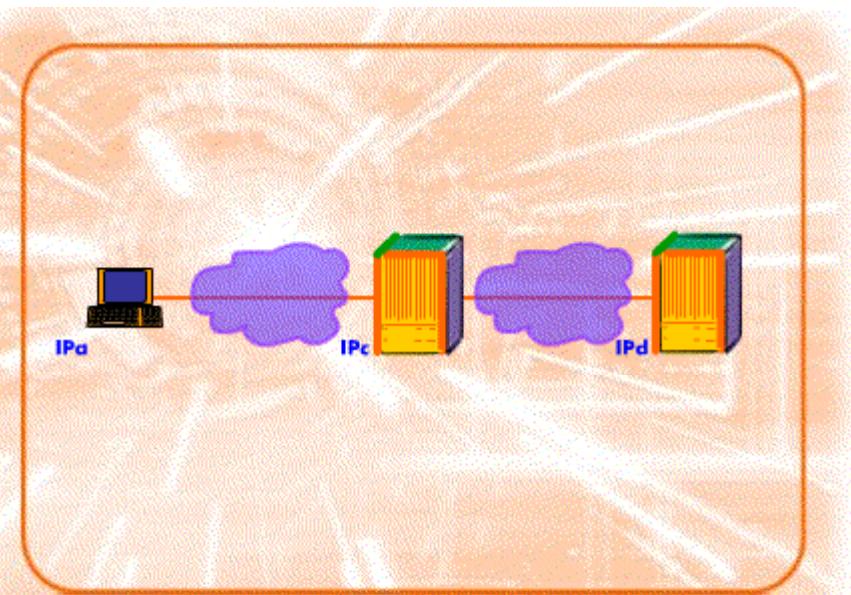
Hay ciertas situaciones en las que se descartan datagramas IP. Por ejemplo, puede que no se llegue a un destino porque el enlace ha caído. Puede que haya expirado el contador Tiempo de vida. O puede que sea imposible que un encaminador reenvíe un datagrama grande porque no se permite la fragmentación.

Cuando hay que descartar un datagrama, se utilizan mensajes ICMP para informar del problema a la dirección origen del datagrama. ICMP notifica rápidamente a los sistemas de los problemas.

ICMP es un protocolo muy robusto, ya que la notificación de errores no depende de la existencia de un centro de administración de red.

También hay desventajas. Por ejemplo, si no se puede alcanzar un destino, los mensajes se propagarán a orígenes de toda la red, en lugar de a una estación de administración de red. De hecho, ICMP no dispone de funciones para avisar de los errores a un centro determinado de operaciones de red. Depende del Protocolo básico de gestión de red (SNMP - *Simple Network Management Protocol*).

## Ejemplo de notificación de errores con ICMP



Los mensajes ICMP viajan a través de Internet en la parte de datos de los datagramas IP, como cualquier otro tipo de tráfico. El destino último de un mensaje ICMP no es un proceso de usuario en la máquina destino, sino el software IP en dicha máquina. Es decir, cuando llega un mensaje de error ICMP, el módulo software IP gestiona por sí mismo el problema; no pasa el mensaje ICMP al programa de aplicación cuyo datagrama causó el problema.

Los datagramas que llevan mensajes ICMP se encaminan exactamente igual que los datagramas que llevan información de usuario.

Por tanto, estos mensajes también son susceptibles de perderse o de descartarse. Estos mensajes son tratados como mensajes de datos excepto que ningún mensaje ICMP será generado por errores que resulten de datagramas que lleven mensajes ICMP.

## Mensajes de error de ICMP

Mensaje	Descripción
Destino inalcanzable	Un datagrama no puede llegar a su host, utilidad o aplicación de destino.
Plazo superado	El Tiempo de vida ha expirado en un encaminador, o el plazo de reensamblado ha expirado en host de destino.
Problemas con los parámetros	Existe un parámetro erróneo en la cabecera de IP.
Acallado de origen	Un encaminador o un destino está congestionado. Se recomienda que los sistemas no envíen mensajes de acallado (se está trabajando en mecanismos de control de congestión más efectivos).
Redirigir	Un host ha enviado un datagrama al encaminador local equivocado.

La figura muestra un resumen de los mensajes de error que pueden enviar los host y encaminadores para avisar de los problemas.

**Destino inalcanzable:** La entrega de un datagrama puede fallar en muchos momentos debido a un enlace roto, a un encaminador físicamente incapaz de llegar a una subred de destino o de ejecutar el siguiente salto de encaminamiento. El destino puede, incluso, estar fuera de servicio por labores de mantenimiento.

Los encaminadores modernos disponen de potentes funciones de seguridad. Se puede configurar un encaminador para que examine el tráfico que pasa por él, de forma que puede que los datagramas no se puedan entregar porque, por razones administrativas, están prohibidas las comunicaciones con el destino.

**Plazo superado:** Un datagrama puede expirar porque su tiempo de vida ha llegado a cero mientras se encontraba en tránsito. Otra razón es cuando el plazo de reensamblado del host expira antes de que lleguen todos los fragmentos.

**Problemas de parámetros:** Se utiliza para informar de otros problemas que no se cubren con ningún otro tipo de mensaje de error. Por ejemplo, puede existir información inconsistente en un campo de opciones que haga que sea imposible entregar el datagrama correctamente, por lo que hay que descartarlo. Lo más habitual, en cuanto a problemas de parámetros, se debe a errores de implementación en el sistema que escribió los parámetros en la cabecera IP.

**Acallamiento de origen:** El objetivo de este mensaje de error era aliviar los problemas de congestión que se producen cuando, por ejemplo, una aplicación en un host está generando datagramas más rápidamente de lo que un encaminador puede procesarlos. Sin embargo, no ha tenido éxito. De hecho, se recomienda no utilizarlo, y se están estudiando otros mecanismos de control de congestión.

**Redirección:** Puede que haya más de un encaminador conectado a una LAN. Si el host local envía un datagrama al encaminador equivocado (puede que éste no sea el que proporciona la ruta más corta para llegar al destino), el encaminador reenviará el datagrama al encaminador adecuado y un mensaje Redirección al host origen. En adelante, el host debería enviar el tráfico por el encaminador sugerido en el mensaje de *Redirección*.

## Tratamiento de los mensajes de error ICMP entrantes

¿Qué se hace cuando llega un mensaje de error ICMP a un host de origen?

La forma en que los fabricantes implementan el software de red es muy variada, y la norma de TCP/IP intenta dar mucha libertad.

Las guías que se dan para los distintos tipos de mensajes se muestran en el gráfico.

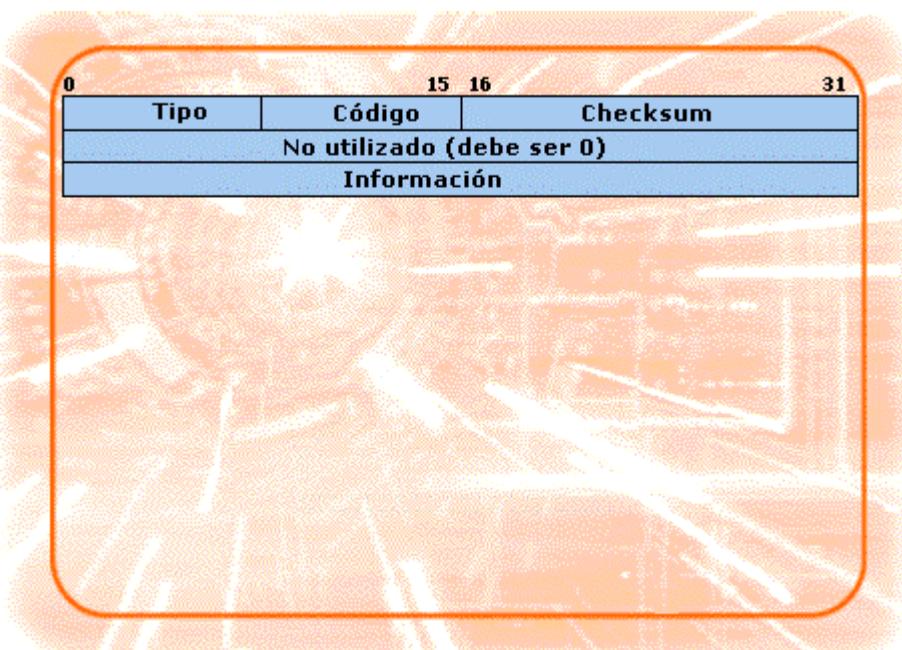
Mensaje	Descripción
Destino inalcanzable	Entregar el mensaje ICMP a la capa de transporte. La acción depende de si la razón es permanente o transitoria; ej, se prohíbe la comunicación por razones administrativas.
Plazo superado	Entregar el mensaje ICMP a la capa de transporte.
Problemas con los parámetros	Entregar el mensaje ICMP a la capa de transporte. Opcionalmente, notificar al usuario.
Acallado de origen	Entregar el mensaje ICMP a la capa de transporte o a un módulo de procesamiento de ICMP.
Redirigir	El host <i>debe</i> actualizar su tabla de encaminamiento.

A veces, se pueden tratar las condiciones de error mediante la cooperación entre el sistema operativo, el software de comunicaciones y la aplicación que realiza la comunicación.

## Formato del mensaje de error

Recordemos que los mensajes de ICMP se transmiten en la parte de datos de un datagrama de IP.

Cada mensaje ICMP empieza con los mismos tres campos: un campo de **Tipo**, un campo de **Código** que, a veces, ofrece la descripción concreta del error, y un campo **Checksum**. El formato del resto del mensaje viene determinado por el tipo de mensaje ICMP de que se trate.



El gráfico nos muestra, a modo de ejemplo, los posibles códigos de error para el mensaje ICMP Destino Inalcanzable.

0	15 16	31
Tipo	Código	Checksum
No utilizado (debe ser 0)		
Información		
<b>Código Significado</b>		
0	No se puede llegar a la red	
1	No se puede llegar al host	
2	El destino no dispone del protocolo solicitado	
3	No se puede llegar al puerto. Puede que la aplicación de destino no esté libre	
4	Se necesita realizar fragmentación, pero la bandera "No fragmentar" está activa	
5	La ruta de origen no es correcta	
6	No se conoce la red de destino	
7	No se conoce el host de destino	
8	El host de origen está aislado	
9 - 10	La comunicación con la red de destino está prohibida por razones administrativas	
11	No se puede llegar a la red debido al Tipo de servicio	

## Mensajes de petición ICMP

---

No todos los mensajes ICMP son señales de error. Algunos se utilizan para obtener información útil de la red. ¿Está vivo el host X? ¿Está ejecutando el host Y? ¿Cuánto tarda un mensaje de ida y vuelta a Z? ¿Cuál es la máscara de mi dirección?

Concretamente, entre los mensajes de petición de ICMP están:

- Mensajes de petición y respuesta de **Eco** que se pueden intercambiar con los host o con los encaminadores.
- Mensajes de petición y respuesta de **Máscara de dirección**, que permiten a un sistema descubrir la máscara de dirección que debería asignar a un interfaz.
- Mensajes de petición y respuesta de **Marca de tiempo**, que leen el reloj de un sistema dado.

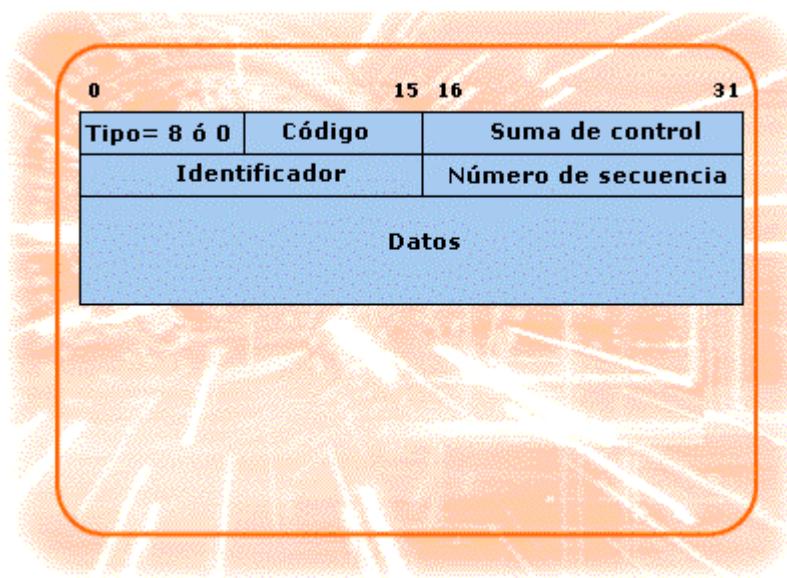
Veamos, a continuación, el caso de Mensajes ICMP de petición y respuesta de eco.

## Mensajes ICMP de petición y respuesta de eco

La petición de Eco y la respuesta de Eco se utilizan para comprobar si un sistema está activo.

Se usa Tipo = 8, para la petición, y Tipo = 0, para la respuesta.

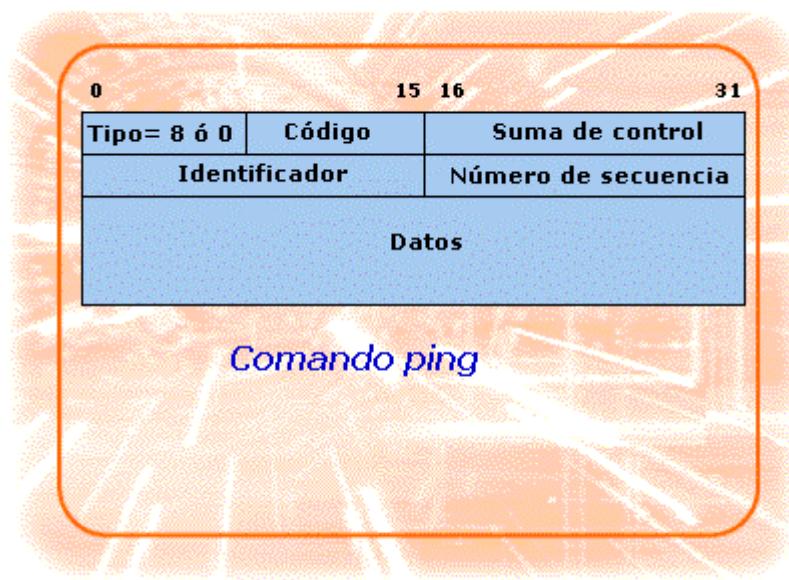
El número de octetos del campo de datos es variable y se selecciona en origen.



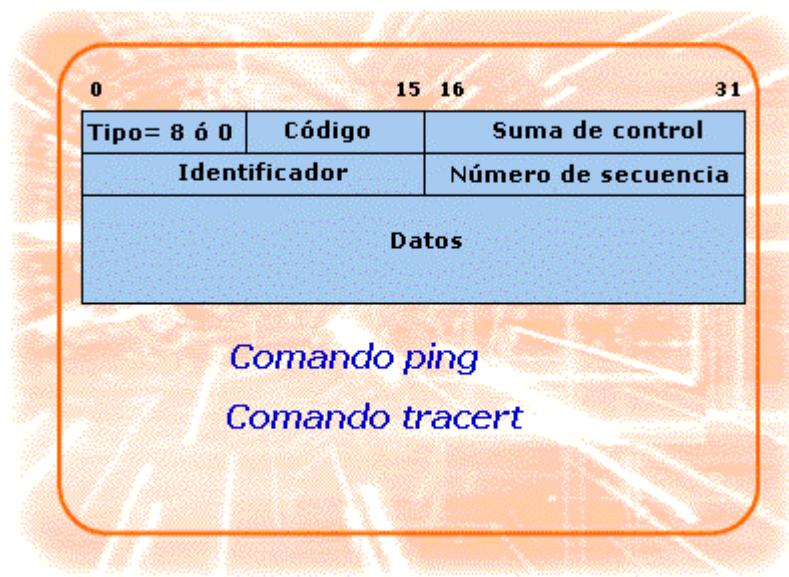
El destino debe enviar, de vuelta, el mismo mensaje recibido.

El campo Identificador se utiliza para hacer coincidir la respuesta con la petición original.

El famoso comando **ping**, que existe en casi todos los sistemas con TCP/IP, está programado usando los mensajes de petición y respuesta de eco.



Pero no es el único comando que los utiliza; así, el comando **traceroute** o **tracert** también hace uso de dichos mensajes, aunque de manera algo diferente.



## 5- Protocolo de datagramas de usuario (UDP)

Una vez que se ha tratado el movimiento de los bits por un medio físico y el encaminamiento de datagramas por una Internet, ya estamos preparados para empezar con los servicios que utilizan las aplicaciones para transferir sus datos. El primero que trataremos será el **Protocolo de datagramas de usuario (UDP - User Datagram Protocol)**. El funcionamiento de este protocolo es muy directo, UDP permite que las aplicaciones se envíen entre sí mensajes individuales.

¿Por qué definir este tipo de servicio? Existen muchas aplicaciones que se pueden construir utilizando los datagramas de usuario. Por ejemplo, una simple base de datos. Para este tipo de aplicaciones se puede evitar la sobrecarga de enviar y recibir los múltiples mensajes necesarios para establecer y cerrar una conexión, mediante la implementación de un simple mecanismo de petición y respuesta, y es aquí donde entra UDP.

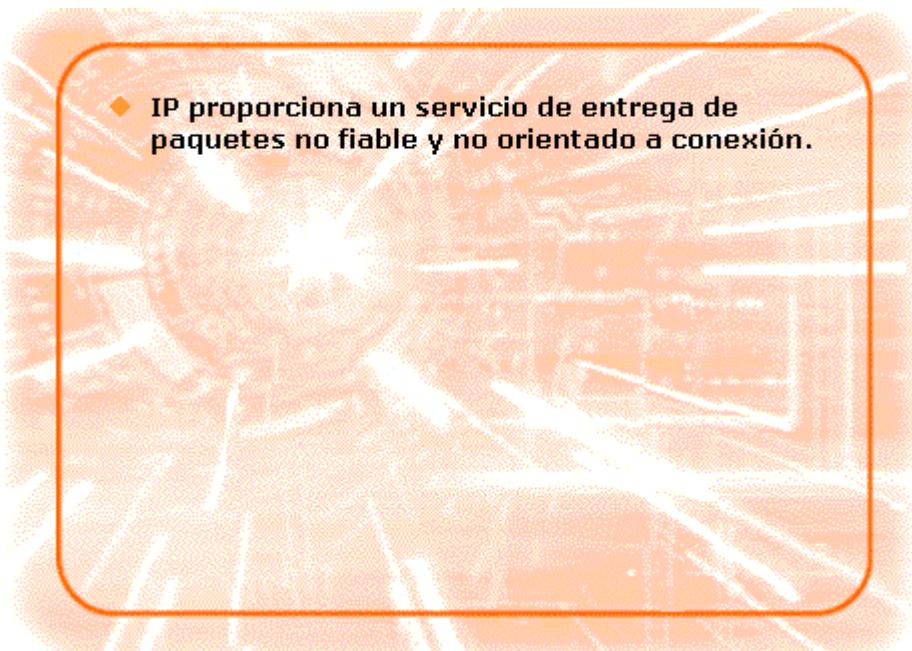
UDP también es una herramienta perfecta para realizar funciones de monitorización, depuración, gestión y prueba.

## 1- Protocolo UDP

Averigua como funciona el protocolo de transporte más simple de los dos que utiliza la pila TCP/IP, el protocolo UDP.

### Protocolo UDP

Ya hemos mencionado que el protocolo IP proporciona un mecanismo para el envío y la entrega de paquetes sobre una variedad de redes físicas interconectadas. Sin embargo, IP se basa en un servicio no orientado a conexión y no fiable. Por lo tanto, es necesario un software que proporcione fiabilidad adicional si se requiere un servicio seguro.

- 
- ◆ IP proporciona un servicio de entrega de paquetes no fiable y no orientado a conexión.

Supongamos que no se requiere un servicio fiable. ¿Debe entonces IP proporcionar un mecanismo para alcanzar a cada usuario?

- ◆ IP proporciona un servicio de entrega de paquetes no fiable y no orientado a conexión.
- ◆ Si no necesitamos fiabilidad, ¿podría IP, por sí mismo, hacer llegar los datos de los programas de aplicación al destino final?

La mayoría de los ordenadores modernos soportan el multiproceso, lo cual significa que permiten que múltiples programas de aplicación se ejecuten simultáneamente. Cada programa de aplicación puede requerir una comunicación con alguien en la red y, por tanto, se necesita una capacidad para que este programa pueda ser accedido desde la red. En el protocolo IP, una dirección de destino identifica a un ordenador "host"; no se hace distinción de usuarios ni de programas de aplicación en un mismo ordenador que reciba un datagrama.

- ◆ IP proporciona un servicio de entrega de paquetes no fiable y no orientado a conexión.
- ◆ Si no necesitamos fiabilidad, ¿podría IP, por sí mismo, hacer llegar los datos de los programas de aplicación al destino final?
- ◆ El multiproceso en los ordenadores exige poder distinguir entre varios usuarios o aplicaciones en la misma máquina.

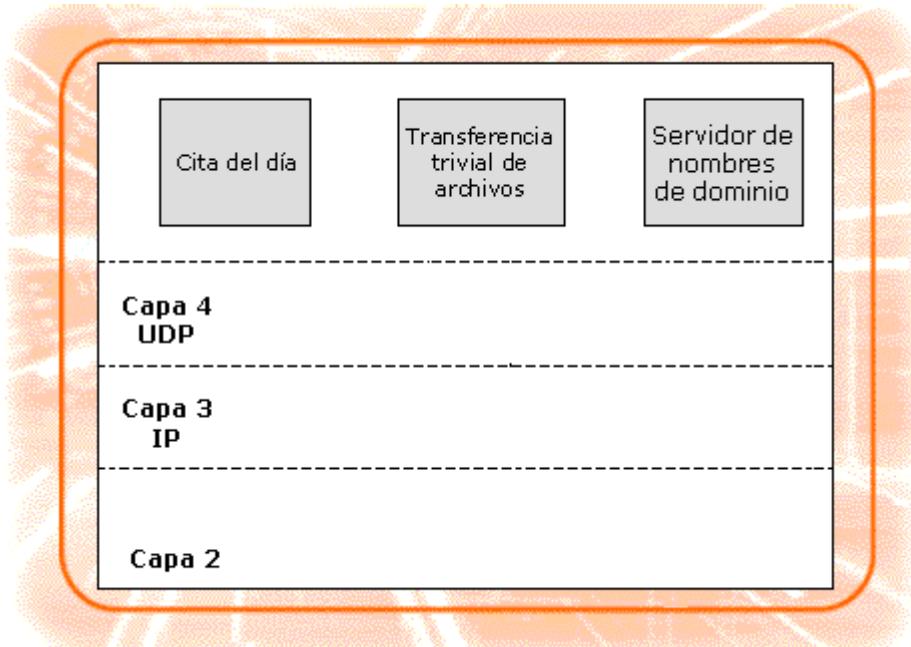
En la pila de protocolos TCP/IP, el Protocolo de Datagramas de Usuario (UDP, "User Datagram Protocol") proporciona un mecanismo que permite, a los que envían datagramas, distinguir entre múltiples receptores en la misma máquina. Además de los datos enviados por un proceso de usuario, cada mensaje UDP contiene un número de **puerto de destino** y un número de **puerto de origen**, lo que posibilita que el software UDP envíe el mensaje al correcto destinatario y a éste, enviar una respuesta.

