

Redes de computadores: Apuntes oficiales II

Escrito por Profesores de redes de computadores de la Facultad de Informática de la UPM.

Editado por Pau Arlandis Martínez.

Tema 2: Internet y la arquitectura TCP/IP

Modelo de comunicaciones TCP/IP

Sistemas, Redes e Internet

Internet Socialmente

Internet es, tecnológicamente, una red de redes de comunicaciones (red de computadores) con tecnología TCP/IP, que usa un mismo formato de direccionamiento, y que basa su funcionamiento en la tecnología de conmutación o encaminamiento de paquetes. Pero aparte, el aspecto tecnológico hay que tener en cuenta el aspecto estratégico basado en la libertad, cooperación y gratuidad.

Nadie gobierna Internet. No existe ningún organismo propio de Internet que se encargue de controlar la red en su totalidad. **Internet es de todos y de nadie a la vez.** Cada red de comunicaciones conserva su independencia y está controlada y gobernada por su propia organización interna. Al no existir una autoridad central y ser una “red democrática y descentralizada”, todos los nodos pueden dialogar entre ellos de igual a igual. Numerosos individuos e instituciones han colaborado desinteresadamente en el desarrollo de nuevos procedimientos y aplicaciones, cuyo uso se ha ido extendiendo porque, a su vez, otros han participado con críticas, sugerencias, pruebas y mejoras. Por último, **en Internet no hay facturación** en función de la distancia a diferencia de otras redes comerciales. Tampoco se factura por el tiempo de acceso ni por el volumen de tráfico, otra cosa es lo que el usuario compre en Internet y lo que haga cada proveedor de servicios o ISP en función de su esquema de tarifas.

Estadísticas en Internet

Prácticamente desde 1988, Internet ha venido experimentando un crecimiento exponencial en casi todos sus parámetros. Algunos datos son los siguientes:

- En 1969 había una única red de comunicaciones (ARPANET: primera red de comunicaciones o embrión de Internet)
- En 1984 había más de 1000 redes de comunicaciones interconectadas en Internet.
- En 1992 Internet enlazaba más de 10.000 redes de 50 países.
- En 1994 se había logrado integrar 25.000 redes de 146 países.

- En 1995 se interconectaban más de 35.000 redes de comunicaciones y el número de máquinas servidoras era de unos 4.800.000.

Por la compleja infraestructura física y la conectividad privada, existente actualmente en Internet, es prácticamente imposible conocer el número exacto de redes de comunicaciones que la conforman. A su vez, se puede hablar de más de 1900 millones de usuarios (un 29% de la población) y 273 países conectados a Internet (cualquier país con un mínimo de nivel tecnológico). En el contexto español, el número de usuarios es de más de 29 millones (un 63% de la población). Para más detalles sobre estadísticas mundiales de Internet, consultar: <http://www.exitoexportador.com/stats.htm>.

Centros de control en Internet

Aunque Internet es una red descentralizada y sin dueño, es necesario un cierto control tanto para el acceso a la red como para el desarrollo de los protocolos y servicios en dicha red.

Centros de control de acceso a Internet

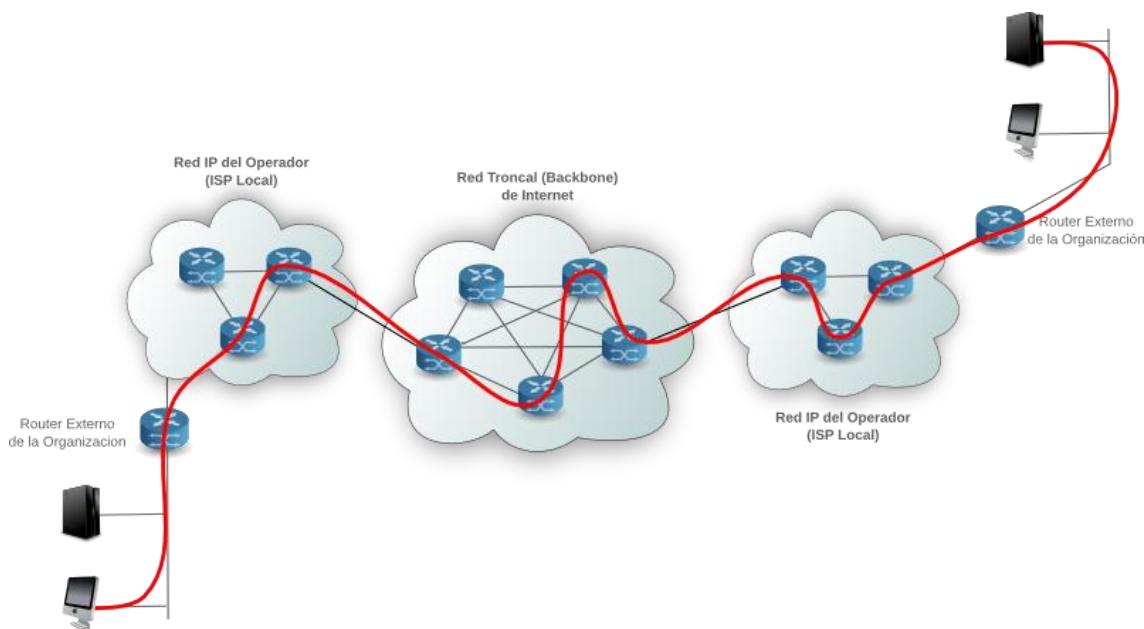
Aquí juegan un papel fundamental los operadores de telecomunicaciones o los ISP (*Internet Service Provider*) como Telefónica, Orange, ONO, Jazztel, etc.

Un **operador o Proveedor de Servicios de Internet o ISP** es una empresa u organización que ofrece acceso a Internet a sus clientes mediante una red IP y un *routerIP* conectado directamente a Internet.

Muchos ISP, también, ofrecen servicios extras relacionados con Internet a través de su red IP, como servidores de correo electrónico, servidores de páginas Web, servidores DNS, etc.

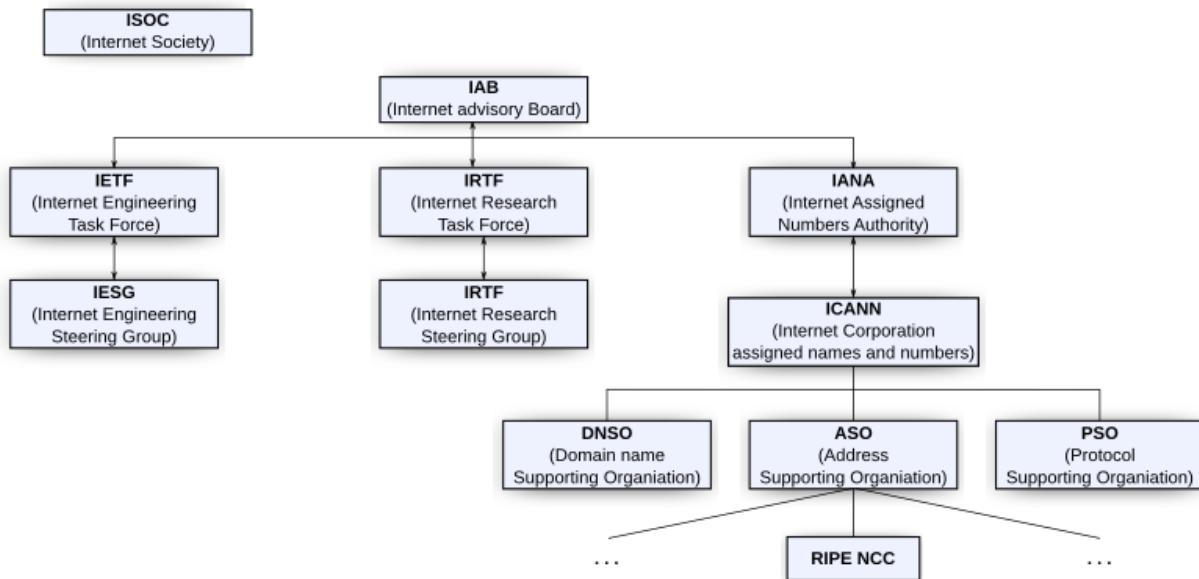
Para que un usuario acceda a Internet desde su casa, debe disponer, bien, de un ISP que también ofrece otros servicios como un servidor de páginas Web, de correo, noticias, etc., o bien, de un **proveedor de acceso a Internet o IAP (*Internet Access Provider*)** que sólo proporciona acceso a Internet.