- ◆ IP proporciona un servicio de entrega de paquetes no fiable y no orientado a conexión.
- ◆ Si no necesitamos fiabilidad, ¿podría IP, por sí mismo, hacer llegar los datos de los programas de aplicación al destino final?
- ◆ El multiproceso en los ordenadores exige poder distinguir entre varios usuarios o aplicaciones en la misma máquina.
- ◆ UDP es el mecanismo de transporte, no fiable y no orientado a conexión, que introduce el concepto de **puertos** para distinguir entre varias aplicaciones en la misma máquina.

UDP hace uso del protocolo Internet subyacente para transportar un mensaje UDP desde una máquina a otra. Proporciona el mismo servicio de entrega, no fiable y no orientado a conexión, que IP. No usa reconocimientos para asegurar que los mensajes lleguen, no ordena los mensajes entrantes y no proporciona realimentación para controlar la velocidad del flujo de información entre las máquinas. Por tanto, los mensajes UDP pueden perderse, duplicarse o llegar desordenados. Además, los paquetes pueden, incluso, llegar más rápido que lo que el receptor pueda procesarlos.

## Puertos de las aplicaciones: encapsulación y demultiplexación

¿Qué ocurre con los datos cuando llegan a un host de destino? ¿Cómo se entregan al proceso apropiado?



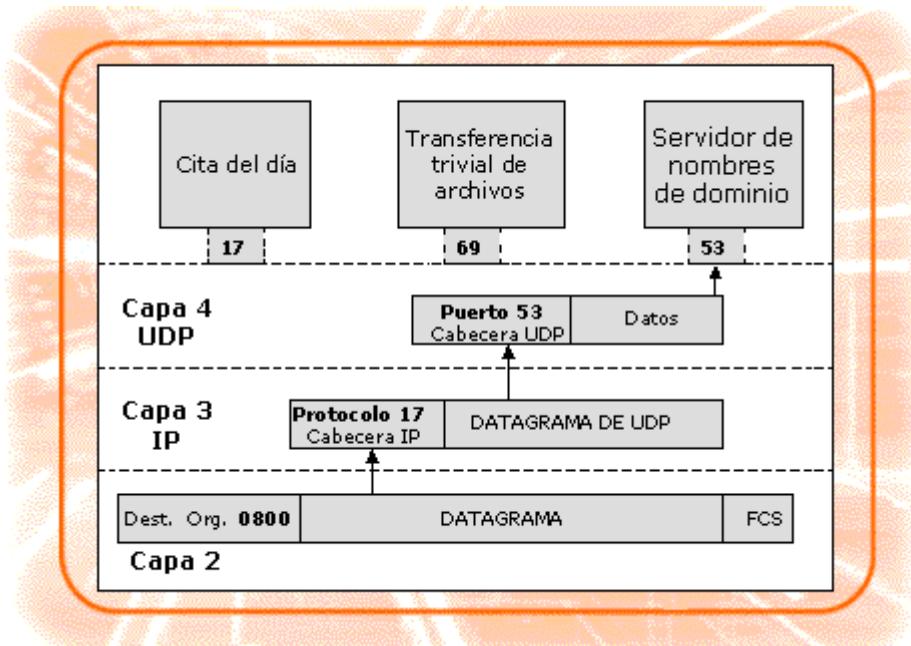
Como se muestra en la figura, en los paquetes que llegan a cada capa, existe un identificador de protocolo que indica qué hacer con los datos entrantes.

En la capa 2, un paquete Ethernet que incluye en la cabecera de trama el valor 0x0800, indica que la trama se debe pasar a IP.

En la capa 3, el campo Protocolo en la cabecera del datagrama IP identifica al protocolo de la capa 4 al que hay que trasladar las tramas, siendo por ejemplo, 6 para TCP y 17 para UDP.

Se puede esperar que un host participe en muchas comunicaciones simultáneas en un cierto momento.

¿Cómo se ordenan y se entregan apropiadamente los datagramas de UDP a los procesos de la capa de aplicación?



La respuesta es que, a cada extremo de una comunicación UDP se le asigna un identificador de 16 bits llamado **número de puerto**.

Los números de puerto del 0 al 1023 están reservados para servicios estándar.

Los puertos estándar se llaman **puertos públicos** (*well-known*).

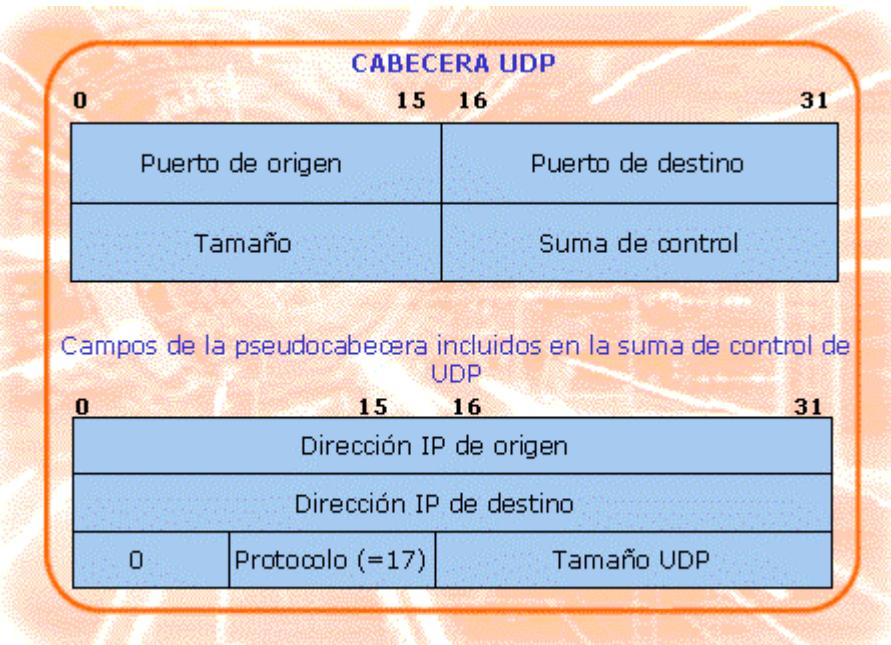
El uso de puertos públicos permite que los clientes identifiquen el servicio al que se desea acceder.

Por ejemplo, al servicio de nombres de dominio, usando UDP, se accede por el puerto público 53.

## Mecanismos del protocolo UDP

¿Qué mecanismos se necesitan para que funcione el Protocolo de datagramas de usuario?

El identificador único de protocolo asignado al protocolo UDP es el 17, éste es el número que se habrá de situar en el campo **Protocolo** en los mensajes salientes de UDP. Los mensajes entrantes, cuyo campo **Protocolo** contenga un 17, se entregan a UDP.



### Cabecera UDP

UDP crea un mensaje añadiéndole una cabecera simple a los datos de la aplicación. Esta cabecera contiene los números de puerto origen y destino, de 16 bits, que identifican los extremos de la comunicación.

El campo tamaño indica el número total de octetos en la cabecera y en la parte de datos del mensaje UDP.

El campo **Suma de control** se utiliza para validar el contenido del mensaje, aunque su uso es opcional.

La suma de control de UDP se calcula sobre una combinación de una pseudocabecera construída especialmente con cierta información de IP, la cabecera UDP y los datos del mensaje. Fíjate en la imagen.

## 6- Servicio de transporte de flujo confiable (TCP)

IP es simple para que la capa de red se centre en la importante función de encaminar los datos desde el emisor hasta el destino. La función de TCP es convertir el intercambio de datagramas en una conexión, sólida y fiable, de datos entre aplicaciones implementadas en los host finales, no en los encaminadores. Los servicios como World Wide Web (WWW), conexión a terminales remotos, transferencia de archivos y transferencia de mensajes, se efectúan con conexiones TCP.

## 1- Protocolo TCP

TCP se encarga de "cubrir las espaldas" al protocolo IP.

### Protocolo TCP

Los dos servicios más importantes de Internet son IP y TCP.

IP es responsable del servicio de entrega de paquetes independiente de la red subyacente. **TCP ("Transmission Control Protocol") proporciona un servicio de transporte orientado a conexión y fiable extremo a extremo.**

TCP vs. UDP		
Servicio	TCP	UDP
Establecimiento y liberación de la conexión	*	
Entrega en secuencia	*	
Multiplexación de varias conexiones de transporte en un único servicio IP	*	*
Control de flujo	*	
Reconocimientos extremo a extremo	*	
Chequeo de errores	*	*

Comparado con UDP, TCP añade una sustancial funcionalidad pero, como consecuencia, también es sustancialmente más complejo.

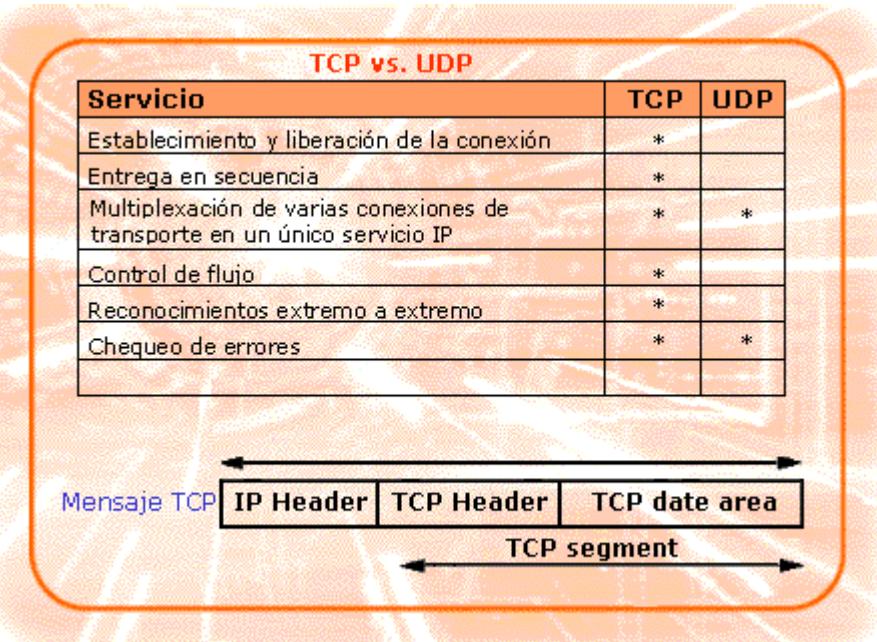
Un sumario de las funciones de TCP y UDP se muestra en el gráfico.

TCP es parte del grupo de protocolos de Internet. Junto con IP, ofrecen un servicio de transporte fiable cualquiera que sea la red física subyacente. En un entorno de redes múltiples, la combinación TCP/IP es muy útil.

Sin embargo, TCP, así como IP, es una entidad independiente y podría usarse individualmente sobre una red sencilla, como una Ethernet.

De igual manera que UDP, TCP se coloca encima de la capa IP.

Esto significa que un mensaje completo TCP, incluyendo la cabecera y los datos, se encapsula en un datagrama IP, y en él viaja a través de Internet.

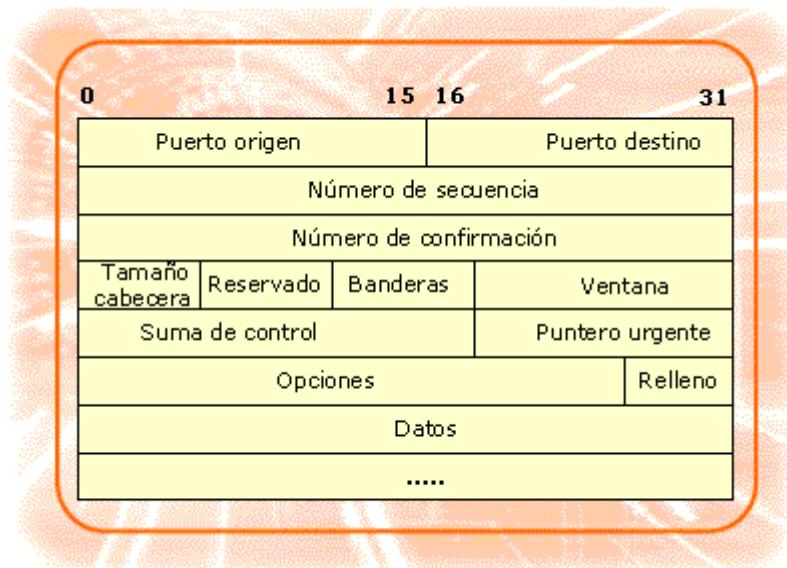


La unidad de transferencia entre entidades TCP en dos máquinas se denomina **SEGMENTO**.

Los segmentos son intercambiados para establecer la conexión, transferir datos, enviar reconocimientos, notificar el tamaño de ventana y cerrar la conexión.

## Formato del segmento

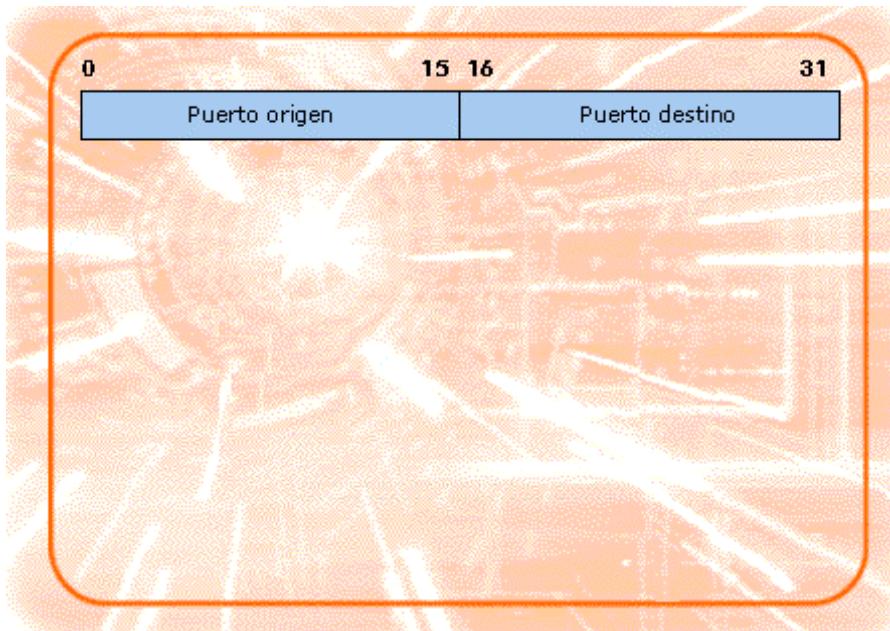
En la imagen se muestra el formato del segmento TCP.



## Puertos de aplicación

Un cliente debe identificar el servicio que desea. Esto se realiza especificando la dirección IP del host y su número de puerto TCP.

Al igual que para el Protocolo de datagramas de usuario (UDP), los números de puerto de TCP están en el intervalo 0 a 65.535. Recordemos que los puertos en el intervalo 0 a 1023 ya están asignados y se utilizan para acceder a servicios estandarizados.



En la imagen se muestra una lista de algunos puertos de TCP y sus aplicaciones.

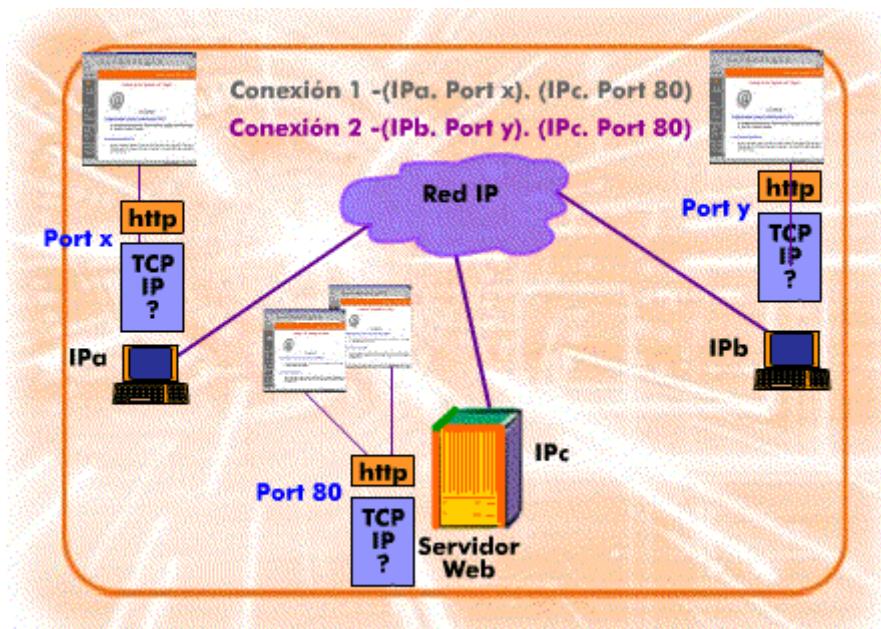
El tráfico que se envía a un puerto TCP está totalmente separado del tráfico que se envía al mismo número de puerto UDP.

0	15 16	31
Puerto origen		Puerto destino
<b>Puerto Aplicación Descripción</b>		
20	Datos	<b>Puerto de transferencia de datos para la transferencia de archivos.</b>
21	FTP	<b>Puerto de diálogo para la transferencia de archivos.</b>
23	TELNET	<b>Puerto de conexión remota mediante Telnet.</b>
25	SMTP	<b>Puerto de Protocolo simple de transferencia de correo.</b>
80	HTTP	<b>Puerto de Protocolo de transferencia de hipertextos.</b>
110	POP3	<b>Servicio de recuperación de correo de PC.</b>
119	NNTP	<b>Acceso a las noticias de la red.</b>

¿Y los puertos que usan los clientes? La mayor parte de las veces, un cliente que quiere una conexión, pide al sistema operativo que le asigne un número en desuso, sin reservar.

Al finalizar la conexión, el cliente devuelve el puerto al sistema y lo puede utilizar otro cliente. Como existen más de 63.000 puertos sin reservar, los clientes no tienen ningún problema de uso de puertos.

## Identificadores de conexión



Mientras el UDP utiliza la noción de puerto, TCP usa la noción de "conexión", donde cada conexión está identificada por la doble pareja: (D1, P1) y (D2, P2), siendo D1 y D2 las direcciones IP de la máquina remitente y destinatario, y P1 y P2 los puertos usados para la conexión entre ellos.

Así, debido a que el TCP identifica una conexión con una pareja de puntos finales, un puerto TCP determinado puede ser compartido por múltiples conexiones similares en la misma máquina.

## Mecanismos de fiabilidad de TCP



### Numeración y confirmación

TCP emplea la numeración y la confirmación (ACK) para la transferencia fiable de datos.

El esquema de numeración de TCP no es el habitual: *todos los octetos* enviados por una conexión de TCP es como si tuviesen su número de secuencia.

La cabecera de un segmento TCP contiene el número de secuencia del *primer octeto de datos en el segmento*.

Se espera que el receptor confirme la recepción de los datos. Si no llega un ACK en un plazo dado, se retransmiten los datos.

El TCP receptor va observando cuidadosamente la secuencia de números que llegan para mantener los datos en orden y para asegurarse de que no se pierden datos.

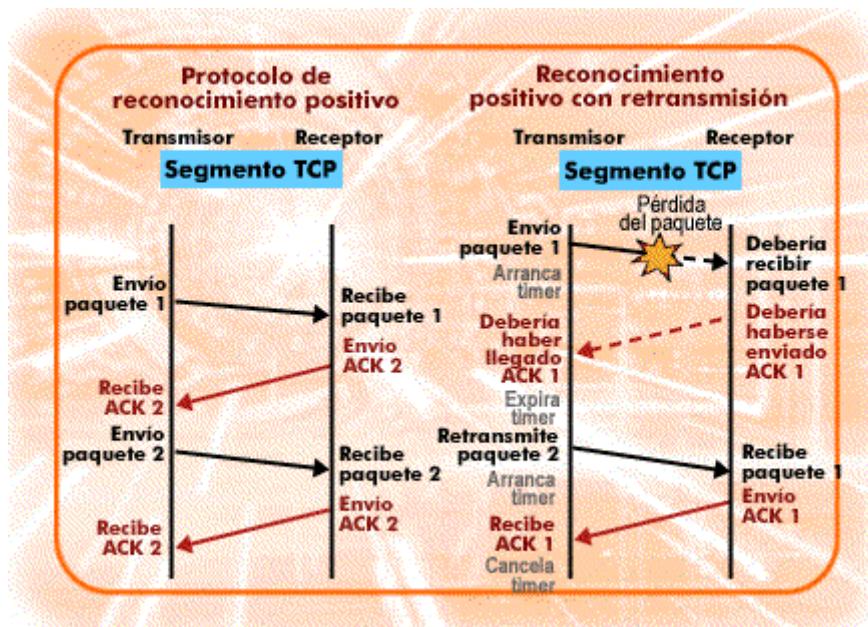
Como a veces se pierden algunos ACK, o llegan tarde, pueden llegar segmentos duplicados al receptor.

Los números de secuencia indican cuáles son los datos duplicados que, por tanto, se pueden descartar.

Como ya hemos mencionado, el campo número de secuencia contiene el número de secuencia del *primer octeto de datos en el segmento*.

Sin embargo, el campo *número de confirmación* contiene el número de secuencia del *siguiente* octeto que se espera del otro extremo.

## Reconocimientos



Uno de los servicios proporcionados por TCP es la fiabilidad. El software del protocolo debe hacer esto con independencia del servicio de red de entrega de paquetes subyacente.

La técnica usada en TCP/IP se conoce como **Reconocimiento Positivo con Retransmisión**.

En esta técnica, el receptor envía hacia atrás un mensaje de reconocimiento cada vez que recibe datos. El transmisor recuerda cada paquete que ha enviado y espera a recibir un reconocimiento antes de enviar el siguiente paquete.

El transmisor también arranca un temporizador cuando envía un paquete, y retransmite el paquete si dicho temporizador expira antes de que llegue el reconocimiento.

## RTT

Es probable que las redes experimenten grandes retardos que puedan causar retransmisiones prematuras y, por tanto, duplicación de mensajes. Para evitar la confusión causada por mensajes retrasados o duplicados, el protocolo de reconocimiento positivo utiliza ciertos **mecanismos de ayuda**:

- Primero, los números de secuencia se envían de vuelta en el campo ACKNOWLEDGMENT. Esto permite al transmisor asociar reconocimientos con segmentos transmitidos.
- Segundo, para evitar retransmisiones innecesarias debido a los altos retardos de transmisión, TCP utiliza un algoritmo adaptativo de retransmisión.