En la siguiente figura se muestra un escenario típico de comunicaciones en Internet.



Centros para el control y evolución de Internet

Una de las características esenciales de Internet es su descentralización y que nadie gobierna esta inmensa red de computadoras, conservando cada red conectada su propia independencia. Sin embargo, para que semejante “anarquía” funcione es necesaria la existencia de una coordinación que sólo se preocupe de promover la red y de buscar respuestas a posibles problemas técnicos.



En este contexto, y según se describe en la figura anterior, la **Sociedad Internet o ISOC (Internet Society)**, promueve el conocimiento general de Internet y su tecnología especificando las reglas de carácter técnico. Consecuentemente, su principal objetivo es fomentar el crecimiento de Internet en todos sus aspectos (número de usuarios, nuevas aplicaciones, mejor infraestructura, etc.). Dentro de dicha sociedad está el “gran consejo de patriarcas” o **IAB (Internet Advisory Board)**, que organiza y gestiona la Sociedad Internet. El IAB está formado por un grupo pequeño de investigadores (*senior researchers*), la mayoría, diseñadores y desarrolladores iniciales de la arquitectura TCP/IP. El IAB se encarga de determinar las necesidades técnicas a corto, medio y largo plazo y de la toma de decisiones, guiando la evolución del conjunto de protocolos TCP/IP. También aprueba las recomendaciones y estándares de Internet. A su vez, del IAB dependen fundamentalmente dos organizaciones:

- **IETF (Internet Engineering Task Force):** Es el grupo de ingeniería de Internet que cuida los aspectos técnicos a corto y medio plazo. El grupo de dirección del IETF es el **IESG (Internet Engineering Steering Group)**.
- **IRTF (Internet Research Task Force):** Es el grupo de investigación de Internet que estudia los aspectos técnicos a largo plazo. El grupo de dirección del IRTF es el **IRTSG (Internet Research Steering Group)**.

Existe otro órgano de ISOC como es **IANA (Internet Assigned Number Authority)**, <http://www.iana.org>), responsable de la definición de políticas para la asignación de

diversos recursos de Internet (nombres simbólicos o de dominios, direcciones IP y valores o números utilizados en los protocolos TCP/IP). En este contexto, **ICANN**(<http://www.icann.org>), es una entidad delegada del IANA que lleva a cabo pragmáticamente todo el trabajo definido “en papel” por IANA. Asimismo, del ICANN dependen tres organismos:

- **DNSO:** Encargado del registro de nombres simbólicos o de dominios de primer nivel bajo la raíz (IANA/ICANN) en el planeta. En definitiva, controla que no existan dos o más países compartiendo el mismo nombre simbólico.
- **ASO:** Responsable de la asignación de direcciones IP por grandes bloques a cinco registros regionales, destacando el registro europeo:
- RIPE NCC: Europa y, también, Oriente medio y Asia central
- **PSO:** Encargado de la asignación de los diferentes números que identifican a los distintos protocolos y servicios en TCP/IP.

Las especificaciones en Internet: Documentos RFC

Existe una serie de **documentos RFC** (*Request for Comments*: Solicituds de comentarios) numerados en secuencia, de forma cronológica, que permite a los protocolos y servicios de la arquitectura TCP/IP ir evolucionando como estándares en Internet mediante un procedimiento documental aprobado por el IAB.

En 1969, el IAB inició un proceso más activo de centralización e información. El RFC IAB *Official Protocol Standards* mantiene una lista completa de todas las especificaciones o documentos RFC que son actualmente estándares en Internet. Estos RFC se clasifican por un *estatus* (status). El actual IAB *Official Protocol Standards*, es el documento RFC-5000 (STD0001). La revisión y publicación de los documentos RFC es responsabilidad directa del Editor de los RFC (*RFC Editor*: <http://www.rfc-editor.org/>) que es un miembro del IAB.

El IAB descansa en el IETF/IESG para la generación de un nuevo estándar que inicialmente parte como un documento borrador (ID o *Internet Draft*). Generalmente, el proceso para que una especificación se apruebe como estándar comienza por una recomendación de un grupo de trabajo del IETF al IESG (órgano de dirección del IETF). Sin embargo, cualquier usuario en Internet puede enviar una memoria propuesta como ID al Editor RFC. Las normas para escribir un RFC se definen en el documento RFC-2223 (*Instructions to RFC Authors*). A su vez, el documento RFC-2026 (“*The Internet Standards Process -- Revision 3*”) especifica todos los conceptos y terminología de los documentos RFC, así como el proceso de estandarización.

El *estatus* de un *RFC* tiene que ver con los niveles de madurez de su contenido o con las diferentes fases que se han de seguir en su definición. Los posibles *estatus* son:

- **Estándar (Standard):** Reconocido y normalizado. Sólo cuando un protocolo alcanza el estatus de estándar se le asigna un número de estándar (*STAndard number* o STD). Por ejemplo, el protocolo IP cuyo número de RFC es el 791, tiene un estatus de estándar y un STD0005. El STD nunca cambia aunque en un futuro el número actual de RFC sea actualizado por otro. En este último caso, el STD referencia al RFC original y a todos los documentos RFC estándares que actualizan al original.
- **Borrador estándar (Draft Standard):** En fase de estandarización

- **Propuesta estándar (Proposed Standard)**: Se considerará en un futuro, requiere implantaciones y pruebas.
- **Experimental (Experimental)**: En fase de experimentación, no debe ser implantado en el sistema salvo que se esté participando en el experimento y haya coordinado su uso con el desarrollador del mismo.
- **Informativo (Informational)**: Desarrollados por otros organismos de estandarización o vendedores fuera del alcance del IAB.
- **Histórico (Historic)**: Tiene poca probabilidad de transformarse en estándar de Internet, bien por estar obsoleto o por carecer de interés.

Estándar, Borrador estándar y Propuesta estándar, se consideran parte del camino (*The Internet Standards Track*) que debe recorrer un RFC hasta que se convierte en estándar. Por consiguiente, Experimental, Informativo e Histórico son distintos niveles de madurez fuera del ámbito del *Internet Standards Track*.

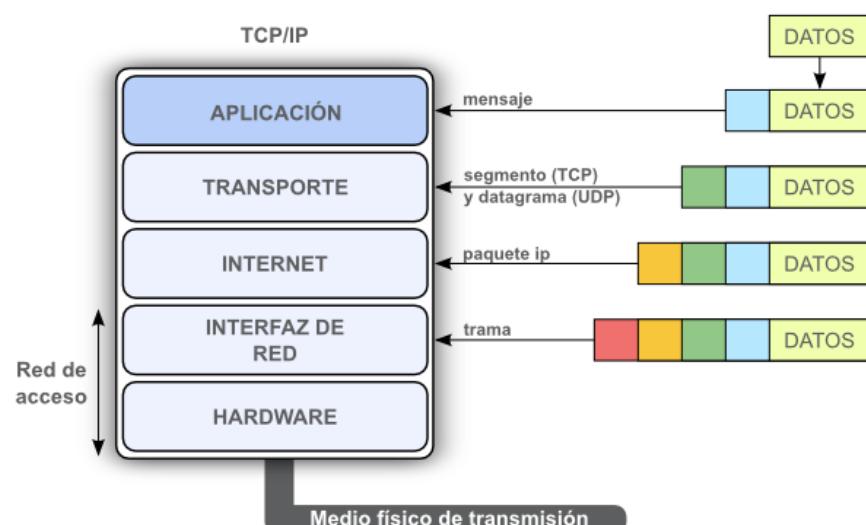
A su vez, existen unas subseries de las series de *RFC* conocidas como **Mejor Práctica Actual** o **BCP** (*Best Current Practice*) con el objetivo de estandarizar prácticas o experiencias o trabajos o resultados de la toda la comunidad de Internet para llegar a una especie de consenso. Así mismo, las series BCP se usan para documentar todas las prácticas que, a su vez, realice el mismo IETF.