En una Internet, el camino entre un par de máquinas puede atravesar una red de alta velocidad, o puede atravesar múltiples redes intermedias a través de múltiples gateways. Es decir, es imposible, a priori, saber con qué rapidez nos va a llegar un reconocimiento.

Por tanto, el software TCP en el transmisor guarda el tiempo en el que envía cada segmento y el tiempo en el que llega el reconocimiento. El intervalo transcurrido se conoce como "*Round Trip Time*" (**RTT**). Cuando se mide un nuevo RTT, TCP ajusta su noción de RTT medio para la conexión.

## Campos



El campo **Tamaño de la cabecera** (*Data offset*) identifica el número de palabras de 32 bits (4 octetos) que hay en la cabecera TCP. Su valor por defecto es 5 (es decir, 20 octetos).

Se trata de un campo **reservado** para uso futuro y que no tiene todavía una utilización específica.

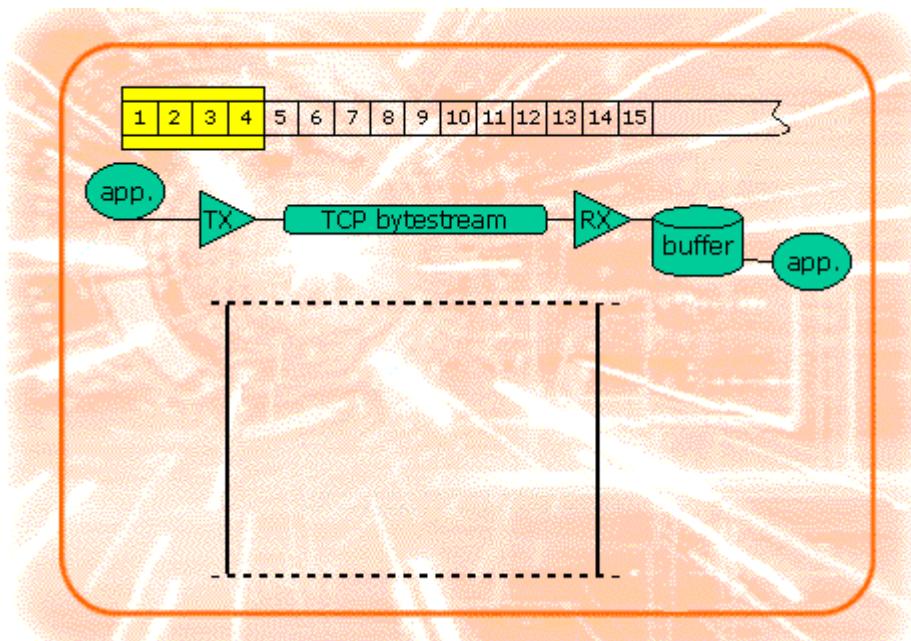
El TCP que recibe los datos se encarga del flujo de los datos de entrada. El receptor decide cuántos datos desea aceptar y el emisor debe actuar dentro de esos límites. Durante el establecimiento de la conexión, cada parte asigna espacio para los búfer de recepción para esa conexión y se lo comunica al otro extremo mediante el campo **Ventana**, especificando en él: "Éste es el número de bytes que puedes enviarme". Este número suele ser un múltiplo entero del tamaño máximo del segmento.

El flujo de datos llega al búfer de recepción y permanece ahí hasta que lo recoge la aplicación asociada a ese puerto TCP, momento en el cual, el espacio se libera para los próximos datos de entrada.

## Sliding Window

La motivación para el reconocimiento positivo y la retransmisión era conseguir fiabilidad. El emisor transmite un paquete y espera su reconocimiento antes de transmitir el siguiente.

La red estará totalmente ociosa durante los momentos que las máquinas retrasen sus respuestas. En una red caracterizada por altos retardos de transmisión, este método resulta altamente ineficiente.



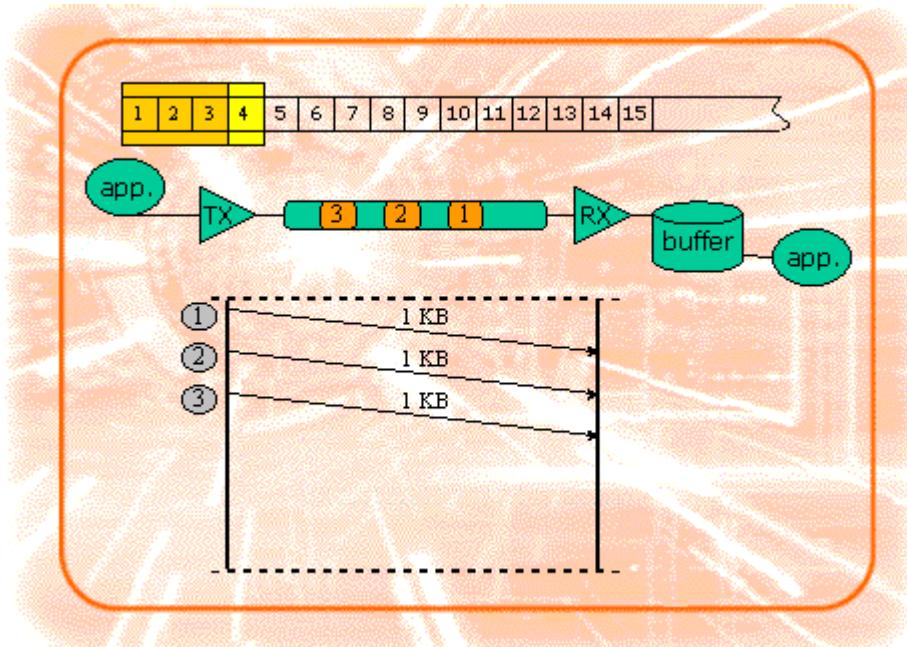
El mecanismo de **ventana deslizante** ("sliding window") utiliza, de una manera más eficiente, el ancho de banda de la red, porque permite al emisor transmitir múltiples paquetes antes de esperar un reconocimiento.

Ilustremos, con un ejemplo, el funcionamiento del mecanismo de ventana deslizante. Partimos de una situación inicial en la que:

- Disponemos de un buffer de almacenamiento de 4 Kbytes en el receptor.
- Por tanto, se ha acordado un tamaño de ventana de 4 Kbytes.
- Cada segmento puede trasnmitir 1 Kbyte.
- No se ha enviado todavía ningún segmento.

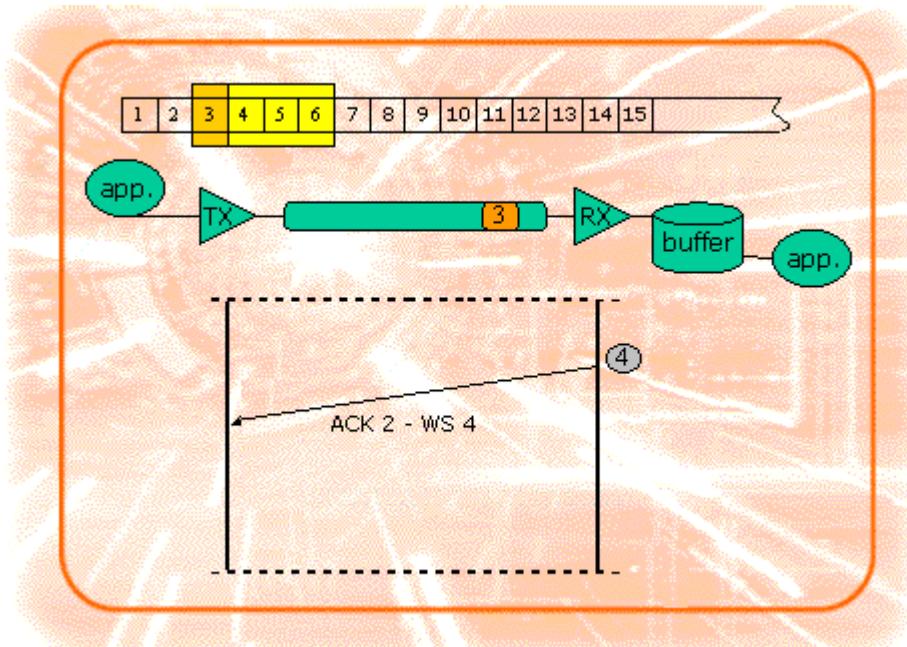
Como el tamaño de ventana es 4 Kbytes, el transmisor puede enviar hasta 4 Kbytes sin esperar ningún reconocimiento.

Supongamos que envía 3 Kbytes en segmentos de 1 Kbyte. El receptor no tiene necesidad de enviar ninguna confirmación, así que no envía ningún ACK.



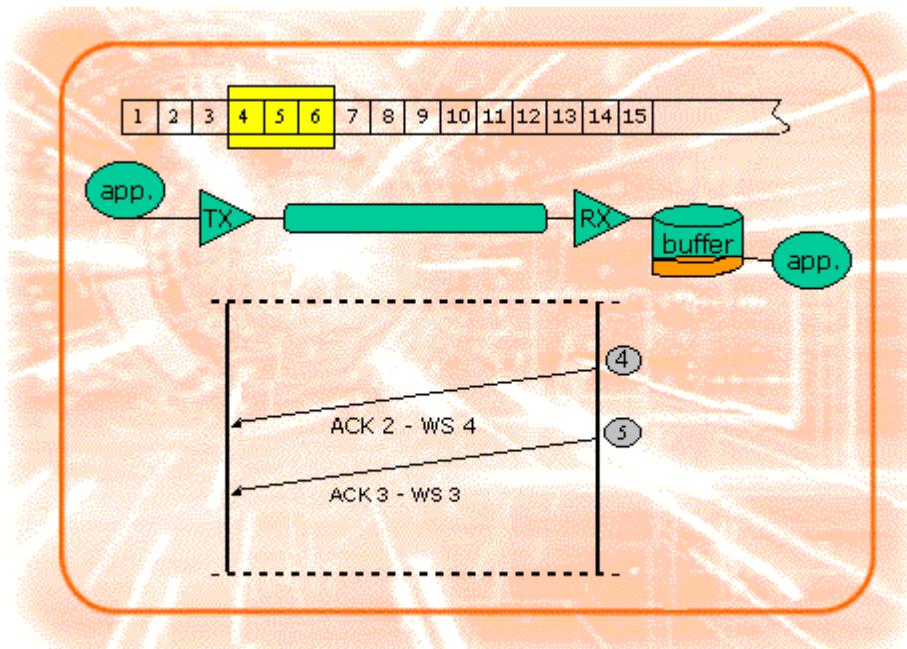
El receptor reconoce los dos primeros segmentos con un mensaje ACK en el que indica que ha aceptado los 2 Kbytes primeros. Además, le comunica al transmisor que el tamaño de ventana es 4 Kbytes, de manera que el transmisor puede enviar otros cuatro segmentos de 1 Kbyte porque el buffer de recepción del receptor está vacío, ya que los segmentos 1 y 2 han sido pasados a las aplicaciones.

La ventana se abre hasta 6, conteniendo todavía al 3 porque éste ha sido enviado, pero no reconocido. Esto significa que todavía está en el canal.

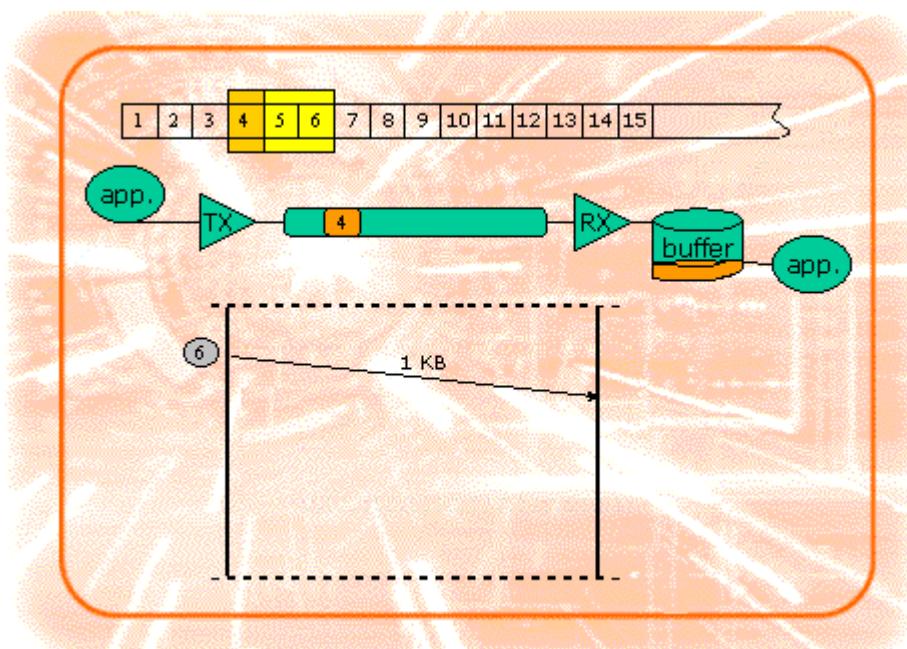


El receptor reconoce el segmento 3, pero la aplicación no ha recibido los datos todavía, por lo que estos permanecen en el buffer de recepción. Es decir, en estos momentos, el receptor sólo podría recibir 3 Kbytes más de información para no llenar su buffer de recepción. Esta contingencia se la comunica al transmisor modificando el tamaño de su ventana a 3 Kbytes (Window Size = 3).

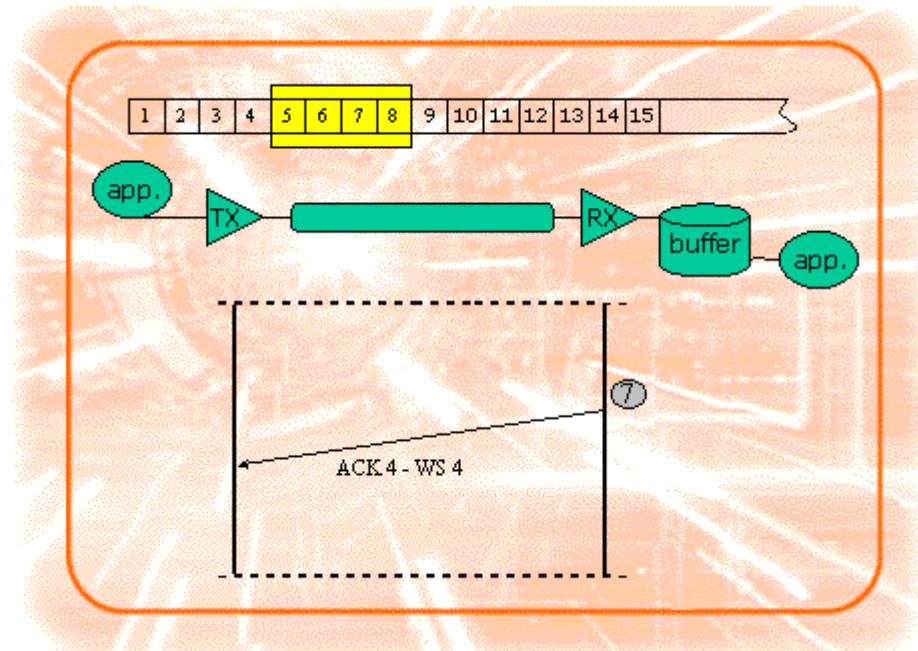
La recepción de un nuevo tamaño de ventana obliga al transmisor a acortar su tamaño en 1 Kbyte.



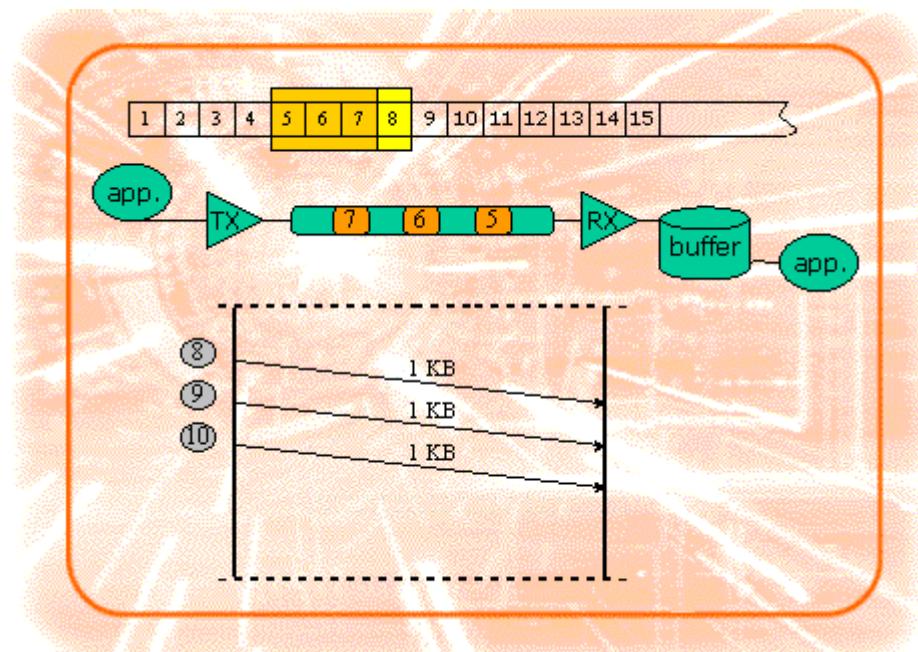
Se envía un nuevo segmento.



Todos los segmentos son reconocidos y enviados a la aplicación. El buffer de recepción está nuevamente vacío. Por tanto, se vuelve a modificar el tamaño de ventana a 4 Kbytes y se abre en 5 y se cierra en 8.

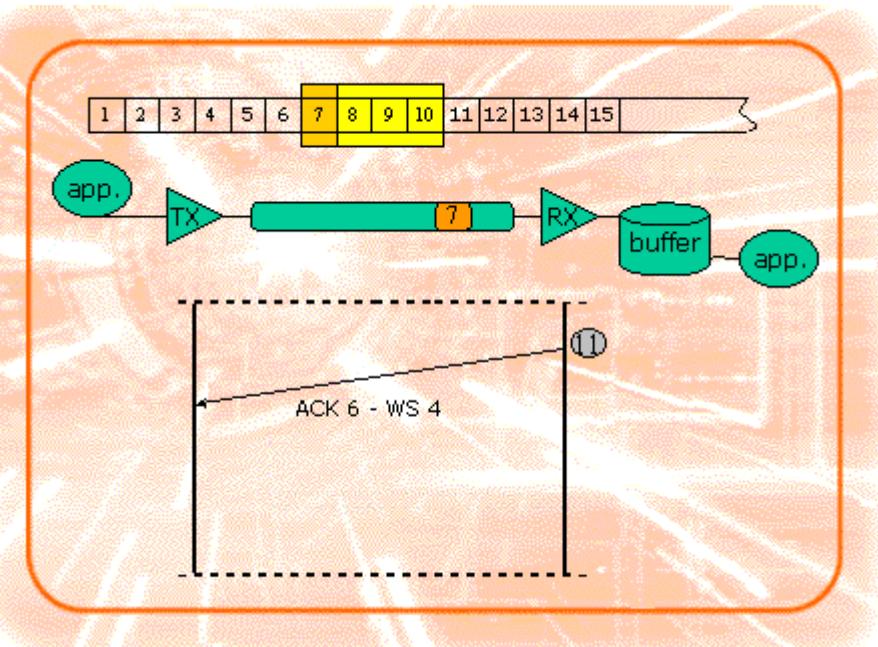


Se envían 3 segmentos más.

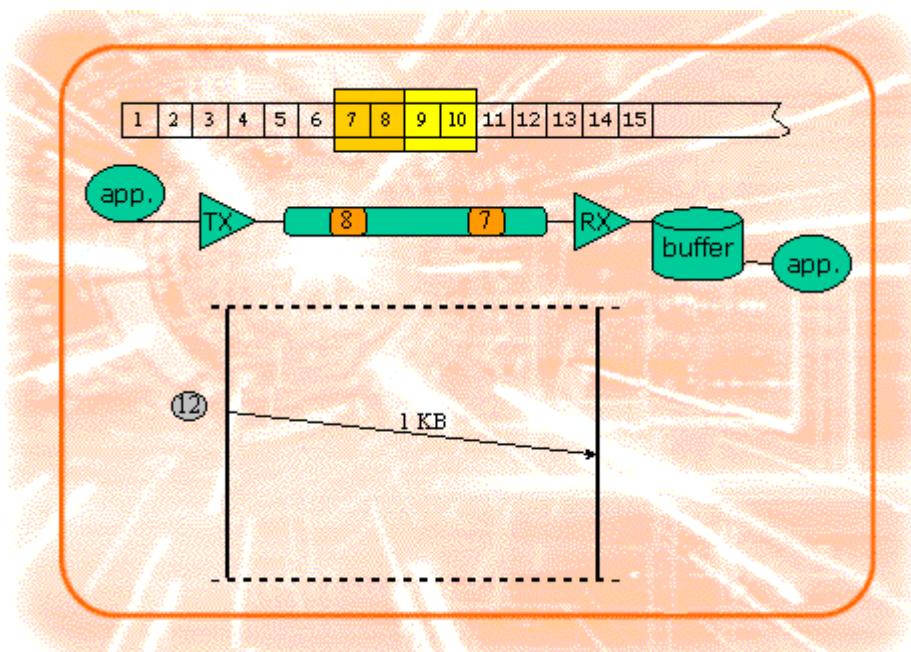


El receptor confirma que ha recibido los segmentos 5 y 6 (con ACK 6) y que ya los ha entregado a la aplicación, vaciando nuevamente el buffer, (con WS 4).

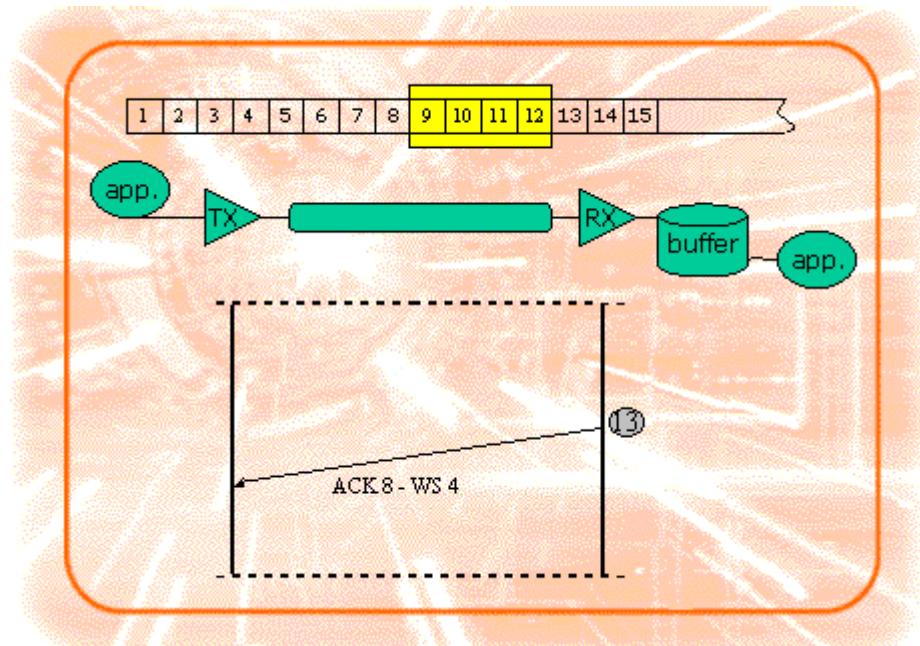
El segmento 7 permanece en el canal.