También hay un conjunto separado de documentos *RFC* que no contienen especificaciones de protocolos y servicios de Internet, sino información útil y que se conoce como **documentos FYI** (*For Your Information*). Disponen, como los STD, de su propio número pero sólo está asociado a un RFC.

Hay un gran número de servidores *RFC* por Internet en donde se pueden consultar de forma gratuita los documentos *RFC* debidamente actualizados. Sin embargo. La dirección más recomendable es la del propio Editor de los *RFC* (<http://www.rfc-editor.org/rfcsearch.html>).

Arquitectura TCP/IP

Los protocolos TCP/IP se estratifican en una arquitectura estructurada en cinco niveles de comunicaciones. El nivel más alto o nivel de aplicación es el nivel con el que interactúan los usuarios. El nivel más bajo o nivel físico o de hardware viene definido por el hardware de acceso al medio físico de interconexión. Cada uno de sus niveles de software de comunicaciones salvo en el nivel físico, maneja diferentes unidades de datos de protocolo (**PDU: Protocol Data Unit**).



Los niveles que conforman la arquitectura de comunicaciones TCP/IP son:

- **Nivel de Aplicación:** Es el nivel más alto de la arquitectura TCP/IP. Aquí se ubican las entidades de software o programas, procesos, protocolos o servicios (transferencia de ficheros o FTP, correo electrónico o SMTP, navegación Web o HTTP, etc.) con los que interactúa directamente el usuario. Este nivel es el responsable de ejecutar los procesos de aplicación del usuario y de intercambiar mensajes de aplicación entre dos máquinas con el mismo protocolo de aplicación. Las PDU del nivel de aplicación se denominan **mensajes** y constan de una cabecera de información de control propia de la aplicación correspondiente y unos potenciales datos de usuario (si procede).
- **Nivel de Transporte:** Es el nivel responsable del transporte de los mensajes entre entidades del nivel de aplicación. En este nivel existen únicamente dos protocolos: TCP (*Transmission Control Protocol*) y UDP (*User Datagram Protocol*).
- El **protocolo TCP** realiza un transporte fiable extremo a extremo, independientemente, de la tecnología, topología, número y tipo de redes que hayan intervenido. Existe un control de errores físicos y lógicos y un control de flujo entre entidades TCP para impedir que una entidad transmita más rápidamente de lo que otra es capaz de almacenar y procesar. La PDU de TCP se denomina **segmento TCP**.

El **protocolo UDP** realiza un transporte no fiable pero más rápido de los datos. No se establece ninguna conexión extremo a extremo y cada datagrama UDP se trata como una unidad independiente y se envía aisladamente de las demás. Por consiguiente, no se mantiene ningún tipo de control de errores (solo hay, opcionalmente, una detección de errores físicos sin recuperación) ni de flujo. La PDU de UDP se denomina **datagrama UDP**.

Es importante resaltar que, a partir de este nivel todas las comunicaciones son extremo a extremo ya que no va a intervenir nunca una entidad TCP o UDP en el camino entre las dos entidades de transporte origen y destino.

- **Nivel de Internet o Nivel de Red o Nivel IP:** Es el nivel responsable del encaminamiento de los segmentos TCP y datagramas UDP encapsulados en datagramas IP, en función de la dirección IP del destinatario y siempre entre dos máquinas vecinas conectadas a la misma red de acceso en el trayecto entre el origen y el destino. Aquí se ejecuta un protocolo que se denomina IP (*Internet Protocol*). La PDU se denomina **datagrama IP** o datagrama y encapsula un único segmento TCP o datagrama UDP. Este nivel ofrece siempre un servicio no orientado a conexión. Consecuentemente, no se mantiene ningún tipo de control de errores ni de flujo. Se asume, por tanto, que si un datagrama se pierde y la aplicación funciona sobre TCP, será el protocolo TCP del nivel de transporte el encargado de su recuperación. Si la aplicación está montada sobre UDP, serán los mecanismos fiables de la aplicación (si están implementados) quienes lleven a cabo la citada recuperación.
- **Nivel del Interfaz de la Red de Acceso o Nivel de enlace:** Es el nivel responsable del intercambio de paquetes IP encapsulados en tramas de la red de acceso (por ejemplo, *Ethernet* o *WiFi*) y siempre entre dos máquinas vecinas conectadas a la misma red de acceso y con el mismo protocolo del nivel de enlace (*Ethernet* o *WiFi*).

Este es el nivel de software de comunicaciones más bajo de la arquitectura TCP/IP. La PDU de este nivel se denomina **trama** y cada trama encapsula un único datagrama IP.

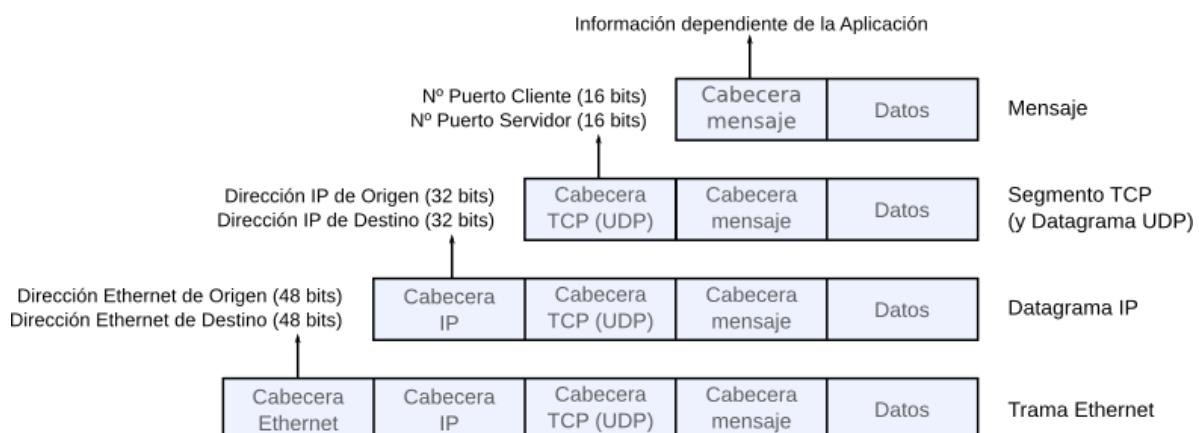
- **Nivel Físico o Nivel de Hardware:** Es el nivel responsable del acceso directo de las tramas a la red de comunicaciones (por ejemplo, *Ethernet* o *WiFi*). En este nivel, se definen las características físicas (tipo de conectores, número de pines, etc.) eléctricas (tensión o voltaje en los cables) y funcionales (señales intercambiadas con el correspondiente dispositivo transmisor y receptor) para acceder al medio físico de interconexión (red de acceso). Es importante resaltar que en este nivel no se incluye ningún tipo de software y, por tanto, no existe ningún protocolo de comunicaciones.

Para finalizar, conviene resaltar que aunque la arquitectura TCP/IP está formada por muchos protocolos y no sólo por TCP (nivel de transporte) e IP (nivel de red); éstos dos protocolos por su relevancia, dan nombre a toda la arquitectura de comunicaciones.

Comunicaciones entre niveles TCP/IP

El protocolo de aplicación añade una cabecera de aplicación a cada mensaje. Asimismo, el protocolo TCP segmenta el mensaje de aplicación y añade una cabecera TCP a cada segmento TCP numerando cada octeto de datos. A su vez, el protocolo UDP añade una cabecera UDP, sin numeración, a cada datagrama o mensaje UDP. Además, el protocolo IP añade una cabecera IP a cada segmento TCP o datagrama UDP y encamina en función de la dirección IP del destinatario. Para finalizar el protocolo del nivel de enlace o de la red de acceso (*Ethernet* o *WiFi*) añade una cabecera del nivel de enlace (*Ethernet* o *WiFi*) a cada datagrama IP y transmite hacia la siguiente máquina contigua o vecina, conectada a la misma red de acceso, en función de la dirección (*Ethernet* o *WiFi*) de dicha máquina.

En la siguiente figura se indica la información más significativa que aparece en la cabecera de información de control de los niveles de aplicación, transporte, Internet e interfaz de la red de acceso (por ejemplo, una *Ethernet*):

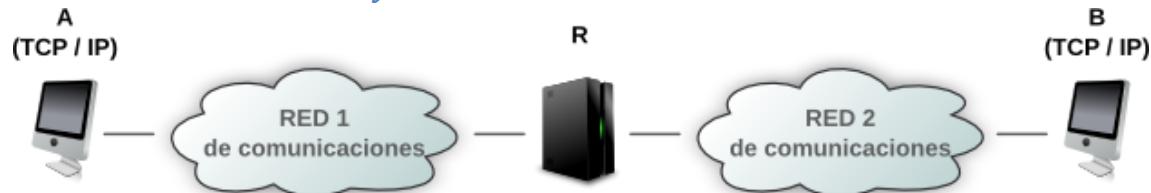


- La cabecera del mensaje de aplicación contiene la información de control propia de la correspondiente aplicación TCP/IP. Todas las cabeceras del nivel de aplicación son diferentes.
- La cabecera de un segmento TCP o datagrama UDP incluye como información más significativa, los números de puerto del proceso cliente y servidor.

- La cabecera de un datagrama IP contiene como información más relevante, las direcciones IP del sistema origen y destino.
- La cabecera de una trama incluye como información más significativa, las direcciones MAC de las tarjetas de red (*Ethernet*) del sistema origen y destino.

Asimismo, se resalta que en las cabeceras de información de control del nivel de enlace (trama) y del nivel de red (datagrama IP) existe un campo con un valor numérico que identifica al proceso “vecino” del nivel superior al que hay que entregar la información almacenada en el campo datos ya sea de la trama o del datagrama IP.

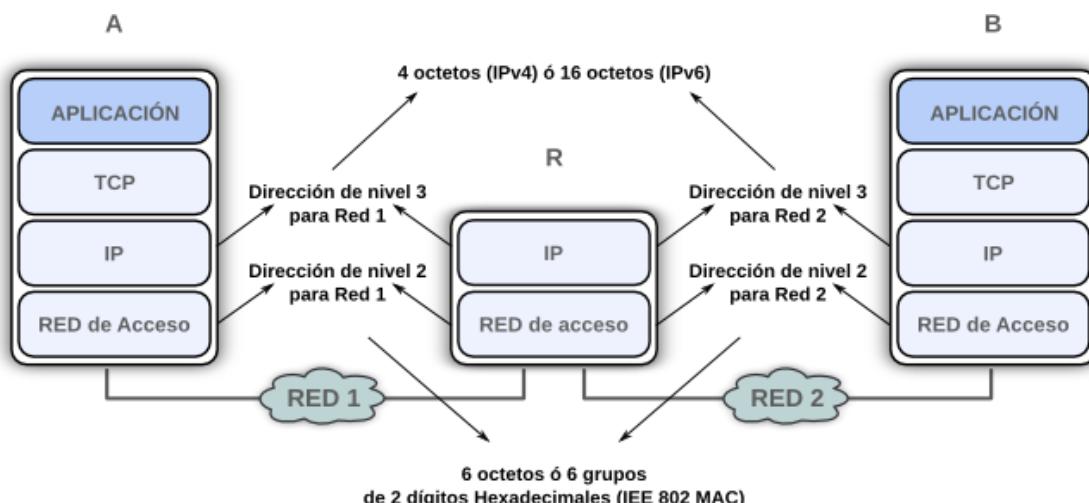
Direcciones del nivel de red y enlace



Todo sistema tiene tantas direcciones del nivel de red y del nivel de enlace como redes de comunicaciones (por ejemplo, *Ethernet* o *WiFi*) a las cuales esté conectado. Por ejemplo, el sistema “A” está conectado a una red de comunicaciones. Por tanto, dispone de una dirección del nivel de red propia de dicha red de comunicaciones y otra del nivel de enlace en función de la tarjeta de comunicaciones instalada en dicha máquina para dicha red. Ídem para el resto de sistemas de la figura.

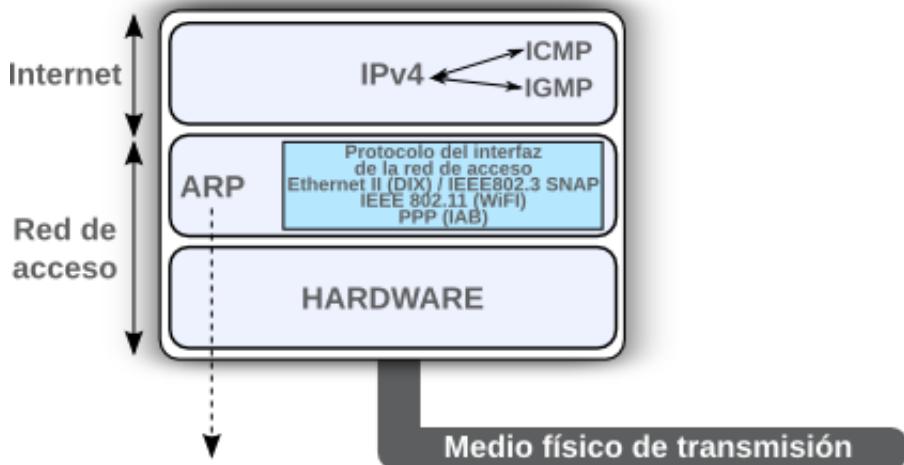
El formato de las direcciones del nivel de red y enlace es completamente diferente. En el nivel de red se maneja un **formato de direccionamiento IP** propio de la arquitectura TCP/IP y de Internet. Sin embargo, en el nivel de enlace se utiliza otro formato conocido como **formato MAC** propio de la tarjeta de comunicaciones instalada en el sistema.

Las máquinas de origen y destino en el nivel de red no tienen porqué coincidir con las máquinas de origen y destino en el nivel de trama.



Niveles inferiores TCP/IP

Los niveles inferiores de la arquitectura TCP/IP se corresponden con los niveles que no son extremo a extremo, es decir: Nivel de Internet y el nivel del interfaz de la red de acceso.