Se envía otro segmento, el 8; con lo cual, pasamos a tener dos segmentos en el canal, el 7 y el 8.

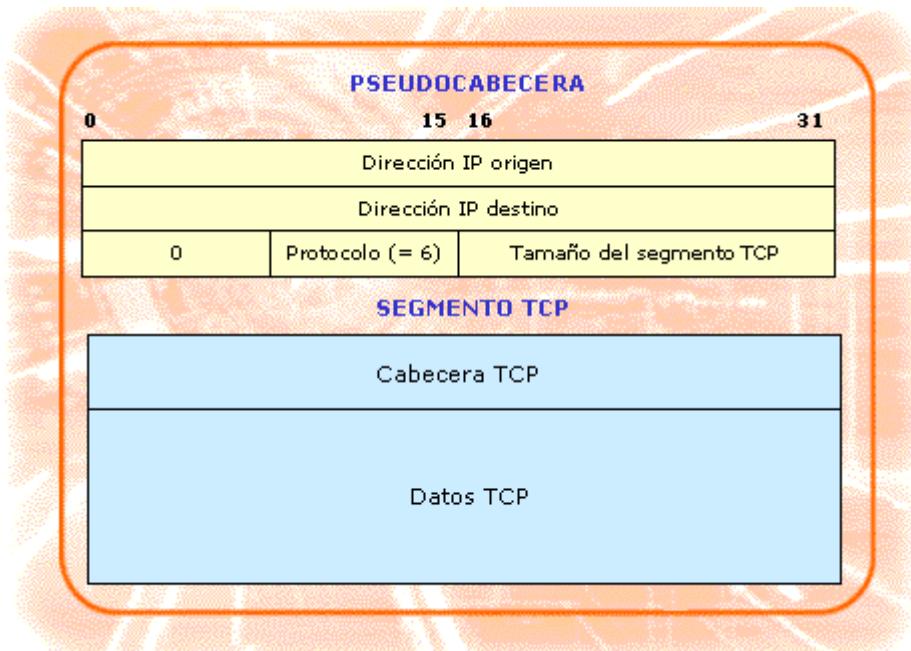


Todos los segmentos son reconocidos y enviados a la aplicación. El buffer de recepción está nuevamente vacío. Por tanto, se vuelve a modificar el tamaño de ventana a 4 Kbytes y se abre en 9 y se cierra en 12.



## Suma de control

El campo Suma de control "cubre" la totalidad del segmento (cabecera + datos) más la pseudocabecera de 96 bits (12 bytes) colocada delante de la cabecera TCP a la hora de calcularlo.



La pseudo-cabecera contiene la dirección IP fuente, la dirección IP de destino, el campo de protocolo del datagrama IP y la longitud del segmento TCP.

En transmisión, estos parámetros son los mismos que se pasan a IP a través de la primitiva SEND.

En recepción, estos parámetros se obtienen mediante la primitiva DELIVER de IP.

TCP incluye esta pseudo-cabecera en la Suma de control para protegerse a sí mismo de una entrega errónea por parte de IP.

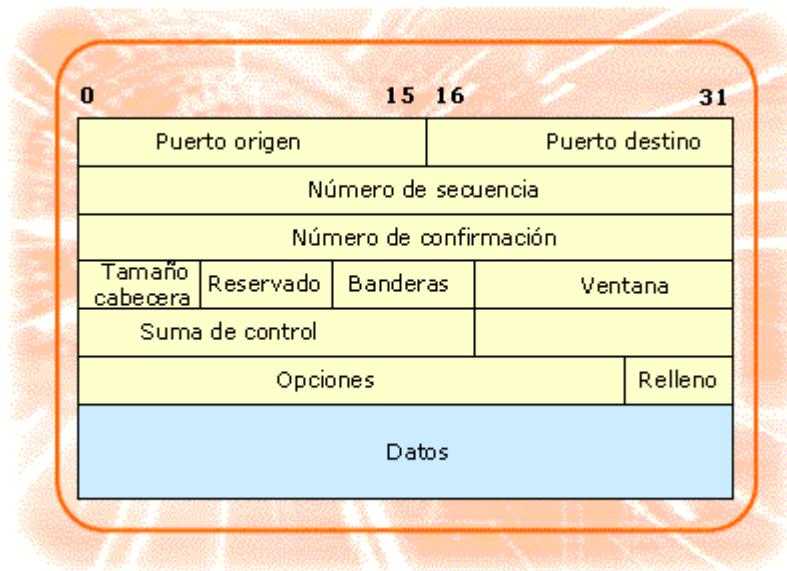
Por ejemplo, un módulo IP puede estar funcionando mal y aceptar datagramas que no van dirigidos a su dirección.

Cuando trate de entregar los segmentos contenidos en dichos datagramas al módulo TCP, éste los rechazará porque el cálculo de la Suma de control que efectúa, basándose en la dirección IP de la máquina receptora, no coincidirá con el "checksum" del segmento.

## Datos urgentes

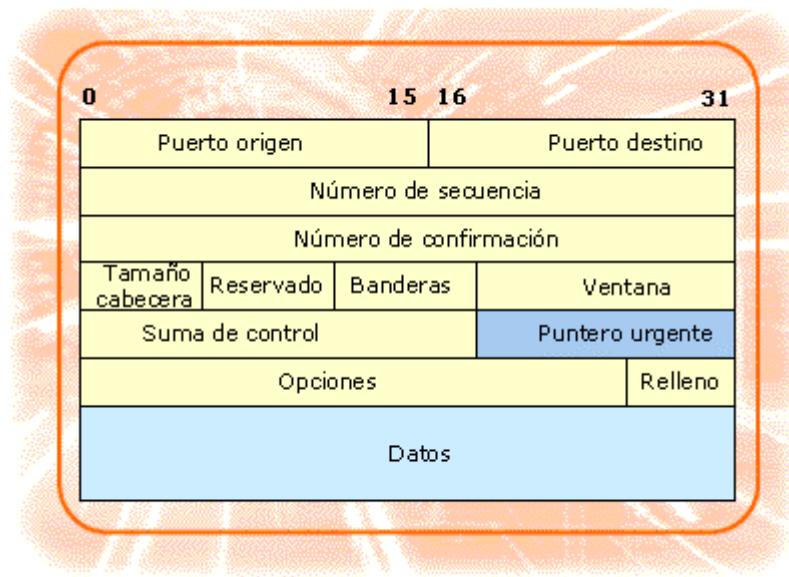
Sabemos que la transmisión de datos de una aplicación se modela como un flujo ordenado de bytes hacia su destino.

Sigamos con el ejemplo anterior de una sesión interactiva y supongamos que un usuario ha pulsado una tecla de aviso de interrupción. La aplicación debería ser capaz de saltarse los bytes intermedios y avisar lo antes posible.



Existe un mecanismo de **Datos urgentes** para marcar un segmento concreto como *urgente*.

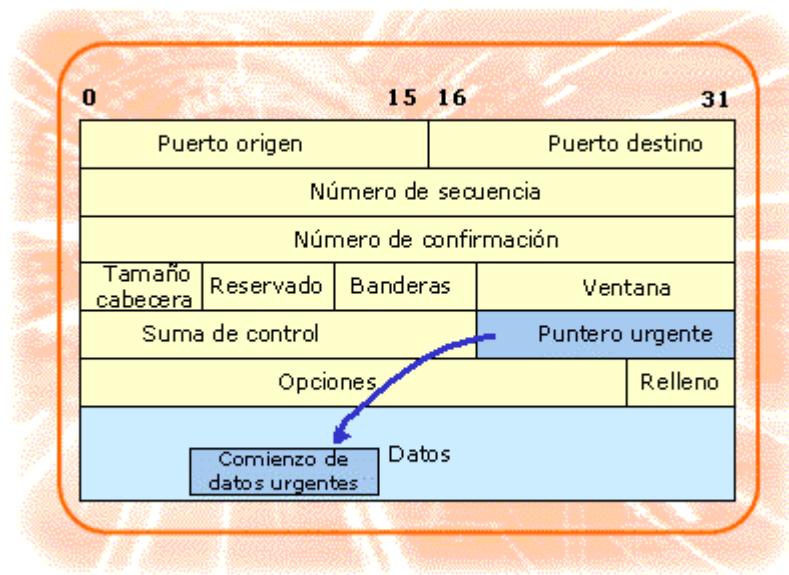
TCP puede avisar al otro extremo de que un segmento contiene datos urgentes, y puede indicar cuáles son los datos.



A veces, a los datos urgentes se les llama datos **fueras de banda**. Este término suele ser confuso. Los datos urgentes se envían por el mismo flujo de datos de TCP. El TCP del otro extremo debe notificar, al programa de aplicación que esté asociado a la conexión, que entre en modalidad urgente.

Después de asimilar todos los datos urgentes, el TCP indica al programa de aplicación que regrese a su modo de operación normal.

El campo **Puntero urgente**, cuando se utiliza, apunta a una posición de la zona de datos del segmento TCP, a partir de la cual, los datos deben ser procesados lo más pronto posible.



## Mecanismo push

- ◆ TCP proporciona un servicio *dúplex* que maneja simultáneamente los dos flujos de datos.
- ◆ TCP intentará crear segmentos de un tamaño adecuado para utilizar eficientemente los servicios de transmisión.
- ◆ Para determinadas aplicaciones, no resultan apropiados trozos de datos grandes y eficientes (aplicaciones interactivas).
- ◆ El mecanismo *push* permite que los datos sean entregados inmediatamente a la aplicación destino.
- ◆ El mecanismo *push* no es aconsejable en todas las aplicaciones (ej. Transferencia de ficheros).

El modelo conceptual de una conexión consiste en que una aplicación envía un flujo de datos a otra aplicación pareja. Al mismo tiempo, recibe un flujo de datos de la otra.

TCP proporciona un servicio *dúplex* que maneja simultáneamente los *dos flujos de datos*.

Normalmente, TCP intentará crear segmentos, conteniendo porciones de datos de las aplicaciones, de un tamaño adecuado para utilizar eficientemente los servicios de transmisión.

Por ejemplo, si estamos utilizando TCP/IP sobre una LAN 802.3, cuya MTU es de 1482 octetos, intentaremos crear segmentos de un tamaño tal que, al añadirle las cabeceras IP, aprovechen al máximo los 1482 octetos que ofrece el protocolo de nivel 2. Por tanto, TCP debería esperar a recoger una cantidad razonable de datos antes de crear un segmento.

Pero, a veces, para una determinada aplicación, no resultan apropiados trozos de datos de tamaño eficiente respecto a la MTU del protocolo de nivel 2.

Por ejemplo, supongamos un programa cliente que ha iniciado una sesión interactiva con un servidor remoto y el usuario ha tecleado un comando y pulsado *retorno de carro*.

El programa cliente quiere que TCP sepa que los datos deberían enviarse al host remoto y entregarse a la aplicación del servidor inmediatamente. Esto es posible con la función *push*.

Si se observa una traza de una sesión interactiva, se podrán observar muchos segmentos que contienen muy pocos datos y, probablemente, se pueda ver una señal de *push* en cada uno de ellos.

No debería utilizarse el mecanismo push durante una transferencia de archivos (excepto para el último segmento), así el TCP podrá empaquetar los datos en segmentos de la forma más eficiente posible.

## Banderas (Flags) de control

- URG
- ACK
- PSH
- RST
- SYN
- FIN

Las banderas son campos de un bit de longitud, cada uno con su significado.

### **URG**

Indica datos urgentes.

### **ACK**

Se envía un reconocimiento.

### **PSH**

El búfer de recepción debe ser borrado.

### **RST**

Indica un error; también se usa para abortar una sesión.

### **SYN**

Establecimiento de conexión.

### **FIN**

Terminación correcta de una conexión.

## Opción de tamaño máximo de segmento



La opción de tamaño máximo de segmento (MSS – *Maximum Segment Size*) se usa para indicar el tamaño del mayor trozo de datos que se puede recibir (y reensamblar) de un flujo.

El nombre resulta un poco equívoco. Normalmente, un segmento se define como la cabecera TCP mas los datos.

A pesar de su nombre, el tamaño máximo de segmento del sistema se define como:

### **El tamaño del mayor datagrama que se puede recibir – 40.**

En otras palabras, el MSS informa del mayor tamaño de carga útil de datos del receptor cuando las cabeceras de IP y de TCP son de 20 octetos, respectivamente.

Si existe cualquier número de opciones de cabecera, hay que restar su tamaño.

Normalmente, los extremos intercambian sus correspondientes valores de MSS junto con el mensaje inicial "SYN" en el establecimiento de una conexión.

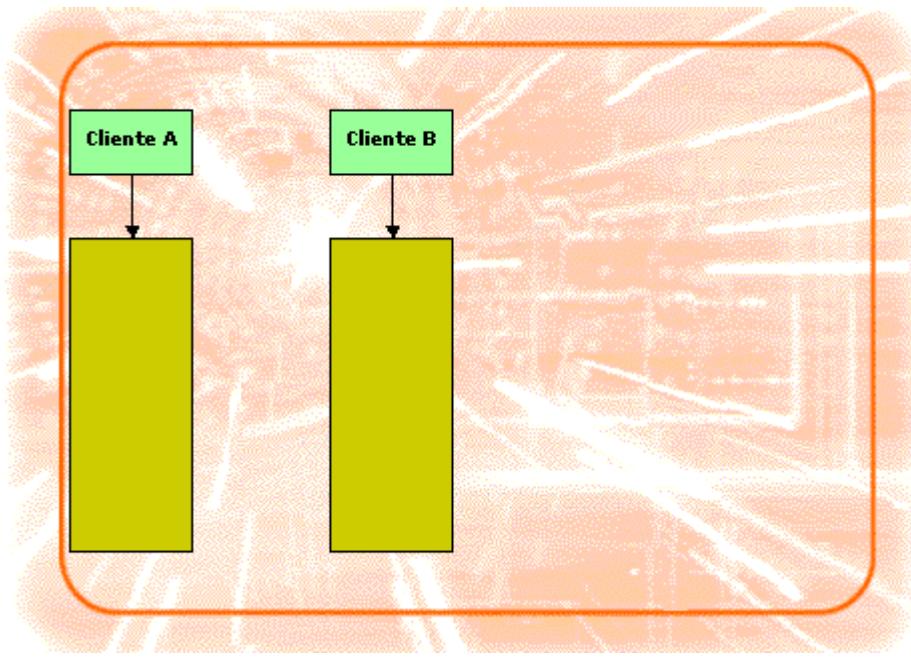
Si el sistema no indica su tamaño máximo de segmento, se supone un valor de MSS, por defecto, de 536 octetos.

### Relleno

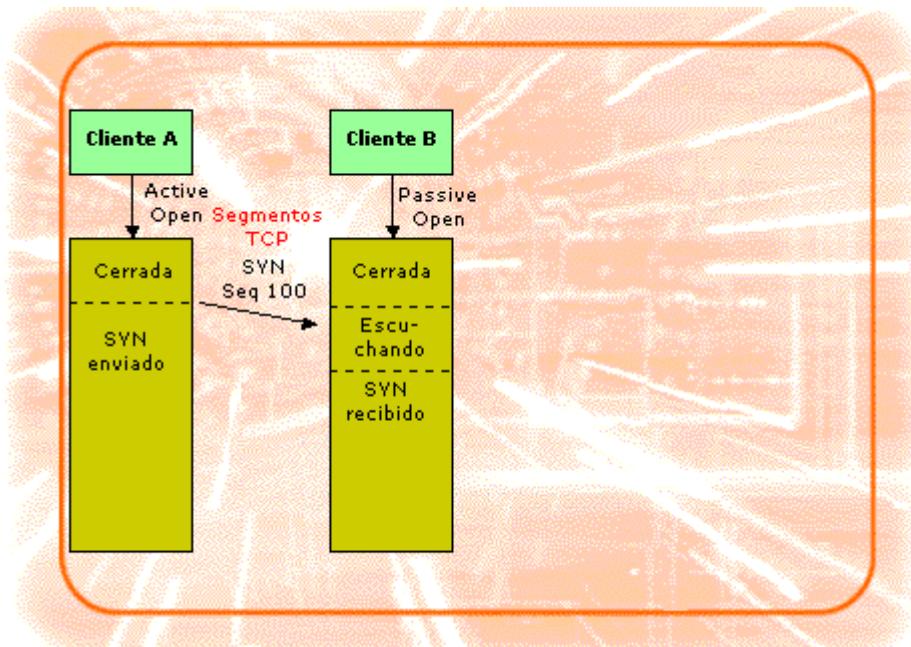
El último campo de la cabecera TCP. Igual que en IP, este campo contiene ceros para asegurar que la cabecera se extiende hasta un múltiplo exacto de 32 bits.

## Establecimiento de la conexión

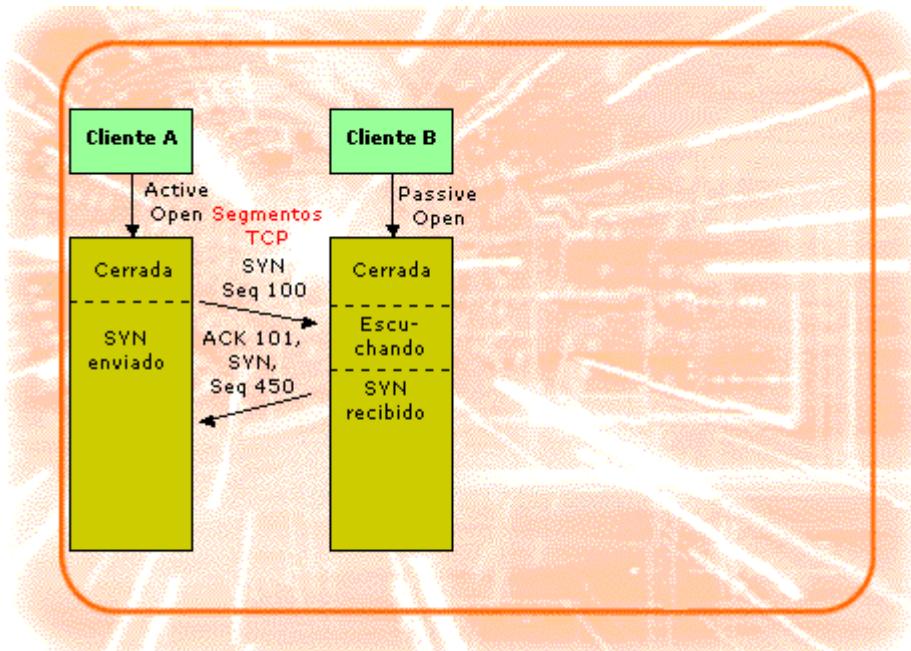
Para el establecimiento de conexión, TCP utiliza un protocolo de tres etapas. El escenario se muestra en el gráfico.



Después de iniciar con un mensaje “active request”, la estación A envía un mensaje con el flag SYN activo.

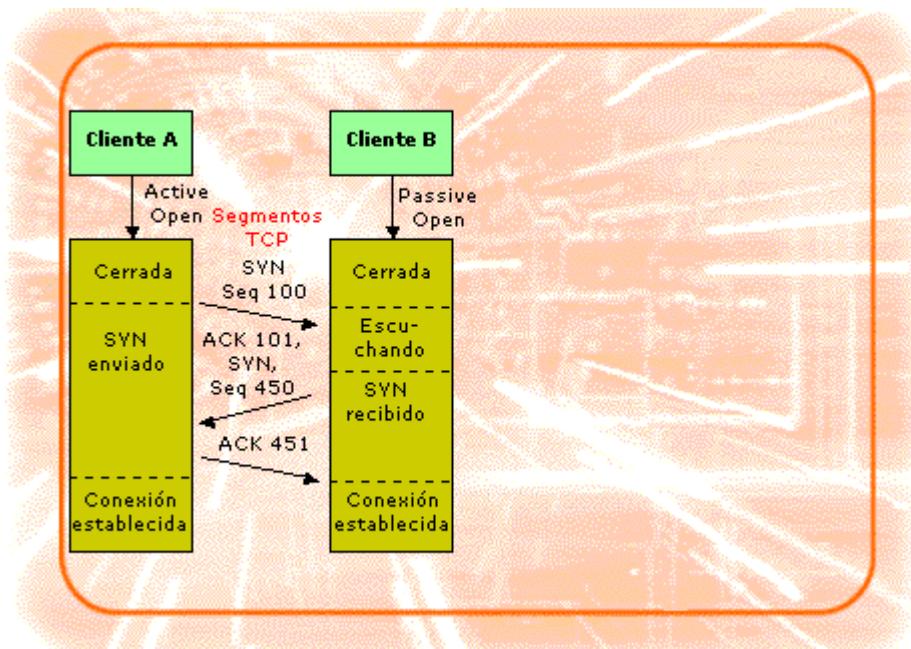


El segundo mensaje tiene los flags SYN y ACK activos, indicando que reconoce el primer segmento SYN así como que continúa el intercambio de señalización (continúa el “handshake”).

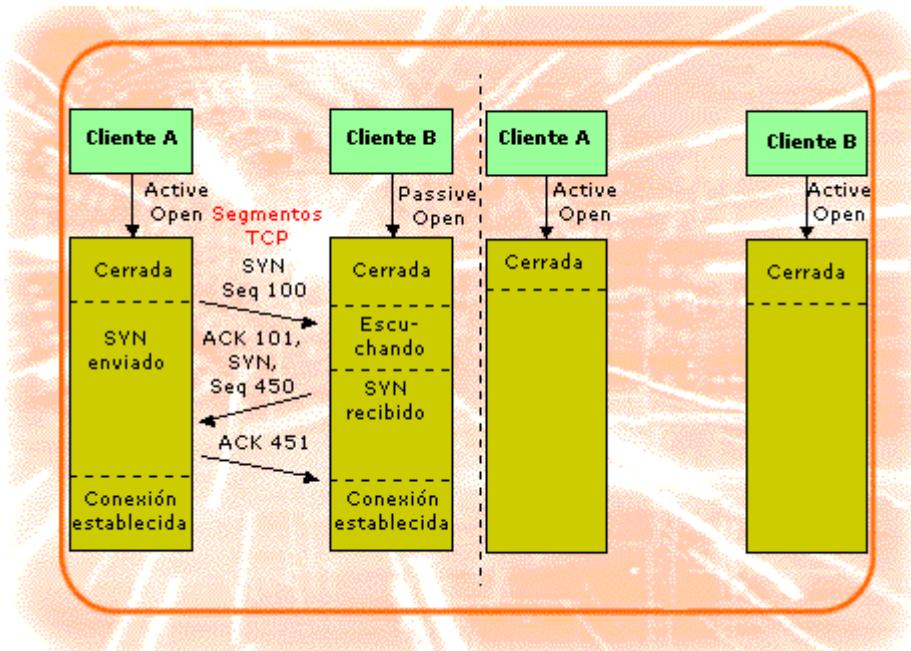


El mensaje final es sólo de “handshake” para indicar reconocimiento al anterior y su utilidad es informar al destino de que ambos lados están de acuerdo en que comience la conexión.