- **Nivel de Internet o nivel de red:** Aparte del protocolo principal del nivel que se corresponde con el protocolo IP, también se encuentran en este nivel los siguientes protocolos:
 - **ICMP (Internet Control Message Protocol):** Es el protocolo de envío de mensajes de control en Internet. Por ejemplo, cuando algo sospechoso ocurre con un datagrama, el protocolo ICMP se encarga de notificar el evento a la máquina origen del datagrama en cuestión. Aunque ICMP se ejecuta en el mismo nivel de Internet que el protocolo IP; sin embargo, conceptualmente, está situado en un subnivel superior al ocupado por IP. Por tanto, un mensaje ICMP se encapsula directamente en el campo de datos de un datagrama IP.
 - **IGMP (Internet Group Management Protocol):** Es el protocolo de gestión de grupos en Internet para descubrir miembros activos en máquinas locales que pertenezcan a grupos de multidifusión o *multicast* en Internet. Al igual que ICMP, el protocolo IGMP está situado conceptualmente en un subnivel superior al ocupado por IP. Por tanto, todo mensaje IGMP se encapsula directamente en un datagrama IP.
- **Nivel de interfaz de la red de acceso o nivel de enlace:** Es la capa de software de comunicaciones de más bajo nivel de la arquitectura TCP/IP. En este nivel se ejecuta el protocolo de comunicaciones de la interfaz de la red de acceso.

Los principales protocolos específicos de redes de área local son:

- **Protocolo Ethernet II (DIX):** Define el formato y orden de las PDU (tramas) intercambiadas entre dos entidades pares del nivel del interfaz de una red de acceso de cable del tipo *Ethernet*, así como las funciones que tienen que llevar a cabo dichas entidades pares para proporcionar el correspondiente servicio de dicho nivel. El estándar *Ethernet* define las especificaciones de una red de área local de cable o alámbrica, cubriendo el nivel de enlace de datos y nivel físico de dicha red. Se resalta que es el protocolo que se usa por omisión en una red *Ethernet*.
- **Protocolo IEEE 802.3 SNAP:** Define el estándar (“de iure”) del formato y orden de las PDU (tramas) intercambiadas entre dos entidades pares del nivel del interfaz de una red de acceso de cable, también, del tipo *Ethernet*, así como las funciones que tienen que llevar a cabo dichas entidades pares para proporcionar el correspondiente servicio de dicho nivel. Se resalta que el uso de este protocolo es opcional en una red *Ethernet*.

- **Protocolo IEEE 802.11 (WiFi):** Define el formato y orden de las PDU (tramas) intercambiadas entre dos entidades pares del nivel del interfaz de una red de acceso inalámbrica WiFi, así como las funciones que tienen que llevar a cabo dichas entidades pares para proporcionar el correspondiente servicio de dicho nivel. Los estándares IEEE 802.11 definen las especificaciones de una red de área local inalámbrica, cubriendo el nivel de enlace de datos y nivel físico de dicha red.

El protocolo específico de una línea serie o punto a punto es:

- **Protocolo PPP (Point to Point Protocol):** Define el formato y orden de las PDU (tramas) intercambiadas entre dos entidades pares del nivel del interfaz de una red de acceso basado en una línea serie o punto a punto entre dos sistemas (dos computadoras, una computadora y un *router*, incluso, entre dos *routers* contiguos). Básicamente, se usa con módems en líneas telefónicas o en una emulación de una línea serie en una red ATM o *Ethernet* en accesos ADSL.

Otros protocolos relacionados:

- **ARP (Address Resolution Protocol):** Sólo se usa si el sistema TCP/IP, en donde se ejecuta el protocolo ARP, está conectado a una red de difusión como es el caso de una red *Ethernet* o de una red inalámbrica WiFi. Se utiliza para obtener automáticamente la dirección de la tarjeta de red *Ethernet* o WiFi de otro sistema vecino en la misma red *Ethernet* o WiFi. Aunque ARP se ejecuta en el mismo nivel que los protocolos *Ethernet II DIX*, IEEE 802.3 SNAP y IEEE 802.11; sin embargo, conceptualmente, está situado en un subnivel superior a ellos en el sentido de que un paquete ARP se encapsula directamente en una trama de información.

Niveles superiores TCP/IP

A partir del nivel de transporte en la arquitectura TCP/IP todas las comunicaciones son extremo a extremo y todos los niveles (transporte y aplicación) son superiores.

Interfaz de sockets

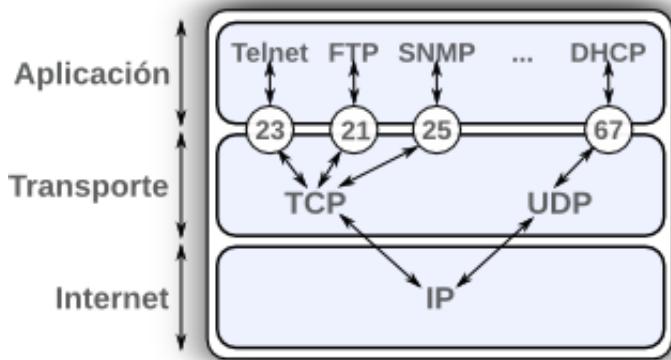
En la línea divisoria entre los niveles de transporte y aplicación se encuentra la interfaz de sockets y que permite la integración y montaje de aplicaciones en dicha arquitectura de una manera cómoda ya sea sobre el protocolo TCP, UDP y directamente sobre IP. En dicha interfaz se manejan los números de puerto y sockets del nivel de transporte.

Asimismo, conviene reseñar con antelación que independientemente de que el protocolo de transporte sea TCP o UDP, la mayoría de las aplicaciones en Internet funcionan según el típico modelo cliente y servidor:

- Servidor: Es un proceso que ofrece un servicio en la red.
- Cliente: Es un proceso que envía a un servidor una solicitud específica de servicio.

Cada proceso servidor (al igual que cada proceso cliente) del nivel de aplicación viene definido por un número de puerto que lo identifica.

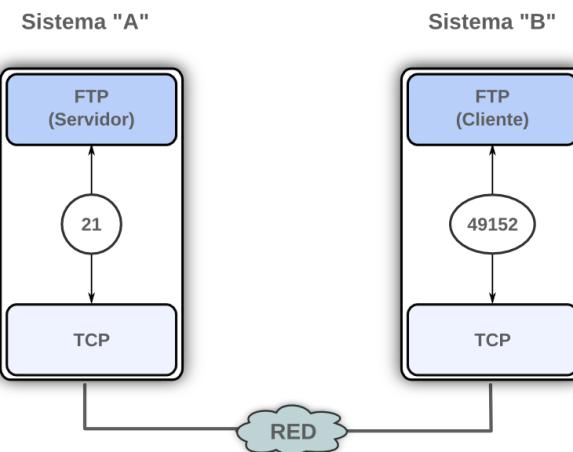
Un número de **puerto** es un entero positivo que manejan, en el nivel de transporte, las entidades TCP y UDP y que identifica tanto al proceso servidor como al proceso cliente donde se espera la respuesta.



Tanto TCP como UDP tienen definidos un grupo determinado de números de puerto, algunos de los cuales están ya reservados para las aplicaciones estándares en Internet. Tal es el caso, del 23 (TELNET), 21 (FTP), 25 (SMTP) para TCP o del 69 (TFTP) para UDP. Los puertos TCP son independientes de los puertos UDP, ya que la cabecera de IP especifica el tipo de protocolo.

- Del 0 al 1023: reservados para procesos servidores estándares de Internet.
- Del 1024 al 49151: para procesos servidores de aplicaciones desarrollados por un particular o por una empresa para su ámbito privado.
- Del 49152 al 65535: se recomienda que se asignen dinámicamente por el sistema operativo a los procesos clientes.

En la siguiente figura se muestra una comunicación entre el proceso cliente FTP de una máquina, identificado con el número de puerto 49152 (primer número libre fuera del rango de números reservados para procesos servidores estándares y no estándares) y el proceso servidor FTP de otra máquina definido por el número de puerto 21 reservado exclusivamente para dicho proceso.



Un **socket** o punto de acceso permite establecer una comunicación entre un proceso cliente y otro servidor.

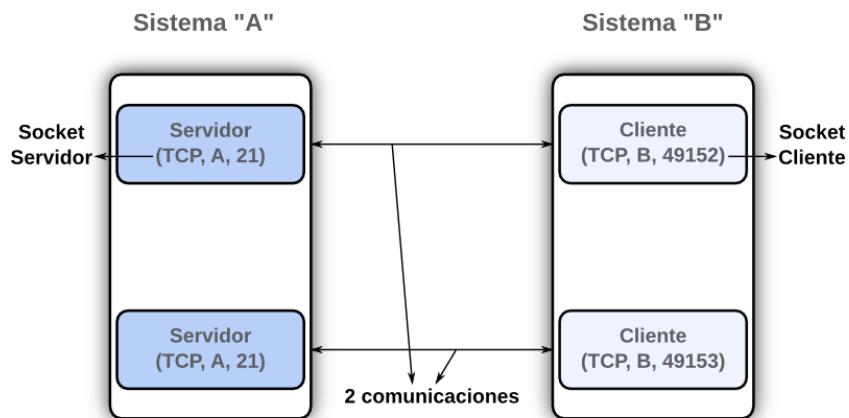
Este punto de comunicación o socket queda definido por las siguientes tres informaciones:

- Dirección IP del sistema.
- Protocolo del nivel de transporte: TCP o UDP.
- Número de puerto asociado al proceso correspondiente (cliente o servidor).

Con las anteriores tres informaciones, queda totalmente identificado cualquier proceso cliente o servidor de cualquier aplicación, independientemente del sistema en donde se esté ejecutando. El proceso cliente y el proceso servidor se pueden ejecutar en el mismo sistema o en sistemas diferentes (esto último es lo más habitual).

Una **conexión** queda plenamente definida por una pareja de sockets (socket cliente y socket servidor) que se comunican.

La siguiente figura muestra dos conexiones vía socket entre dos procesos clientes en el sistema "B" y un mismo proceso servidor de transferencia de ficheros en el sistema "A" cuyo número de puerto 21 está reservado exclusivamente para dicho proceso.



Nivel de control del enlace de datos

Introducción y generalidades

El **nivel de enlace** o nivel del interfaz de la red de acceso es un nivel inferior de la arquitectura TCP/IP y, por tanto, un nivel de comunicaciones entre máquinas vecinas, es decir, conectadas a la misma red de acceso (*Ethernet* o *WiFi* o una línea serie o punto a punto) para el intercambio de datagramas IP encapsulados en tramas de dicha red de acceso. Se resalta que igual que en cualquier otro nivel de comunicaciones (excepto el nivel físico que no existe un protocolo de comunicaciones), las dos máquinas deben usar un mismo protocolo del nivel de enlace.

Por el medio físico de interconexión o red de acceso irán las correspondientes tramas *Ethernet* en el caso de una red de acceso *Ethernet*; tramas *WiFi* si la red de acceso es *WiFi* o, finalmente, tramas *PPP* si la red de acceso es una línea serie o punto a punto. Cada trama del nivel de enlace encapsula un único paquete IP.

Funciones principales de un nivel de enlace fiable

Un protocolo de nivel de enlace fiable efectúa dos funciones:

- **Control de errores:** Implica detección de errores y su recuperación. Por errores se entienden tanto los errores lógicos (tramas de datos perdidas, desordenadas y duplicadas) como físicos (bits cambiados en las tramas). Los mecanismos de detección utilizan códigos polinómicos y sumas de comprobación. Los mecanismos de recuperación utilizan temporizadores y retransmisión (cada trama de datos tiene

asociado un temporizador o plazo de espera para la confirmación de dicha trama de datos y al vencimiento sin confirmación se produce una retransmisión) y también, tramas específicas de rechazo.

- **Control de flujo:** Ejercido por el proceso receptor sobre el proceso emisor para evitar que éste desborde el *buffer* de memoria del receptor mediante mecanismos de parada y espera (el emisor no puede transmitir una nueva trama de datos sin confirmación de la anterior) y ventana deslizante (el emisor transmite hasta un número autorizado de tramas pendientes de confirmación).

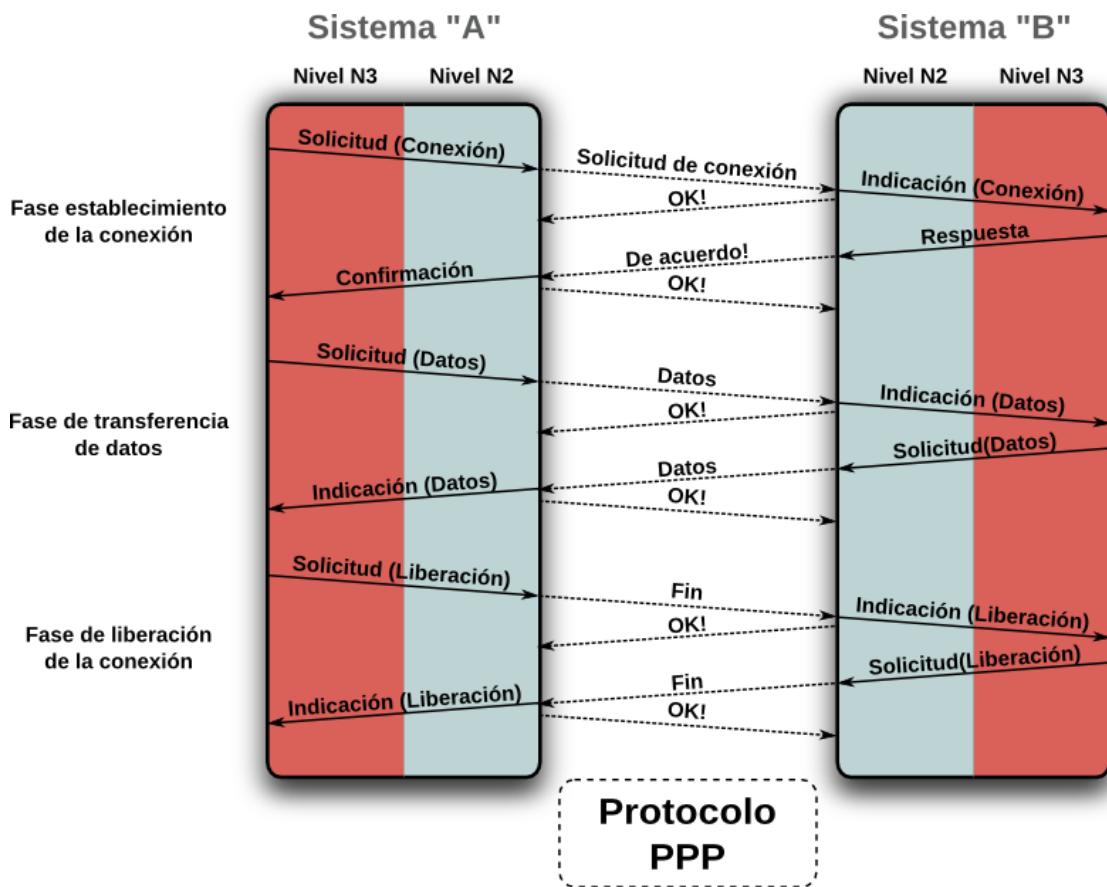
Servicios de un protocolo del nivel de enlace

Existen dos tipos de servicios ofrecidos por el correspondiente protocolo de nivel de enlace al nivel de red:

- **Servicio orientado a conexión:** formado por tres fases: Establecimiento de la conexión (fase confirmada por el nivel de red), transferencia de datos y liberación de la conexión.
 - **Con confirmaciones** o con fiabilidad en el nivel de enlace (Ej: protocolo PPP con negociación previa de fiabilidad): Incluye, en cada una de las tres fases, confirmaciones de recepción correcta de tramas por el propio protocolo del nivel de enlace. Con control de errores y control de flujo.
 - **Sin confirmaciones** o sin fiabilidad en el nivel de enlace (Ej: protocolo PPP configuración por omisión): Sin incluir confirmaciones de recepción correcta de tramas de datos por el propio protocolo del nivel de enlace. Sin control de errores ni control de flujo.
- **Servicio no orientado a conexión:** formado por una fase: Transferencia de datos.
 - **Sin confirmaciones** o sin fiabilidad en el nivel de enlace (Ej: protocolo Ethernet II o SNAP): No incluye confirmaciones de recepción correcta de tramas de datos por el propio protocolo del nivel de enlace. Sin control de errores ni control de flujo.
 - **Con confirmaciones** o con fiabilidad en el nivel de enlace (Ej: protocolo IEEE 802.11 – WiFi): Incluye en fase de transferencia de datos confirmaciones de recepción correcta de tramas de datos por el propio protocolo del nivel de enlace. Con control de errores y control de flujo.