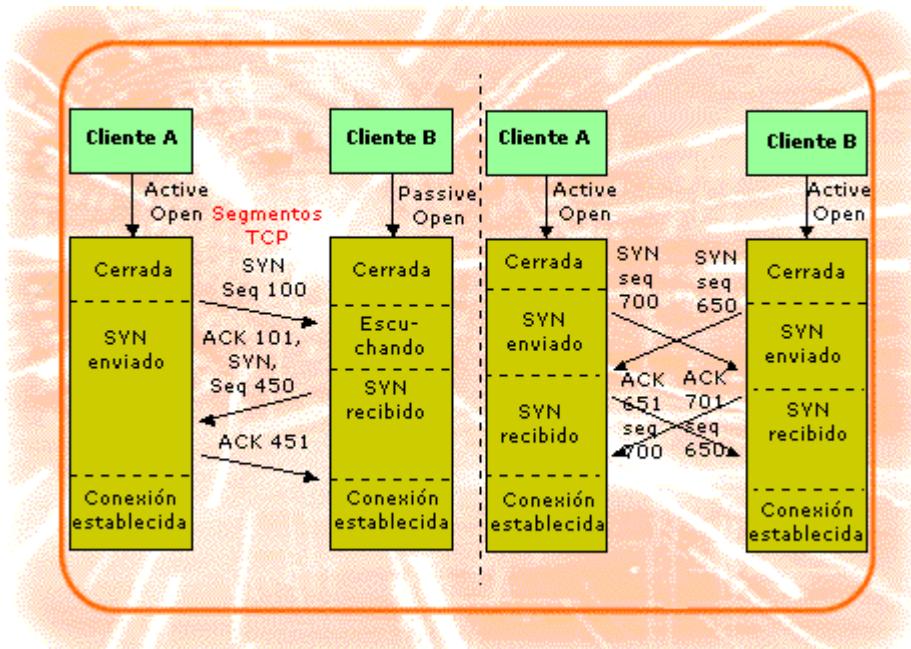
Es posible enviar el valor MSS (“maximum Segment Size”) en el mensaje SYN. El otro lado puede aceptar el valor MSS o, por ejemplo, devolver un valor menor.



Generalmente, el software TCP en una máquina espera pasivamente por el “handshake” y el software TCP de la otra máquina lo inicia. Sin embargo, el “handshake” está diseñado de manera que ambas máquinas puedan iniciar la conexión al mismo tiempo.



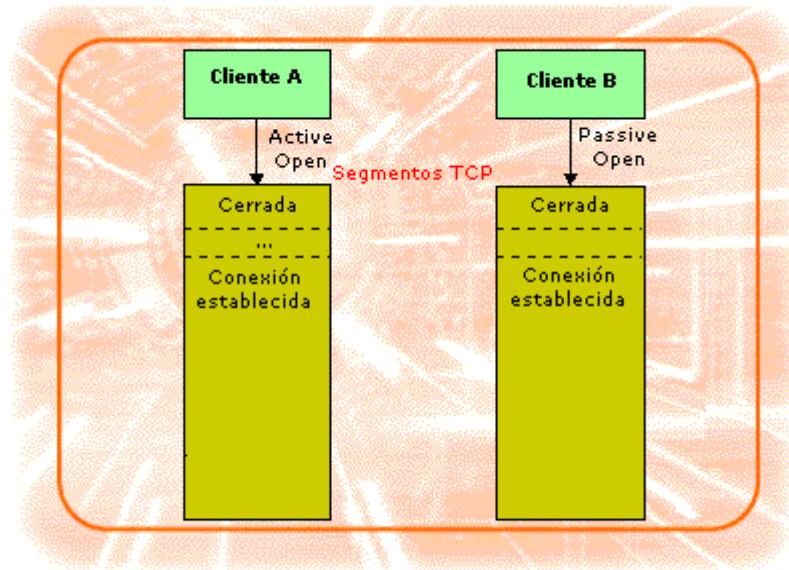
Por ejemplo, dos servidores de correo electrónico contactando uno con otro. Entonces, una conexión puede ser establecida simultáneamente desde cualquiera de los dos lados. Una vez establecida la conexión, los datos pueden fluir igualmente en ambas direcciones.



## Liberación de conexión

El flag RST se usa en situaciones anormales que fuerzan a un programa de aplicación, o al software de red, a romper una conexión.

Cuando el flag RST se recibe activo, el receptor responde inmediatamente abortando la conexión.

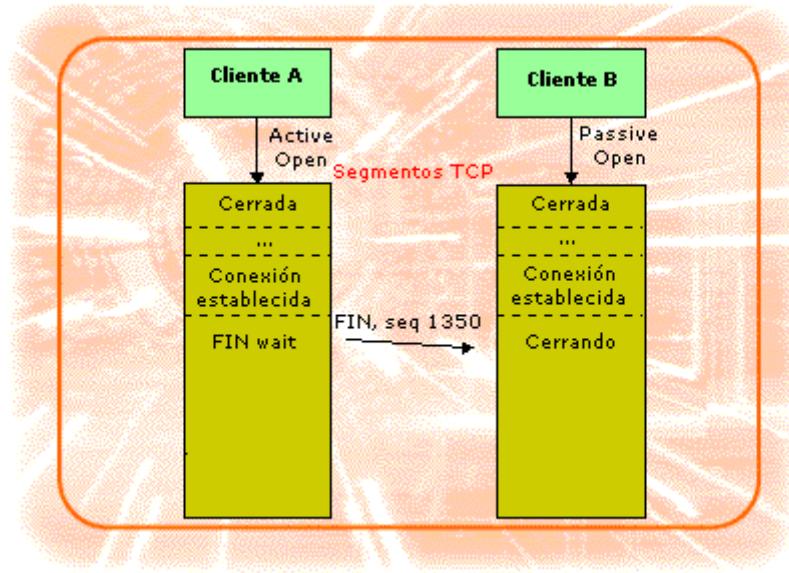


También informa al programa de aplicación de que ha ocurrido un “reset”.

Un aborto significa que la transferencia en ambos sentidos cesa de inmediato, y los recursos de los búfer son liberados.

Como ya hemos mencionado, la función de RESET se usa en situaciones anormales.

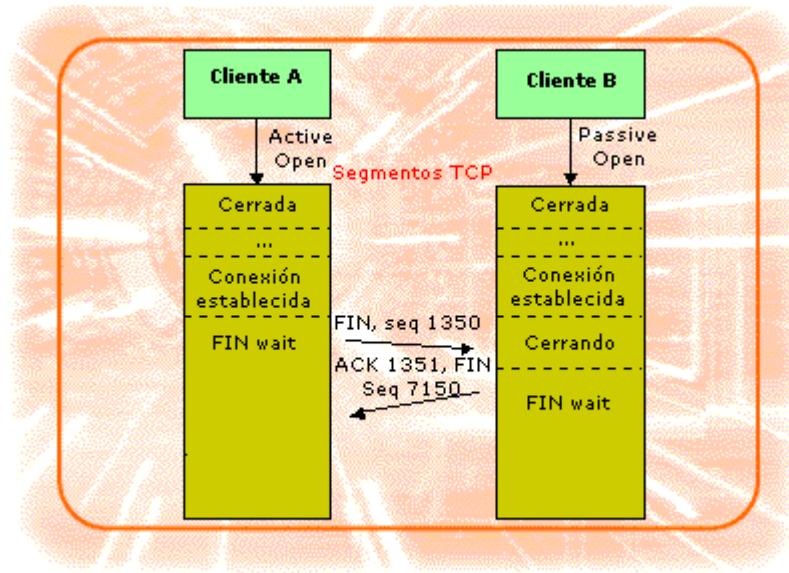
En una desconexión TCP normal, se usa otro mecanismo.



Cuando un programa de aplicación comunica a TCP que no hay más datos que enviar, TCP cerrará la conexión en esa dirección.

Para cerrar su mitad de conexión, el TCP transmisor termina de mandar los datos restantes y, acto seguido, envía un segmento con el flag FIN activo.

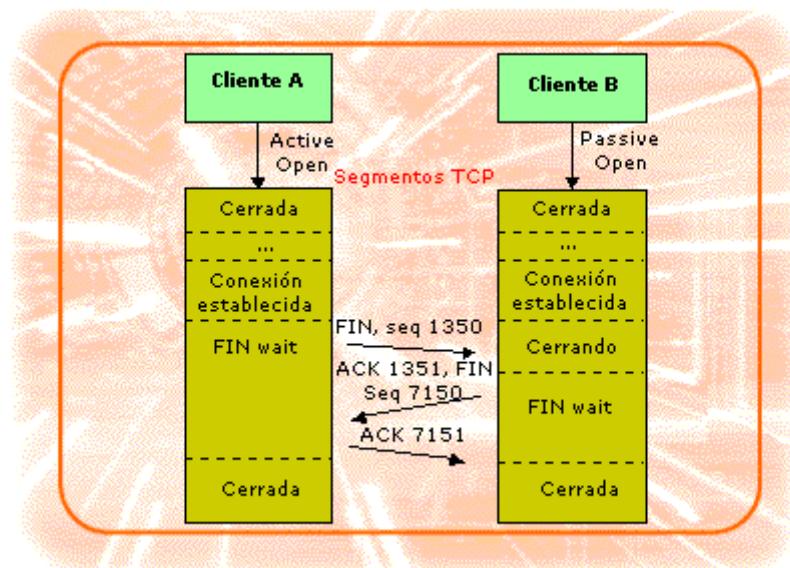
El TCP receptor reconoce el segmento FIN e informa a la aplicación de que no hay más datos disponibles.



Una vez que la conexión ha sido cerrada en un sentido determinado, TCP rechaza cualquier intento de envío de datos en esa dirección.

Mientras tanto, los datos pueden seguir fluyendo en el otro sentido, hasta que el transmisor decida cerrar la conexión de ese sentido.

Cuando ambos sentidos están cerrados, la conexión es eliminada.



## 7- Protocolos de aplicación

En los anteriores capítulos hemos presentado los detalles de la tecnología TCP/IP, haciendo hincapié en los protocolos que proporcionan los servicios básicos. Ahora que conocemos la tecnología básica, podemos examinar los programas de aplicación que se aprovechan del uso cooperativo de una Internet de TCP.

El patrón de interacción primario que se da entre las aplicaciones de cooperación se conoce como paradigma **cliente-servidor**. La interacción cliente-servidor forma la base de la mayor parte de la comunicación por redes y es fundamental, ya que nos ayuda a comprender las bases sobre las que están construidos los sistemas distribuidos.

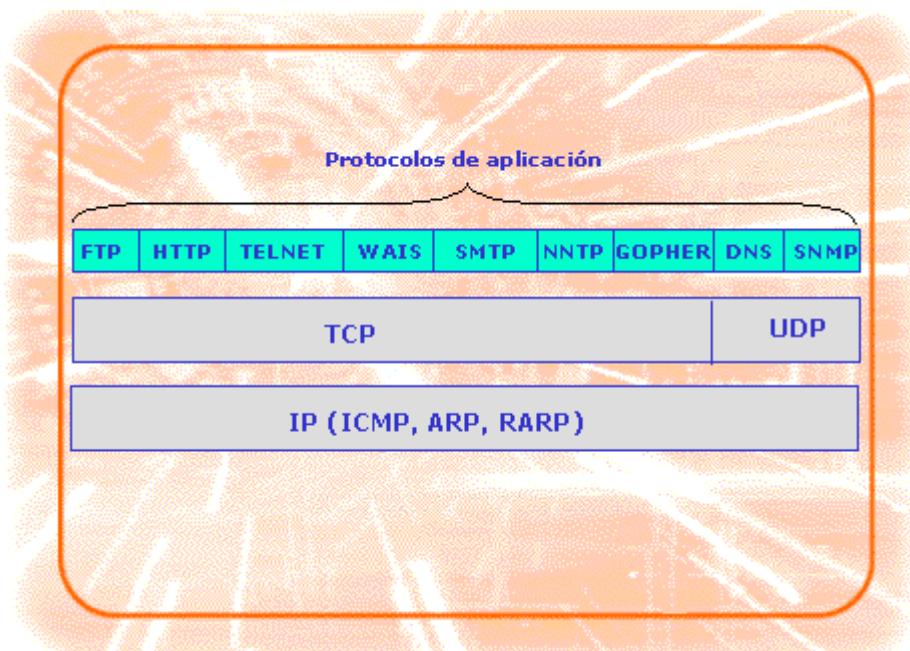
## 1- Protocolo DHCP

Te mostraremos una visión general acerca de los protocolos de aplicación que corren sobre la red TCP/IP, seguro que muchos de ellos te resultan familiares. Después, analizaremos el funcionamiento del protocolo DHCP.

### Introducción

#### Introducción a los protocolos de aplicación

Los protocolos de aplicación forman la capa más alta de la pila de protocolos TCP/IP. Es importante no confundir los **protocolos de aplicación** con las **aplicaciones** que utilizan dichos protocolos, aunque es fácil confundirlos porque muchos de ellos se denominan igual.



Los protocolos de aplicación **permiten la comunicación entre aplicaciones situadas en distintas máquinas** siguiendo el modelo de interacción cliente-servidor.

Utilizan mensajes entre ellos para transmitir las peticiones de los usuarios. Protocolos de aplicación conocidos son: http, ftp, telnet, smtp, etc.

Las aplicaciones son programas de usuario que sirven de interfaz entre el usuario y los protocolos de aplicación.

Aplicaciones:

- los navegadores: Netscape, Internet Explorer, etc
- los interfaces de usuario de correo electrónico: Outlook, Lotus cc:Mail, Eudora, etc
- los interfaces de usuario de gestión de red: VUE, etc.

Además, algunas aplicaciones sirven de interfaz a varios protocolos de aplicación.

Por ejemplo, las aplicaciones de correo electrónico permiten acceder a protocolos específicos de correo electrónico, como SMTP; pero también permiten acceder al protocolo de noticias en red (NNTP).

Otro ejemplo es Netscape, que puede utilizarse con el protocolo HTTP para obtener "páginas web"; con el protocolo SMTP para la trasferencia de correo, con TELNET para la conexión con máquinas remotas, etc.

## Protocolo de aplicación DHCP

- ◆ La gran expansión de la conectividad IP exige un gran esfuerzo a los administradores de redes.
- ◆ Se hace necesario el uso de herramientas automatizadas para la configuración en red de los equipos.
- ◆ La forma más efectiva es almacenar los parámetros de configuración en estaciones de arranque de red.
- ◆ El protocolo de configuración dinámico de host DHCP es el mecanismo que permite a los clientes configurarse automáticamente.

Uno de los cambios más destacables del uso de los ordenadores en los últimos años ha sido la expansión de la conectividad de red con TCP/IP. La infraestructura necesaria para soportar el crecimiento de la red, routers, bridges, comutadores y concentradores, ha crecido a una velocidad similar.

El personal técnico lucha para dar respuesta a las nuevas demandas de conectividad, así como a los frecuentes cambios, movimientos y reconfiguraciones de red que caracterizan el entorno actual.

Estas circunstancias han generado una necesidad de mecanismos que permitan automatizar la configuración de nodos y la distribución del sistema operativo y del software en la red.

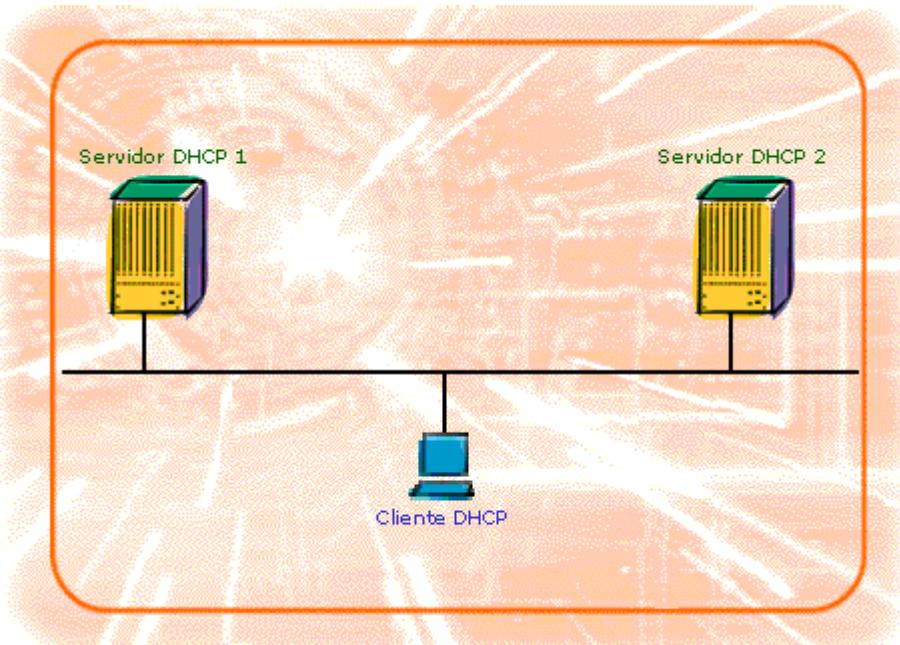
La forma más efectiva de conseguirlo es almacenar los parámetros de configuración e imágenes del software en una o más estaciones de arranque de red. Al arrancar, los sistemas interactúan con un servidor de arranque, recogen los parámetros de arranque y, opcionalmente, descargan el software apropiado.

El protocolo DHCP ("Dynamic Host Control Protocol") permite automatizar completamente la asignación de direcciones IP, así como otros muchos parámetros de red (máscara de red, dirección IP de la pasarela de salida de la red, etc).

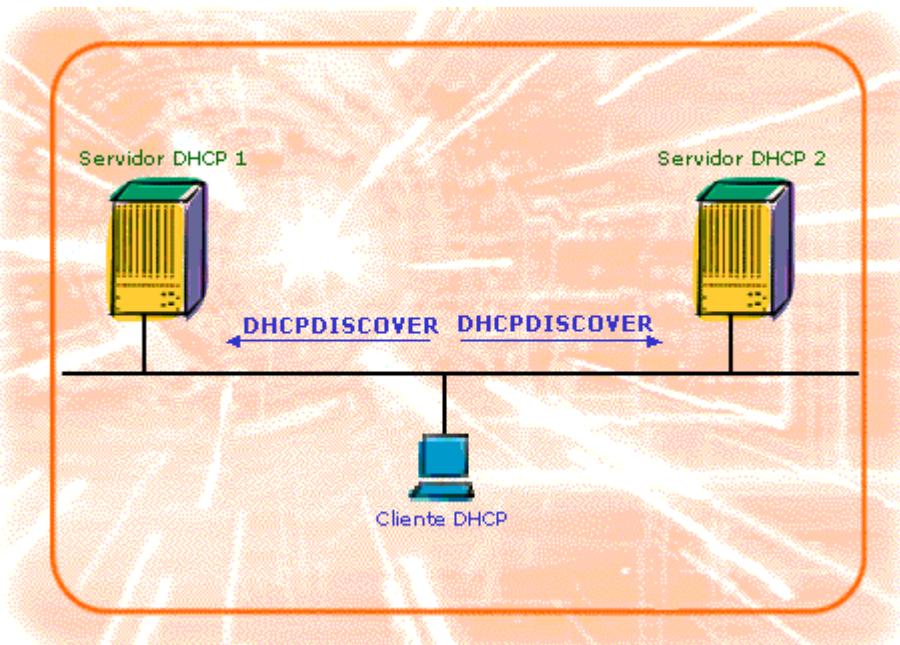
La asignación de estos parámetros a un cliente DHCP se hace durante un tiempo limitado, hasta que el cliente los devuelve y quedan libres para ser asignados a otro cliente.

## Escenario típico DHCP

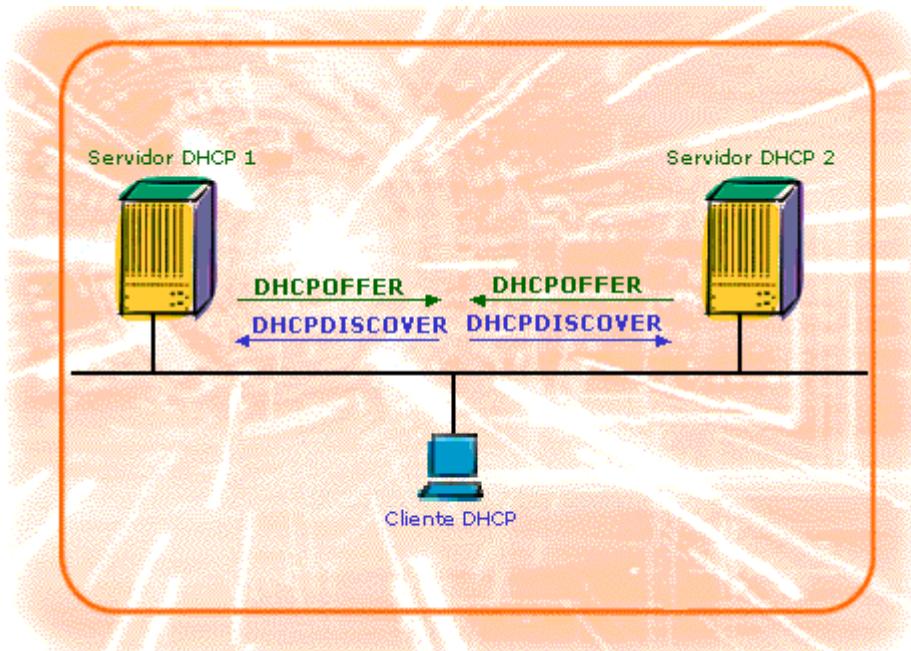
A continuación se describe un ejemplo de interacción inicial entre cliente y servidor:



1. El cliente difunde un mensaje de búsqueda **DHCPDISCOVER** ("broadcast" IP) para descubrir uno o más servidores.



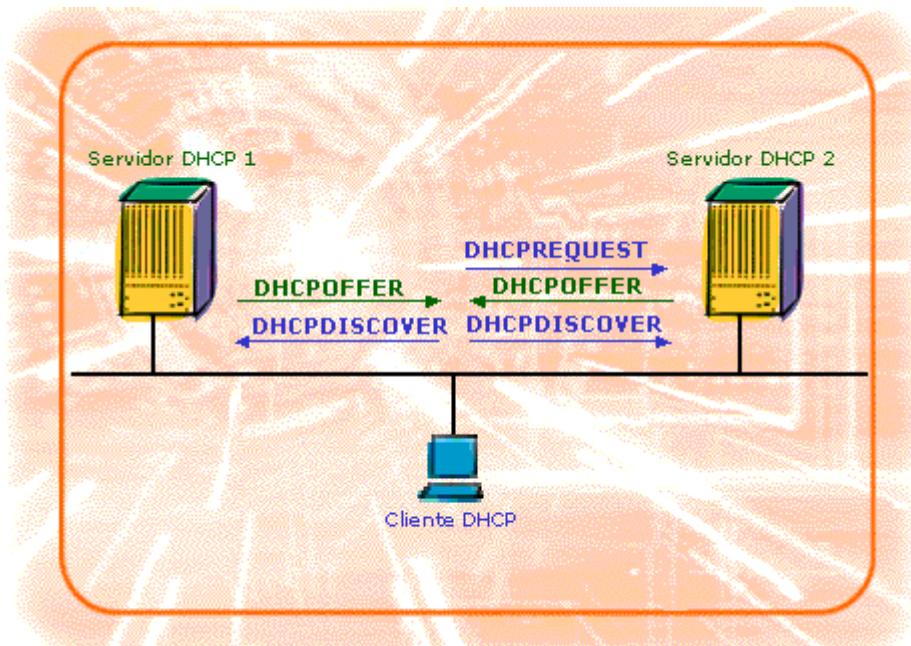
2. Puede haber varios servidores que respondan al cliente (también por mensajes de "broadcast" IP). Así, el cliente espera hasta que ha recibido una o más respuestas **DHCPOFFER**.