Servicio orientado a conexión

En la siguiente figura se muestra el diagrama completo de envío de tramas correspondiente a un servicio orientado a conexión con confirmaciones. Se resalta que éste es el servicio ofrecido por el protocolo PPP.

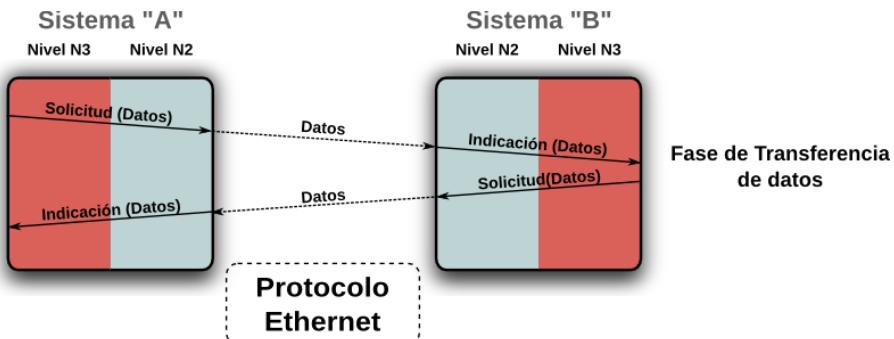


Si el servicio ofrecido por el nivel de enlace es orientado a conexión con confirmaciones, se dice que el servicio es fiable. Dicho servicio dispone siempre de tres fases:

- Establecimiento de la conexión:** El establecimiento de la conexión del nivel de enlace lo llevan a cabo las entidades pares del nivel de enlace por solicitud y confirmación expresa de las entidades del nivel de red. Es una especie de aviso; primero, para que la entidad receptora del nivel de red dé su consentimiento para recibir datos del nivel de red y, segundo, para que ambas entidades pares del nivel de enlace lleven a cabo, en la siguiente fase de transferencia de datos, todas las funciones que proporcionan fiabilidad. La fase de establecimiento de la conexión es siempre un servicio del nivel de enlace confirmado por el nivel de red mediante 4 llamadas y con fiabilidad en el nivel de enlace.
- Transferencia de datos:** Se corresponde con la transferencia de las tramas entre las entidades pares del nivel de enlace. La fase de transferencia de datos es siempre un servicio del nivel de enlace no confirmado por el nivel de red mediante 2 llamadas y con fiabilidad en el nivel de enlace.
- Liberación de la conexión:** La liberación de la conexión del nivel de enlace lo llevan a cabo las entidades pares del nivel de enlace por solicitud expresa de las entidades del nivel de red cuando éstas no tienen más datos que transmitir. Una vez las entidades del nivel de red han transferido todos sus datagramas IP, se procede a la liberación de la conexión del nivel de enlace previamente establecida. La fase de liberación de la conexión es siempre un servicio del nivel de enlace no confirmado por el nivel de red mediante 2 llamadas y con fiabilidad en el nivel de enlace.

Servicio no orientado a conexión

En la siguiente figura se muestra el diagrama completo de envío de tramas correspondiente a un **servicio no orientado a conexión sin confirmaciones**. Se resalta que éste es el servicio ofrecido por el protocolo *Ethernet*

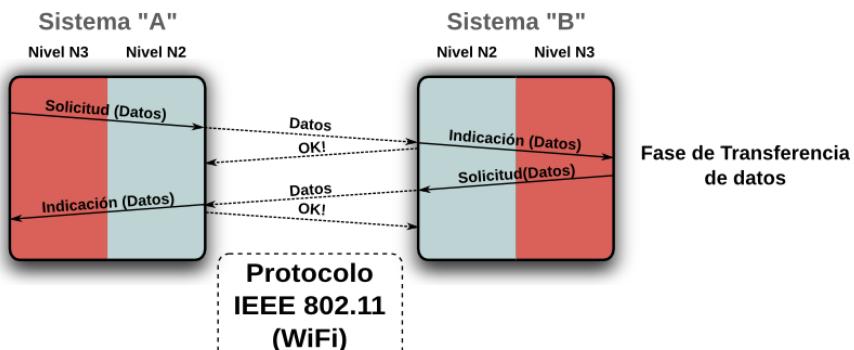


Si el servicio ofrecido por el nivel de enlace es no orientado a conexión sin confirmaciones (ni por el nivel de red ni enlace), se dice que el servicio es no fiable y dispone siempre de una fase:

- **Transferencia de datos:** Se corresponde con la transferencia de los datagramas IP entre las entidades pares del nivel de enlace. El envío de las tramas se realiza sin fiabilidad, no se mantiene ningún tipo de control de errores ni de flujo. La fase de transferencia de datos es un servicio no confirmado por el nivel de red mediante 2 llamadas y sin fiabilidad en el nivel de enlace.

El servicio no orientado a conexión sin confirmaciones es útil, en primer lugar, cuando los niveles superiores (nivel de transporte vía TCP o nivel de aplicación con fiabilidad vía UDP) ofrecen los mecanismos de control de errores y flujo necesarios. En segundo lugar, cuando interese transmitir por la red lo más rápidamente posible como es el caso de las transmisiones en tiempo real de imágenes y audio. En estos escenarios, la pérdida ocasional de datos puede no ser importante siempre que esta pérdida no afecte demasiado al resultado final y los datos que lleguen lo hagan de forma rápida.

En la siguiente figura, se muestra el diagrama completo de envío de tramas correspondiente a un **servicio no orientado a conexión con confirmaciones** y, por tanto, fiable (aunque no haya establecimiento ni liberación de la conexión). Se resalta que éste es el servicio ofrecido por el protocolo WiFi.

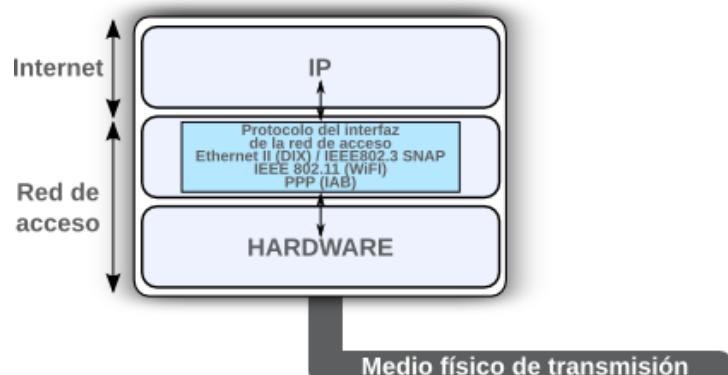


Si el servicio ofrecido por el nivel de enlace es no orientado a conexión con confirmaciones (dispone siempre de una fase):

- **Transferencia de datos:** Se corresponde con la transferencia de las tramas del nivel de enlace entre las entidades pares del nivel de enlace. El envío de las tramas se realiza con fiabilidad, se mantiene un control de errores y de flujo. Este servicio es muy útil cuando el medio físico de interconexión es el aire por las condiciones del medio y su susceptibilidad a interferencias electromagnéticas como es el caso de las redes de acceso WiFi. También, tiene utilidad en la gestión de alarmas o señales de control de emergencia de una organización. En este último escenario es muy útil una confirmación de modo que el emisor pueda estar seguro de que el receptor ha recibido la señal o el aviso pertinente. Además, teniendo en cuenta la urgencia de la señal, no se debe perder tiempo en establecer la conexión como paso previo a la transferencia de los datos.