Cada respuesta incluye una dirección IP, una Máscara de subred, una fecha de expiración del alquiler, la identidad del servidor (Identificador del servidor DHCP) y algunos parámetros de configuración para el cliente.

3. De acuerdo con el contenido de las respuestas, el cliente selecciona el servidor que quiere usar.

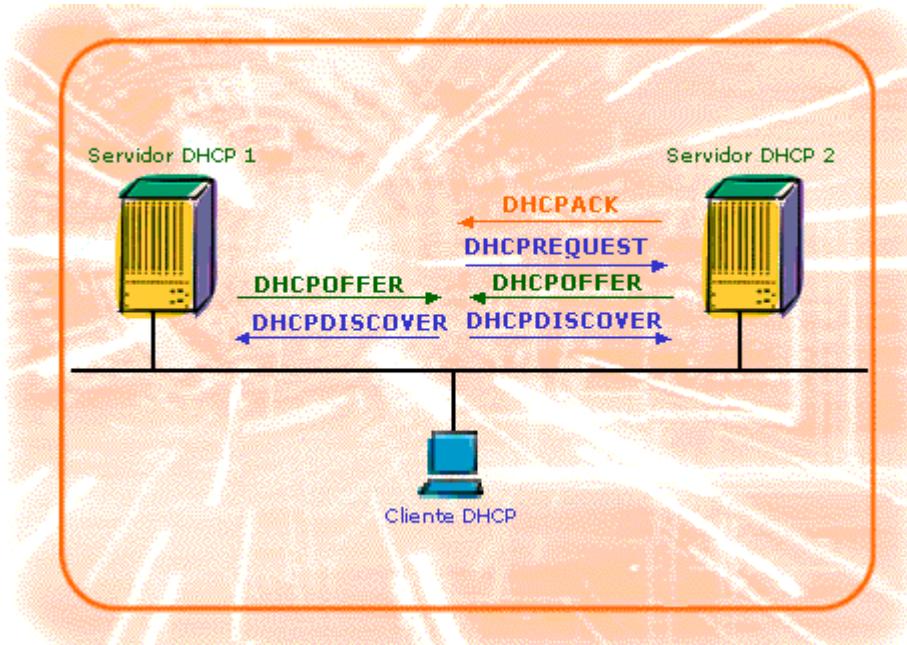
El cliente difunde ("broadcast" IP) una petición **DHCPREQUEST** con el identificador del servidor en el campo Identificador del servidor DHCP.



El mensaje del cliente puede incluir una opción Lista de petición de parámetros de DHCP, que indica que el cliente desea datos de configuración adicionales.

4. El servidor seleccionado guarda el enlace para este cliente en memoria permanente, con un índice de clave apropiado.

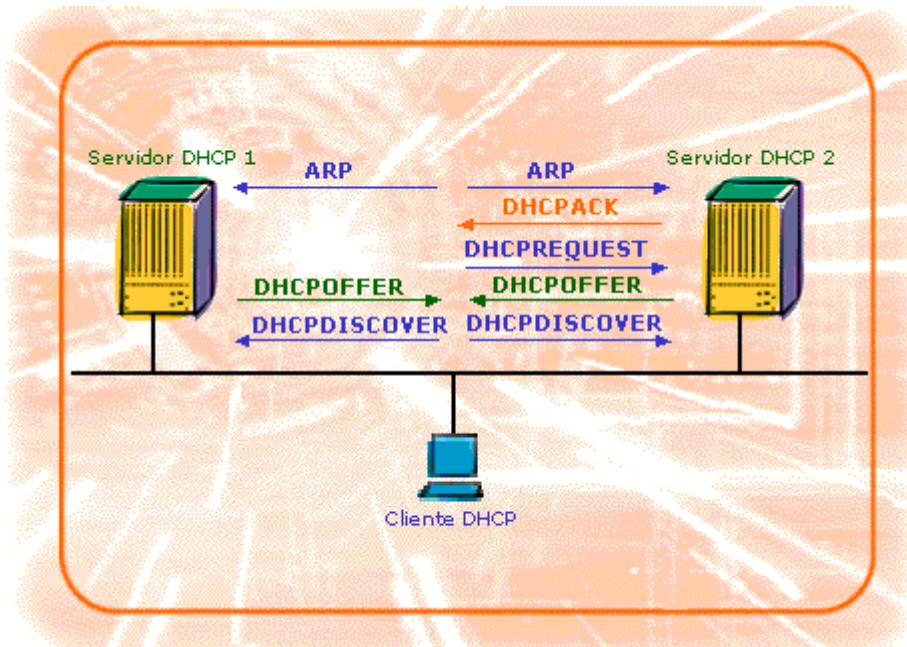
Asimismo envía ("broadcast" IP) los parámetros solicitados al cliente en un mensaje **DHCPACK**.



A partir de ese momento, el cliente ya tiene dirección IP y todos los parámetros de red necesarios.

Cualquier mensaje IP de/hacia el cliente llevará las direcciones IP fuente/destino adecuadas, y no se usará más la difusión ("broadcast" IP).

5. El cliente debería usar una petición **ARP** pidiendo que alguien le devuelva la dirección MAC asociada a la dirección IP que le acaba de ser concedida.



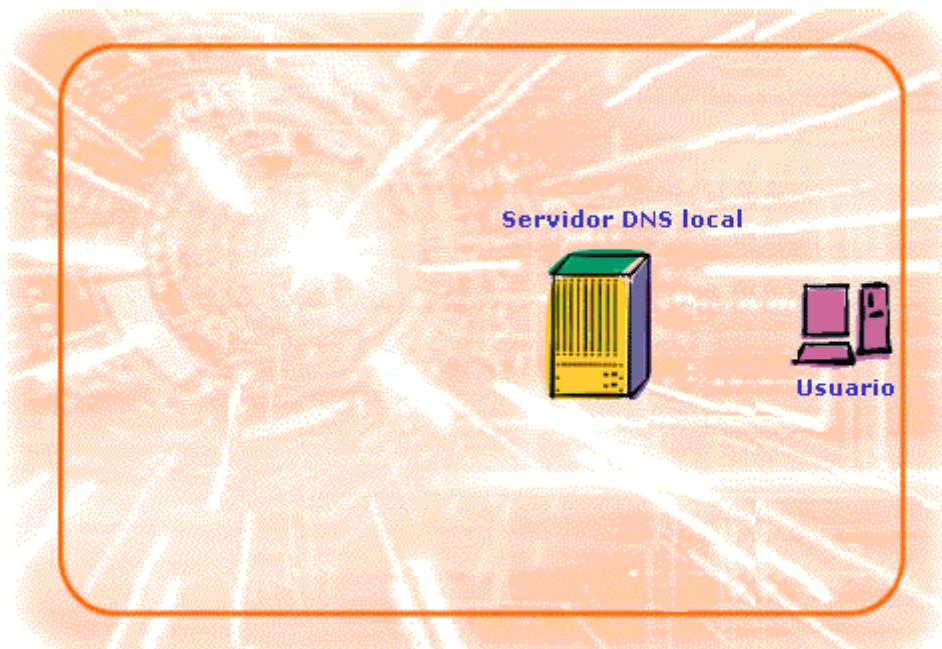
Como nadie contestará, ésta es la manera de verificar que nadie más, en la red, tiene la dirección IP que el servidor DHCP le acaba de conceder.

## 2- DNS

Al acceder a una página web a través de Internet, realmente estamos accediendo a la máquina que la guarda, pero, las máquinas tienen "nombres" determinados por la dirección IP que tienen asignada. ¿Cómo sabe nuestro navegador cuál es la IP que corresponde a la dirección www.fycsa.es?

### DNS

Para evitar que los usuarios tengan que recordar direcciones IP, cualquier servicio en Internet puede invocarse utilizando un nombre propio y único. Por ejemplo, el servidor WWW de Fycsa suele ser invocado mediante el nombre "www.fycsa.es", en vez de utilizando su dirección IP: 194.224.48.191.

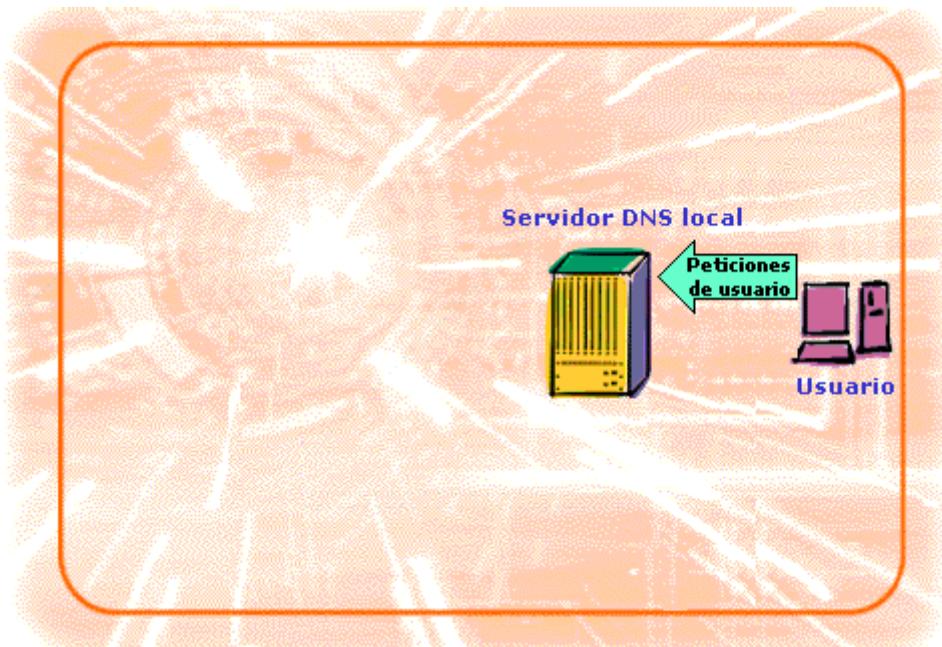


Este nombre se llama **nombre de dominio** del servidor o del proveedor de contenidos y es introducido por el usuario como parte de la URL.

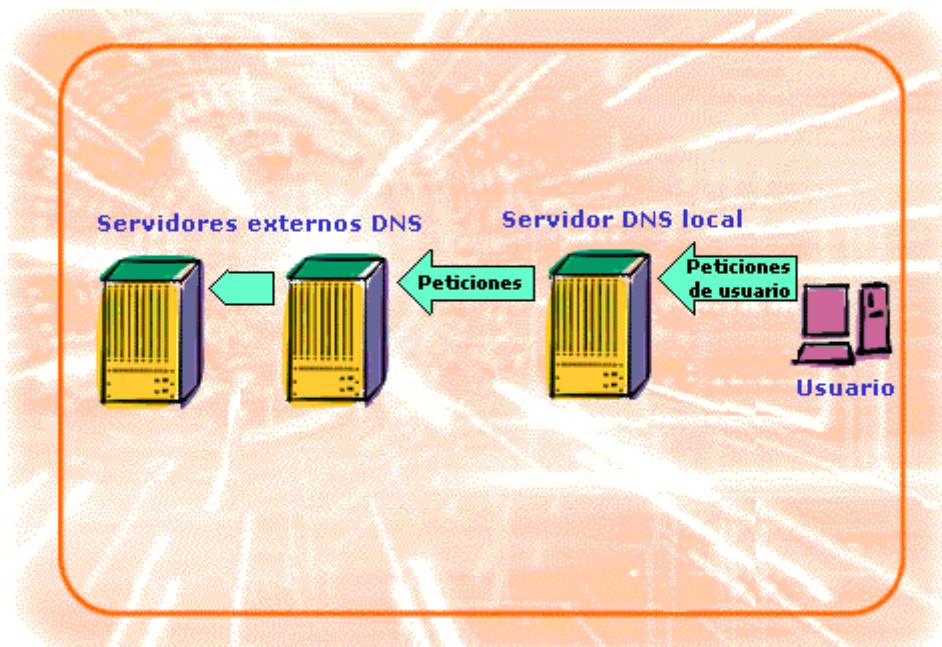
Como puede comprenderse, es imposible almacenar (y gestionar) en todos los "hosts" de Internet todos los nombres con sus correspondientes direcciones IP. En lugar de eso, se utiliza una aproximación de bases de datos distribuidas: el **DNS**.

Cuando un usuario invoca al DNS se desarrolla la siguiente secuencia:

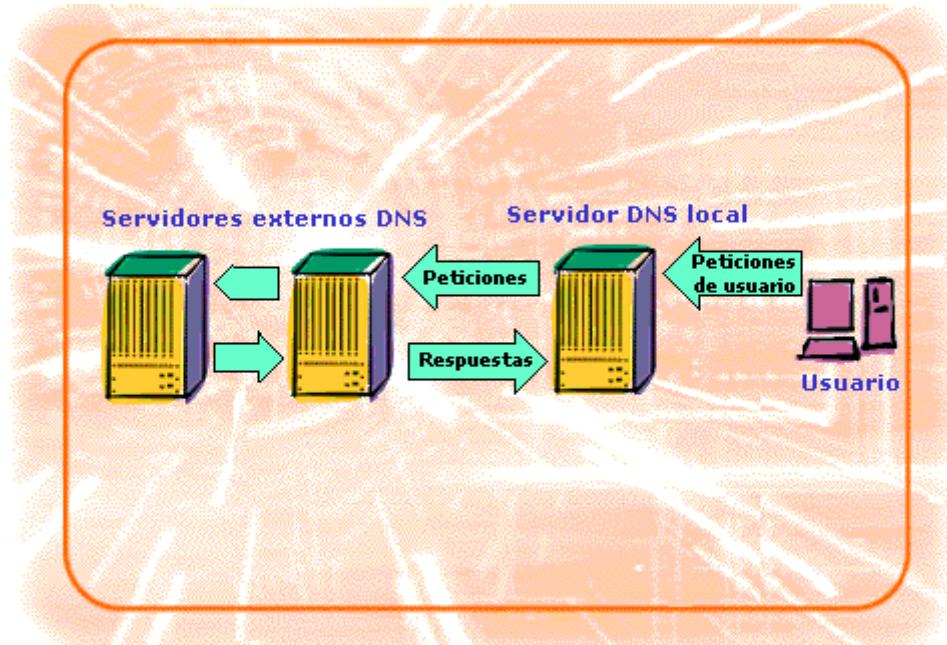
El usuario envía un mensaje DNS (el cual contiene el nombre que se quiere traducir, por ejemplo, www.fycsa.es) a su servidor local DNS. La dirección IP de este servidor local tiene que haber sido configurada con anterioridad en el ordenador del usuario. El servidor local DNS es responsabilidad del ISP ("Internet Service Provider").



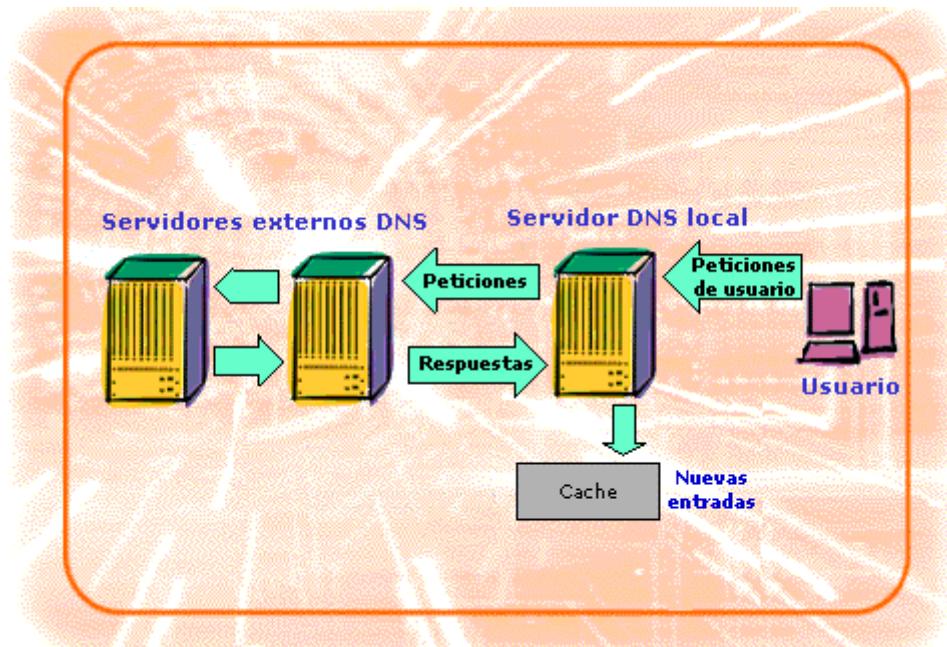
Si el nombre es desconocido por el "DNS database resolver" del servidor local DNS y no está tampoco en su memoria cache, el servidor local DNS enviará la petición a un servidor de nombres externo. Este servidor puede, a su vez, volver a enviar la petición a otro servidor externo, etc; hasta que alguien resuelva la traducción.



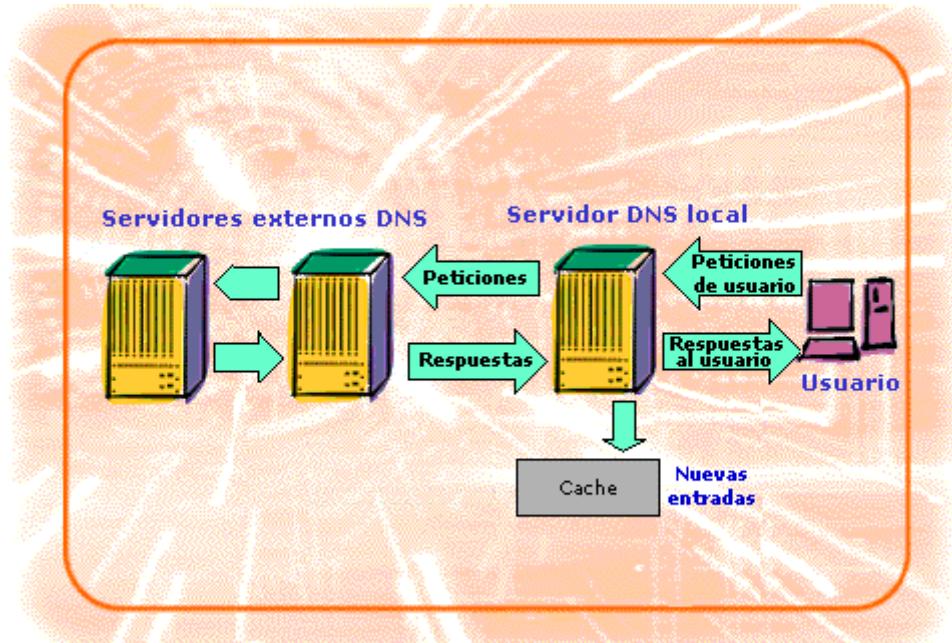
La respuesta (la dirección IP 194.224.48.191) se devuelve al Servidor Local DNS.



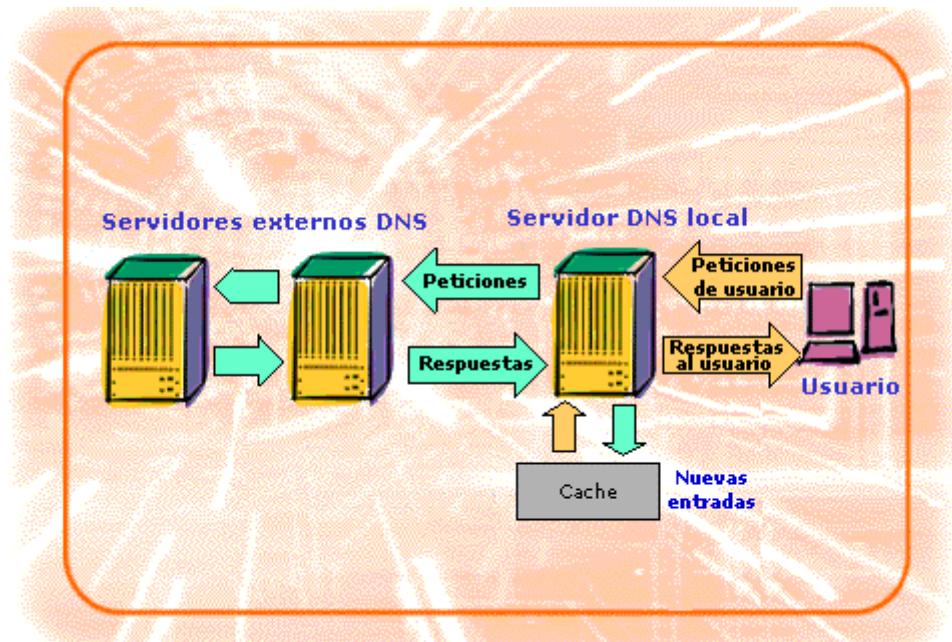
El Servidor Local DNS almacenará en caché la respuesta (con objeto de acelerar nuevas peticiones de traducción del mismo nombre) por un periodo limitado de tiempo.



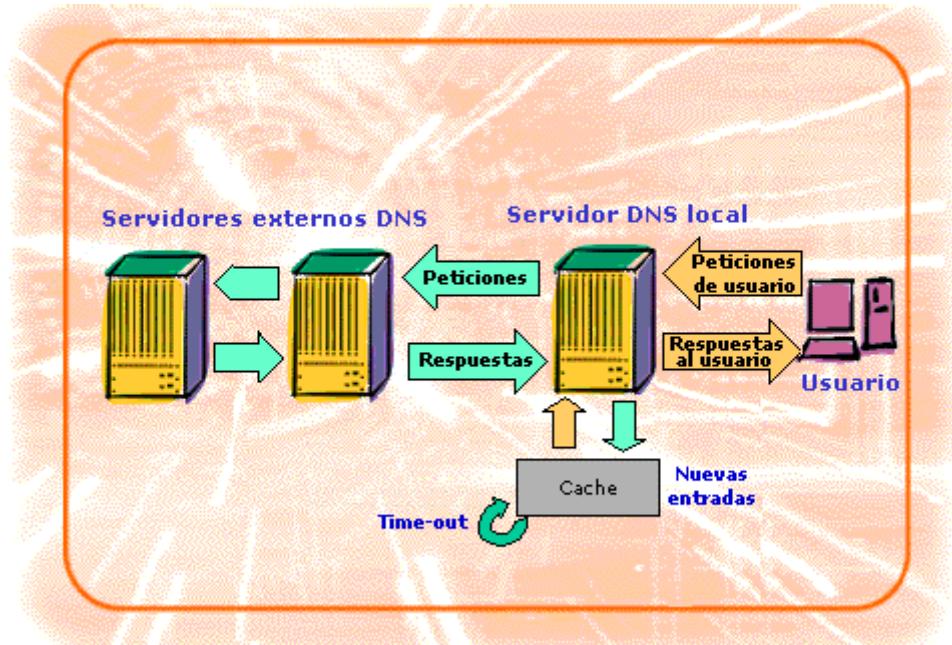
La respuesta (la dirección IP 194.224.48.191) se devuelve al programa de usuario.



Nuevas peticiones de traducción del dominio www.fyrsa.es, del mismo usuario o de cualquier otro que consulte al mismo Servidor Local DNS, serán resueltas con rapidez gracias a los contenidos almacenados en caché.

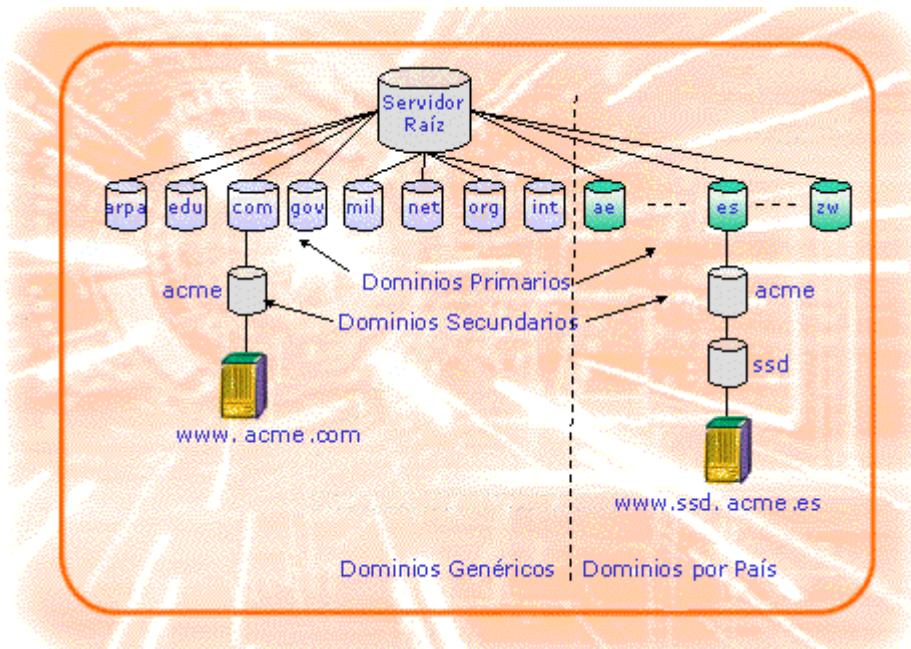


Pasado un tiempo sin peticiones de dicho nombre, se elimina de la caché.



## Organización jerárquica del DNS

La forma más sencilla de entender cómo trabaja un servidor de dominio es imaginándolo como una estructura de árbol donde se pone de manifiesto el carácter jerárquico de la organización de nombres.



La raíz del árbol es un servidor que reconoce el dominio de nivel superior (.com, .edu, .es, etc) y sabe qué servidor resuelve cada dominio.

En el siguiente nivel, un conjunto de servidores de nombre proporciona respuestas para los subdominios que cada uno controla (por ejemplo, el servidor de nombres del dominio .es, conoce y autoriza al servidor de nombres del subdominio .fyrsa.es).

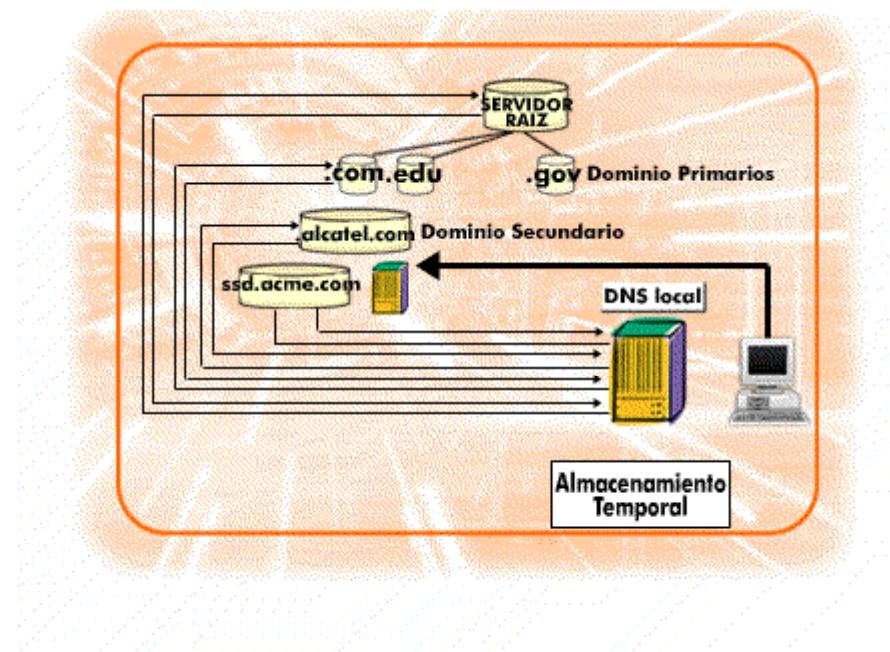
Siguiendo con el ejemplo, el servidor de nombres del dominio .fyrsa.es gestiona los nombres de los subdominios de tercer nivel que dependen de él, como .ssd.acme.es.

El árbol conceptual continúa con un servidor en cada nivel para el que se ha definido un subdominio.

Los enlaces en el árbol conceptual no indican conexiones de red física. Tan sólo muestran qué otros servidores de nombres conoce y contacta un determinado servidor.

El árbol de servidores es sólo una abstracción. Hay que destacar que, en la práctica, los distintos servidores pueden estar ubicados en diferentes ciudades.

## Proceso DNS en estructura jerárquica



Según lo expuesto hasta ahora sobre la estructura jerárquica de los nombres de dominio y de los servidores que traducen dichos nombres, el funcionamiento de un proceso DNS sería como sigue:

1. El usuario solicita la resolución del nombre `www.ssd.acme.com`.
2. El servidor DNS local (que no conoce la existencia de ese nombre) contacta con uno de los nueve servidores raíz (éste servidor debe conocer al menos la dirección IP de uno de los servidores raíz). El servidor raíz envía la dirección IP del servidor de nivel superior del dominio ".com".
3. El DNS local contacta con el servidor del dominio ".com". El servidor del dominio ".com" conoce la dirección DNS del servidor responsable del dominio `<.acme.com>` y se la envía al servidor DNS local.
4. El DNS local contacta con el DNS del dominio `<.acme.com>` y, éste último, le envía la dirección IP del dominio `<ssd.acme.com>`.
5. El DNS local contacta con el DNS del dominio `<ssd.acme.com>` y, éste último, le resuelve la dirección que corresponde a la máquina `<www.ssd.acme.com>`.
6. El DNS local almacena en memoria temporal la dirección de la máquina para agilizar posteriores consultas.
7. El DNS local envía la dirección IP de la máquina solicitada al cliente DNS que se la pidió.
8. El usuario (cliente DNS) ya está en condiciones de contactar directamente con la máquina `<www.ssd.acme.com>`, porque ya conoce su dirección IP.

Por tanto, vemos que en cada nivel de subdominio necesitamos un servidor de nombres que controle los nombres que dependen de ese subdominio. Esto podría dar lugar a árboles con muchos niveles, lo que se traduciría en muchas consultas desde un cliente a los múltiples servidores de cada subdominio, lo cual no sería muy eficiente.

En la práctica, el árbol de servidores contiene pocos niveles pues un solo servidor físico puede contener toda la información para partes extensas de una jerarquía de nombres.

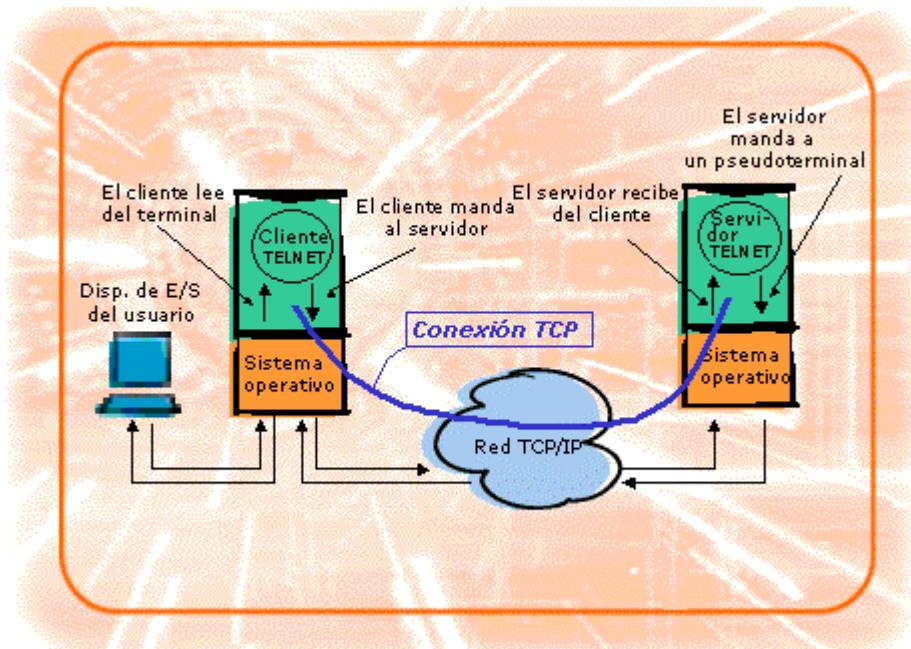
En particular, las organizaciones, a menudo, reúnen información de todos los subdominios desde un solo servidor. Incluso, los servidores raíz suelen contener información de los dominios de dichas organizaciones, con lo que la mayoría de las peticiones se resolverán entre el servidor raíz y el de la organización aludida por la petición.

### 3- TELNET y FTP

Trataremos en esta sección dos de los protocolos más comúnmente usados por los usuarios de Internet, el protocolo de acceso remoto TELNET y el de transferencia de ficheros FTP.

#### TELNET

El conjunto de protocolos TCP/IP incluye un protocolo de terminal remoto sencillo, llamado TELNET. TELNET permite al usuario de una localidad establecer una conexión TCP con un servidor de acceso a un terminal remoto.



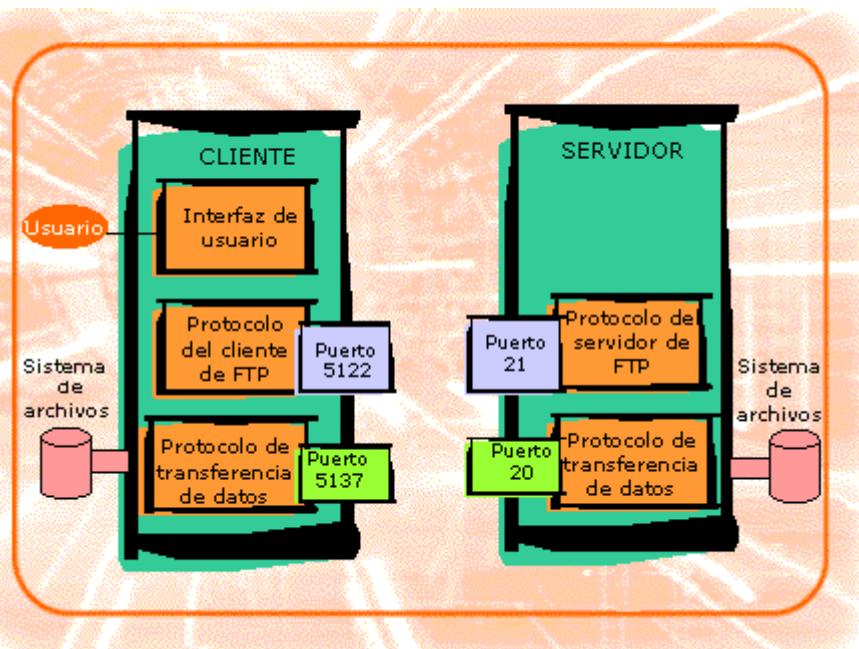
TELNET transfiere después las pulsaciones de teclado, directamente, desde el teclado del usuario, al terminal remoto, tal y como hubiesen sido hechas en un teclado directamente unido a la máquina remota.

TELNET también transporta la salida de la máquina remota, de regreso a la pantalla del usuario.

El servicio se llama "transparente" (transparente) porque da la impresión de que el teclado y el monitor del usuario están conectados de manera directa a la máquina remota.

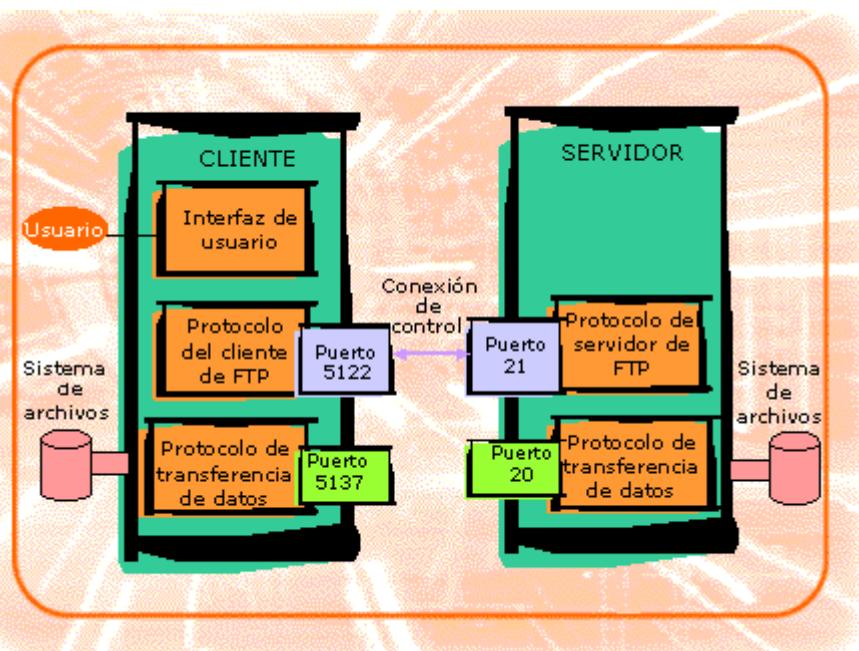
### Transferencia de archivos: FTP

El protocolo de transferencia de archivos (FTP, "File Transfer Protocol") permite a los usuarios copiar archivos de un sistema a otro, ver listados de directorios y realizar tareas normales, como cambiar de nombre o borrar archivos.



### Modo de funcionamiento de FTP

Un usuario interacciona con un proceso del cliente local de FTP. El software del cliente local entabla una conversación formal con el proceso de servidor remoto de FTP a través de una **conexión de control**.

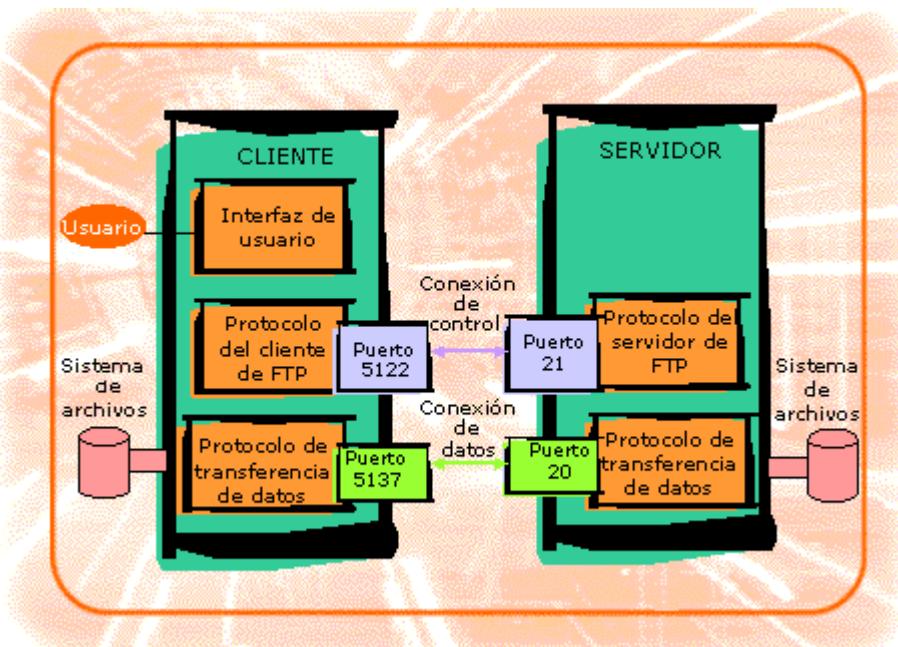


Cuando el usuario final introduce un comando de transferencia o de gestión de archivos (por ejemplo, crear un directorio en la máquina remota para copiar después en él un archivo), el comando se traduce a una de las órdenes especiales de la conexión de control.

Es decir, el cliente envía comandos al servidor a través de la conexión de control y el servidor envía respuestas de vuelta a través de la misma. El servidor utiliza el puerto 21 para su extremo de la conexión de control.

Si el usuario pide una transferencia de archivos, se abre una **conexión de datos** independiente y el archivo se copia a través de dicha conexión.

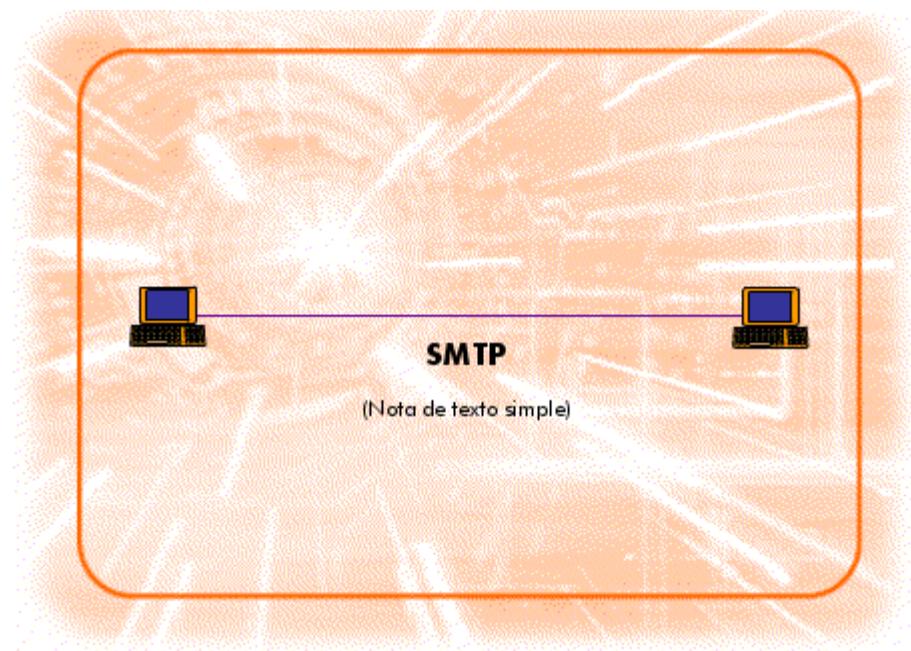
También se utilizan conexiones de datos para transmitir los listados de directorios. El servidor utiliza el puerto 20 para su extremo de la conexión de datos.



## 4- Protocolos de correo en Internet

El servicio de correo postal está en vías de extinción. Conocerás en profundidad el funcionamiento de su sucesor, el correo electrónico.

### SMTP



El correo es muy utilizado y son muchos los protocolos de Internet que han evolucionado para satisfacer las necesidades de los usuarios de correo electrónico.

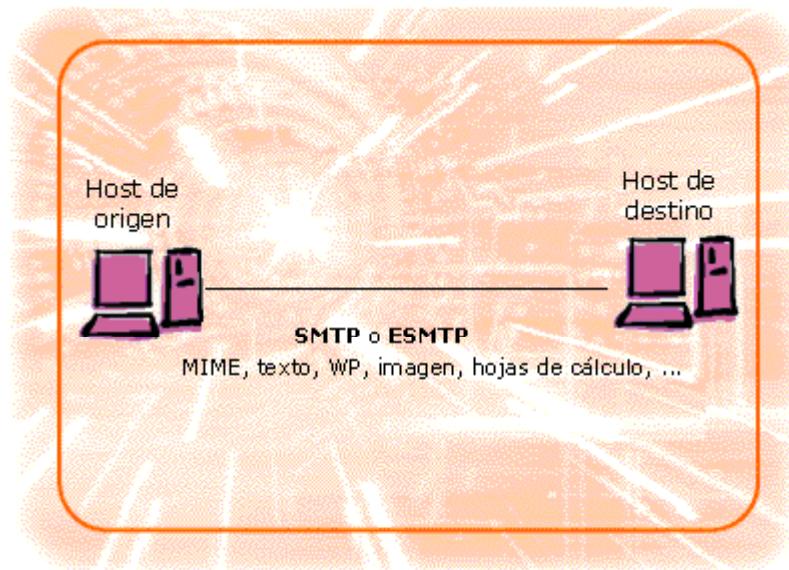
El protocolo básico de transferencia de correo (**SMTP**, "Simple Mail Transfer Protocol") es el estándar clásico de Internet para la transferencia de correo entre ordenadores.

SMTP se diseñó para transportar sencillas notas de texto y se implementó sobre una simple sesión del Terminal virtual de red (NVT, "Network Virtual Terminal") de TELNET.

Al llegar el correo, un Agente de usuario tiene que interpretar algunos elementos del mensaje, como el identificador del remitente, fecha de envío, asunto y la parte de información del mensaje.

El venerable **estándar para el formato de mensajes de texto en Internet de ARPA**, proporciona el formato para mensajes sencillos de correo de texto en Internet.

## ESMTP



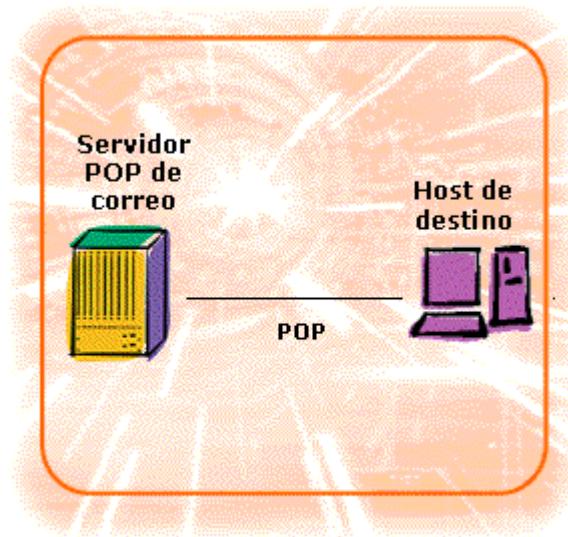
Un conjunto más reciente de normas define las extensiones de SMTP (**ESMTP**), que permiten transportar cualquier tipo de información.

Recientemente se han descrito los cuerpos de mensajes que constan de varias partes en las normas de extensiones de correo multipropósito de Internet (MIME, "Multipurpose Internet Mail Extensions").

Pueden entregarse muchos tipos de información, como documentos creados por procesadores de texto, imágenes, vídeo, sonidos codificados, hojas de cálculo, código ejecutable o cualquier otra cosa.

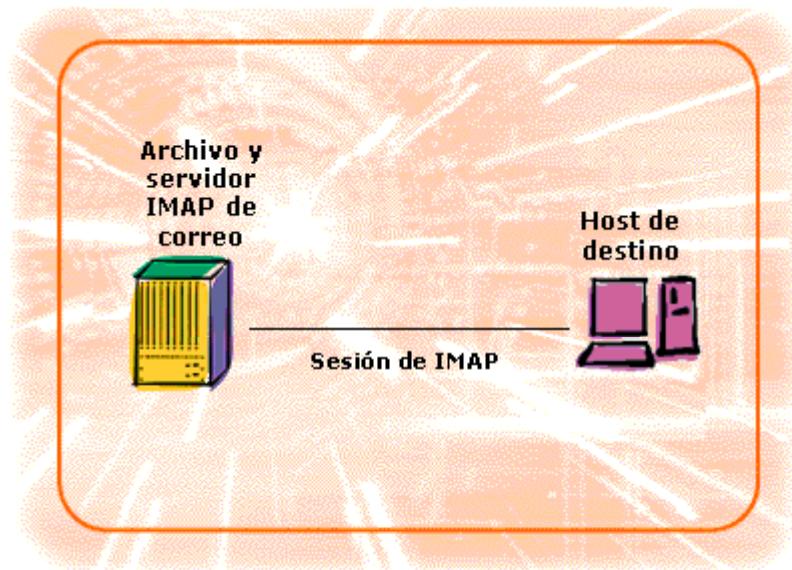
Los nuevos tipos MIME se definen según se necesitan y los registra la Autoridad de asignación de números en Internet.

## POP



Se ha diseñado otro conjunto de normas adaptadas a la forma de trabajo actual de mucha gente. El **Protocolo de oficina de correos (POP, "Post Office Protocol")** permite a un cliente obtener correo de un servidor de correo.

## IMAP



Como alternativa, el **protocolo de acceso a correo por Internet (IMAP)**, “*Internet Message Access Protocol*”), permite que un usuario lea, copie o borre los mensajes almacenados en un servidor, pero el servidor es el depositario autorizado de los mensajes.

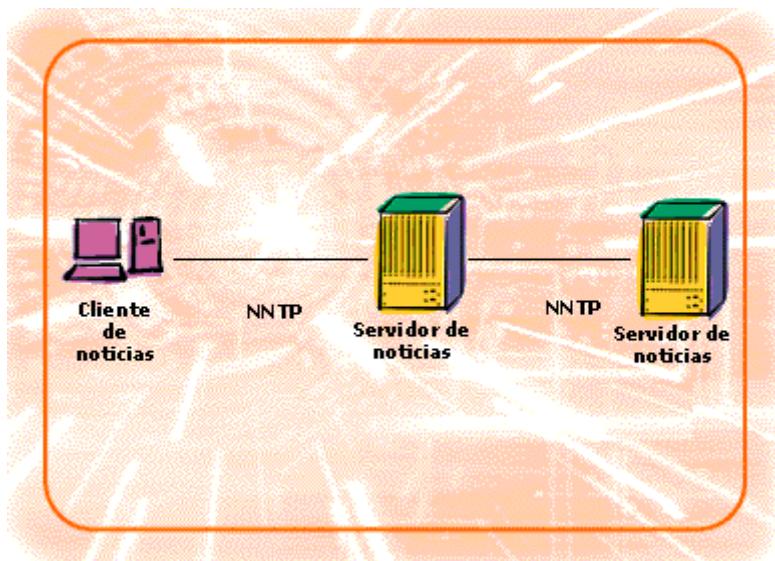
Resulta útil para los usuarios que quieren beneficiarse de los servicios administrativos, como la copia de seguridad diaria, evitar el uso de espacio en disco local o tener acceso a su correo cuando están de viaje.

El correo se entrega a un servidor por medio de SMTP o de ESMTP.

## 5- Noticias y gestión de red

Te enseñaremos como funciona el servicio que actúa como tablón de anuncios universal y algunos mecanismos de control en redes TCP/IP.

### NNTP



#### Noticias de red (News)

En las Noticias de red de Internet (News), todos los días se contribuye con información actualizada sobre ciencia, tecnología, ordenadores, economía, viajes, deportes, educación y mucho. Un grupo de noticias es como un tablón de anuncios (bulletin-board). Cada una de las noticias aportadas por los usuarios se distribuyen en forma de artículos que se envían (post) al grupo.

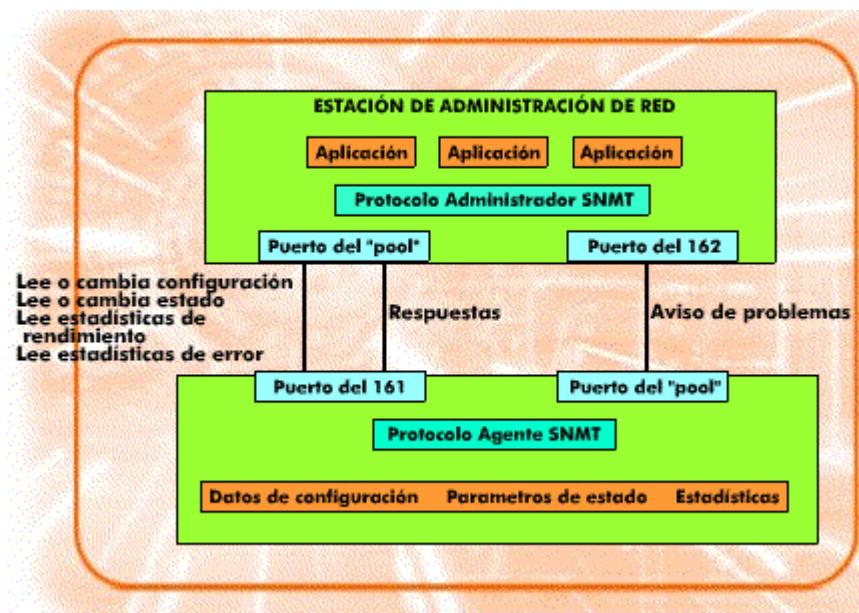
Cada grupo de noticias es mantenido por un administrador en un servidor de noticias primario. Si el grupo de noticias es privado, las noticias residen exclusivamente en dicho servidor y los usuarios pueden recogerlas de él.

Sin embargo, el envío a un grupo de noticias público, como Usenet, se propaga desde el servidor de noticias primario a cientos de otros servidores por todo el mundo.

Un cliente de noticias interactúa con un servidor de noticias de red usando el protocolo de transferencia de noticias de red (**NNTP**, "Network News Transfer Protocol"). NNTP permite:

- A un servidor, obtener noticias de otro servidor de noticias.
- A un agente de noticias del cliente, obtener noticias de un servidor de noticias.
- A un agente de noticias del cliente, enviar un nuevo artículo al servidor de noticias.

## SNMP



Los protocolos de gestión de red dividen el problema de la administración en dos partes, habiéndose especificado estándares separados para cada una de ellas:

- La primera parte se relaciona con la comunicación de la información. El Protocolo simple de gestión de red (**SNMP**, "Simple Network Management Protocol") especifica cómo se comunica el software de administrador con el software del sistema administrado.

El componente software del sistema administrado que posibilita esta comunicación se denomina **agente**. Si el módulo agente no está instalado, no será posible la administración de dicho elemento de red.

- La segunda parte se relaciona con los datos que se están administrando. Un protocolo especifica qué aspectos de los datos que reciba debe conservar un router, así como el nombre de cada uno de los aspectos y la sintaxis utilizada para expresar dichos nombres.

Un router administrado debe conservar el control y los estados de información a los que el administrador puede acceder. Por ejemplo, un router mantiene estadísticas del estado de sus interfaces de red, del tráfico que entra y sale, así como de los datagramas eliminados y de los mensajes de error generados.

Aún cuando se permite al administrador acceder a estas estadísticas, el SNMP no especifica exactamente a qué datos se puede acceder. De hecho, un estándar separado especifica los detalles.

Este estándar es conocido como "*Management Information Base*" (**MIB**). El MIB detalla qué elementos de los datos deben conservar los hosts o los routers, así como las operaciones permitidas en cada caso.

Por ejemplo, el MIB especifica qué software IP ha de llevar la cuenta de los objetos que llegan a cada interfaz de red, y especifica cuál es el único software de administración de red que puede leer estos valores.

## 6- World Wide Web

Seguro que eres un experto navegante, pero, ¿sabes como funciona la red? Estás muy cerca de entenderlo.

WWW

- Hipertexto.
- Hipermedia.
- Hipermedia y la WWW.

### Word Wide Web (WWW)

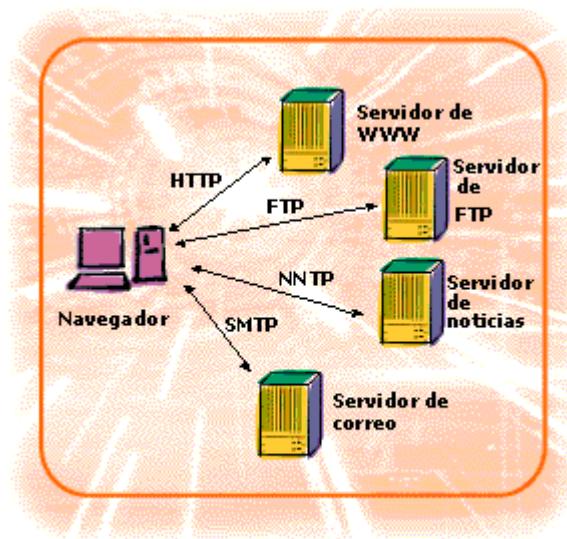
El **hipertexto** es una idea que ha estado presente durante bastantes años. Básicamente reside en que:

- Se asocia una frase subrayada con un puntero a un determinado documento.
- Un usuario puede enlazar con ese documento pulsando sobre la frase.

La idea del Hipertexto se ha extendido con la **Hipermedia** en donde se tiene una frase subrayada, o bien una imagen con elementos seleccionables, que apuntan a archivos de sonido, películas, imágenes o cualquier otro tipo de datos binarios. Este tipo de presentaciones suele distribuirse en CD-ROM.

El uso de hipermedia se extiende a la información en red usando la **World Wide Web (WWW)**. Una frase subrayada puede enlazar con un elemento local, o un elemento que realmente está almacenado en un ordenador remoto. Esta idea tan simple ha generado atractivas interfaces de usuario que permiten navegar fácilmente por Internet.

## Navegadores de la WWW



La utilización de la WWW tuvo un tremendo impulso cuando Marc Andresen, en 1982, creó la potente herramienta Mosaic.

**Mosaic** se concibió como un navegador de Internet, un programa que podría acceder a recursos de múltiples fuentes, incluyendo archivos de hipertexto, bases de datos de búsqueda, lugares de transferencia de archivos y lugares de noticias.

Como se muestra en el gráfico, un navegador puede utilizar los distintos protocolos requeridos para llegar a la información.

A partir de Mosaic nació un navegador comercial muy popular llamado **Netscape Navigator**.

El uso de la WWW y los navegadores ha crecido exponencialmente y los protocolos y tecnologías han avanzado muy rápidamente.

## URL

- URL de hipertexto.
- URL de transferencia de archivos.
- URL de TELNET.
- URL de correo.

### Localizador Uniforme de Recursos (URL)

De los esfuerzos de la World Wide Web nació un concepto unificador muy importante. Todos los recursos de información de la WWW se identifican por su Localizador Uniforme de Recurso, llamado a veces Localizador Universal de Recurso, o URL.

Formato general de un URL:

- Un URL empieza con el protocolo de acceso que se utiliza.
- En las aplicaciones que no sean de noticias o correo electrónico, le sigue el delimitador://
- A continuación va el nombre del servidor.
- Por último, se identifica el recurso al que se accede, o se obtiene el archivo por defecto.

### URL de hipertexto

Si se pone en un navegador Web el URL de un documento de hipertexto, el navegador buscará y obtendrá el documento usando un protocolo llamado Protocolo de transferencia de hipertexto (HTTP, "Hypertext Transfer Protocol").

### URL de transferencia de archivos

Se puede conectar con un equipo para transferir archivos con un URL del tipo:

ftp://host/

Para hacer FTP a un sitio donde hay que introducir un nombre de usuario y una contraseña, se usa:

ftp://nombre-de-usuario:contraseña@host/

### **URL de TELNET**

Se puede realizar una conexión TELNET, por ejemplo, con: telnet://host/

La forma más general es: telnet://nombre-de-usuario:contraseña@host

### **URL de correo**

Existe un URL para enviar correo electrónico: mailto:usuario@direccion-de-correo

El nombre o dirección del servidor de correo se introduce entre la información de configuración del navegador.

- Formatos.
- Cabeceras.
- Párrafos y saltos de párrafo.
- Enlaces.
- Imágenes.

### Lenguaje de marcas hipertextuales (HTML)

Los documentos de la WWW contienen enlaces de hipertexto que se escriben usando el Lenguaje de marcas hipertextuales o HTML. Los archivos escritos con HTML, normalmente, se guardan con la extensión ".htm"

La idea básica del HTML reside en que un autor pone marcas (tags) en el documento que se usa para identificar elementos como su título, las cabeceras de las secciones, los límites de los párrafos, las listas, las figuras y otros. Por ejemplo, la marca TITLE indica el título del documento.

Se puede escribir un documento de hipertexto con un editor de texto normal, utilizando directamente el lenguaje HTML para poner las marcas. Sin embargo, algunos procesadores de texto disponen de funciones que automatizan la creación de las marcas y permiten trabajar de forma "Lo que ve es lo que obtiene".

Las marcas son nombres de elementos y atributos encerrados entre <>. La mayor parte de las marcas son dobles, indicando dónde empieza y termina un elemento. Algunas de las marcas son las que ves en el gráfico.

#### Formatos

Algunas de las marcas se usan para delimitar el inicio y el fin de un documento HTML y para dividirlo en cabeza y cuerpo.

#### Cabeceras HTML

Marcan los capítulos, secciones y subsecciones de un documento. Existen seis niveles de cabeceras.

#### Párrafos y Saltos de párrafo

Un autor debe identificar los límites de los párrafos. Si no, cuando se presente, todo el texto se mostrará seguido.

## Enlaces

Para incluir enlaces en un documento se necesita:

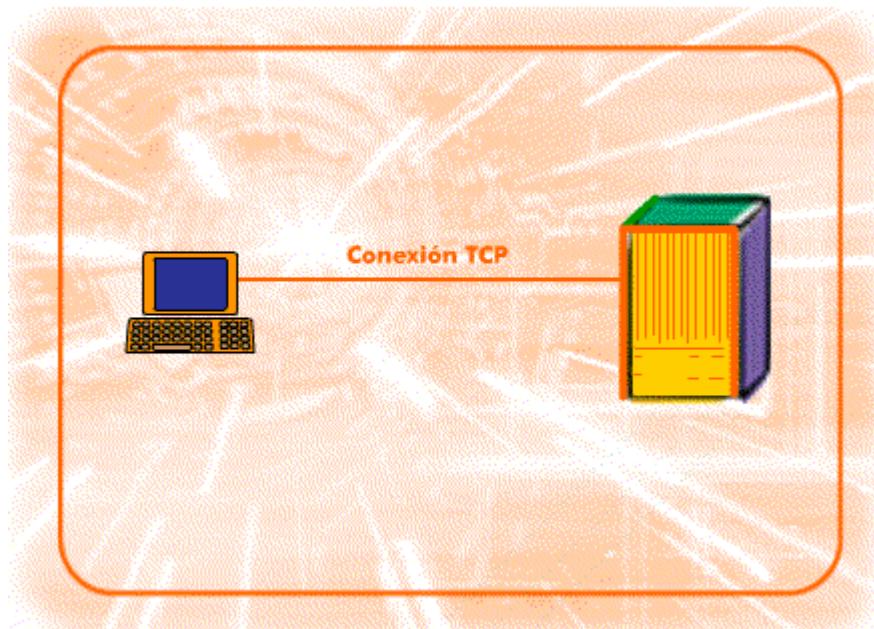
- Marcas de inicio y fin de enlace.
- Un parámetro con el URL que identifica al documento que se enlaza.
- El texto que habrá de ser seleccionado y que se mostrará subrayado en el interfaz de usuario.

Ejemplo: A HREF = "http://www.fyrsa.es/index.html" Pulse aquí para ver algo interesante./A

## Imágenes

Para insertar imágenes en un documento se usa la marca IMG. Esta marca tiene un parámetro SRC que indica el URL del archivo donde está la imagen.

## HTTP



El protocolo de transferencia de hipertexto (HTTP, "Hypertext Transfer Protocol") es el encargado de transferir ficheros HTML desde los servidores WWW (también llamados servidores de páginas Web).

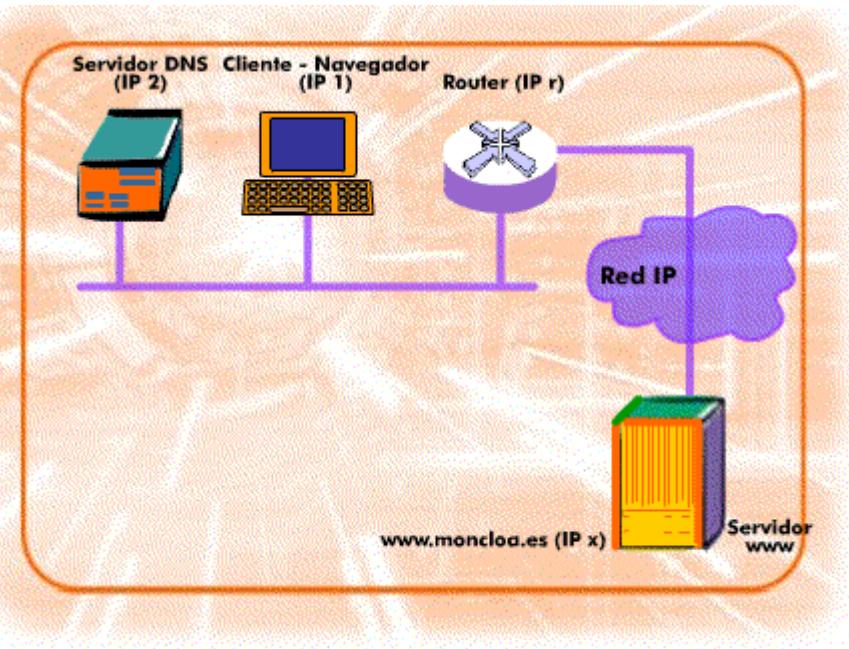
Un servidor WWW funciona de forma simple:

- El cliente se conecta al servidor (por TCP, por ejemplo).
- El cliente envía una petición, por ejemplo:  
- GET /home.html HTTP/1.0 ACCEPT: text/html
- El servidor responde, indicando el tipo de información que se enviará y, a continuación, transmitiendo el elemento.

Una razón por la que un servidor puede comunicarse con muchos tipos diferentes de clientes es que la información que se envía puede acomodarse a las capacidades de los clientes. Un cliente puede indicar sus capacidades enviando sentencias ACCEPT junto con sus peticiones. Un cliente puede indicar que sólo puede aceptar text/html, mientras que otro puede indicar que puede aceptar text, imágenes y sonido.

Normalmente, un servidor de WWW funciona sobre el puerto público de TCP 80.

## Escenario con HTTP



Un **servidor Web** mantiene en ejecución un proceso de escucha del puerto 80 (TCP) para conexiones entrantes de los clientes (navegadores). El usuario introduce un URL en el navegador, éste analiza el URL y se producen los siguientes eventos:

1. El navegador solicita al servidor DNS que sea traducida la URL en una dirección IP. Para ello, lo primero es obtener la dirección Ethernet del servidor DNS.
2. La máquina que contiene al servidor DNS responde con su propia dirección Ethernet.
3. El cliente DNS envía una petición al servidor DNS para que éste le traduzca la dirección www.moncloa.es
4. El servidor DNS contesta con la traducción: IPx
5. Nuestra máquina se da cuenta de que IPx no está en su red, por lo tanto, tiene que encapsular el datagrama hacia IPx en una trama Ethernet con la dirección Ethernet del gateway de salida de la red. Como desconoce esa dirección, ejecuta ARP para obtenerla.
6. El gateway reacciona a la petición ARP y contesta con su dirección Ethernet.

7. Nuestra máquina encapsula su petición HTTP en un segmento TCP, y éste en un datagrama IP, y éste en una trama Ethernet con dirección física de destino la del gateway.

**Cab. TCP:** Puerto origen (aleatorio), puerto destino (80), Número de seq. Y ACK, tamaño de ventana, etc.

**Cab. IP:** Tipo Protocolo (TCP), IP origen (IP2), IP destino (IpX), etc.

**Cab. Ethernet:** MAC origen (cliente), MAC destino (router), etc.

8. El gateway de salida encamina el datagrama hacia su destino, IPx.