Protocolos actuales del nivel de enlace en Internet

En la siguiente figura se muestran los protocolos actuales TCP/IP más relevantes del nivel de enlace en Internet.

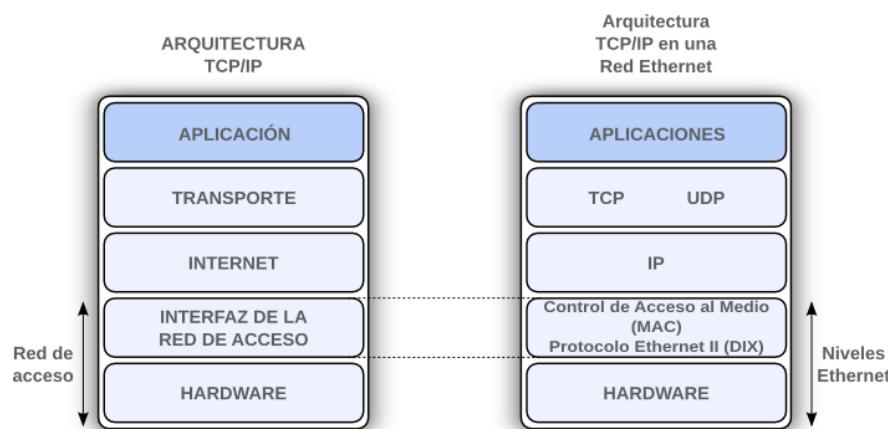


Protocolo Ethernet

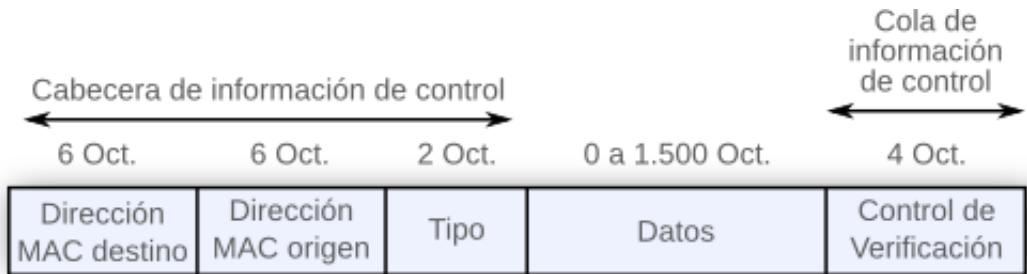
Las principales características de este protocolo son las siguientes:

- Una fase: Transferencia de datos.
- No hay control de errores ni control de flujo. Detecta errores físicos o de transmisión (bits cambiados) y elimina dichas tramas.

Actualmente, el **protocolo Ethernet II (DIX)** es el protocolo “estándar de facto” del interfaz de una red de acceso de cable o alámbrica *Ethernet*.



El formato de la trama *Ethernet II* (DIX) consta de una cabecera de información de control, de los datos que se quieren transmitir (carga útil) y de una cola de información de control, con los siguientes campos:

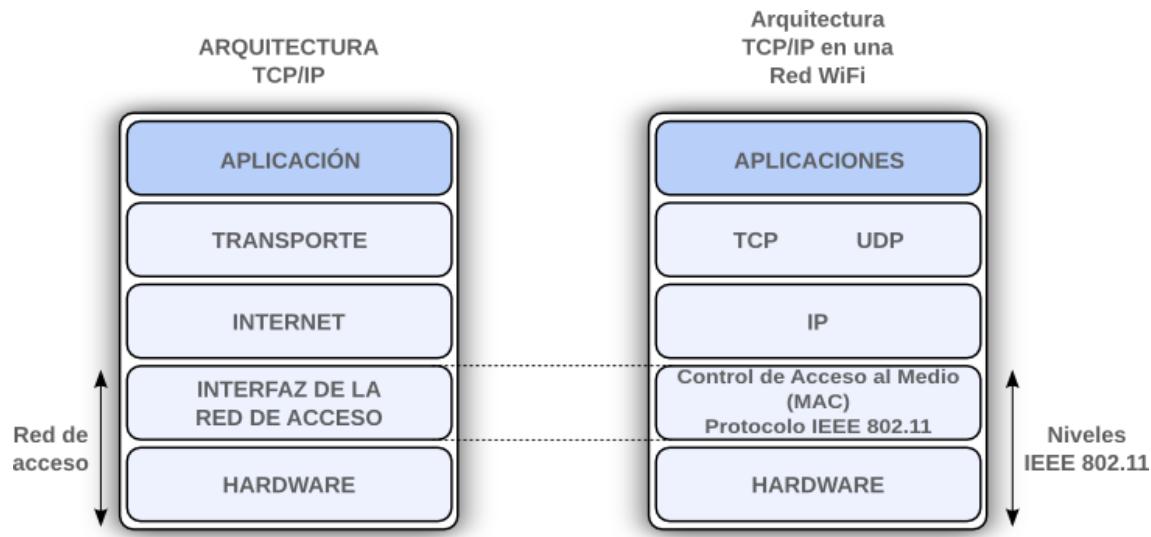


- **Dirección MAC destino** (6 octetos): Contiene los 48 bits de la dirección de la tarjeta de red *Ethernet* del sistema destinatario de la trama. Se expresa en hexadecimal, por ejemplo: 00:E0:4F:00:00:A0.
- **Dirección MAC origen** (6 octetos): Contiene los 48 bits de la dirección de la tarjeta de red *Ethernet* del sistema emisor de la trama. Se expresa en hexadecimal, por ejemplo: 00:15:00:3C:47:33.
- **Tipo** (2 octetos): Contiene el identificador del proceso del nivel superior al que hay que entregar el contenido del campo datos de la trama. Se utilizan números por encima de del 1536. Por ejemplo: el 2054 identifica al proceso ARP, el 2048 identifica al proceso IPv4, etc.
- **Datos** (entre 46 y 1500 octetos): Contiene la PDU encapsulada del nivel superior. Por ejemplo, un datagrama IP o un paquete ARP.
- **Control de verificación** (4 octetos): El emisor lleva a cabo un cálculo con todos los bits de los campos anteriores y el resultado lo almacena en este campo. Si algunos de los bits de datos se reciben erróneamente, el receptor detectará dichos errores al realizar el mismo cálculo que el emisor y no obtener el mismo resultado. Se resalta que sólo hay detección de errores y no corrección de los mismos.

Además, la trama *Ethernet II* (DIX) dispone de un campo **preámbulo** de 7 octetos de 1 y 0 alternos (10101010 10101010 10101010 10101010 10101010 10101010 10101010) que alertan al sistema receptor de la llegada de una trama y le permite sincronizar su entrada. El patrón 10101010 sólo proporciona una alerta y un pulso de sincronización. A continuación del preámbulo aparece un **delimitador de inicio de trama** (10101011) que indica el comienzo de la trama. El preámbulo y el delimitador se añaden realmente en el nivel físico y, por tanto, formalmente no forman parte de la trama. El final de la trama se detecta cuando no se recibe señal.

Protocolo WiFi

El **protocolo WiFi** (o Wi-Fi, Wi-fi, Wifi, wifi) es un conjunto de estándares para redes inalámbricas basados en las especificaciones IEEE 802.11.

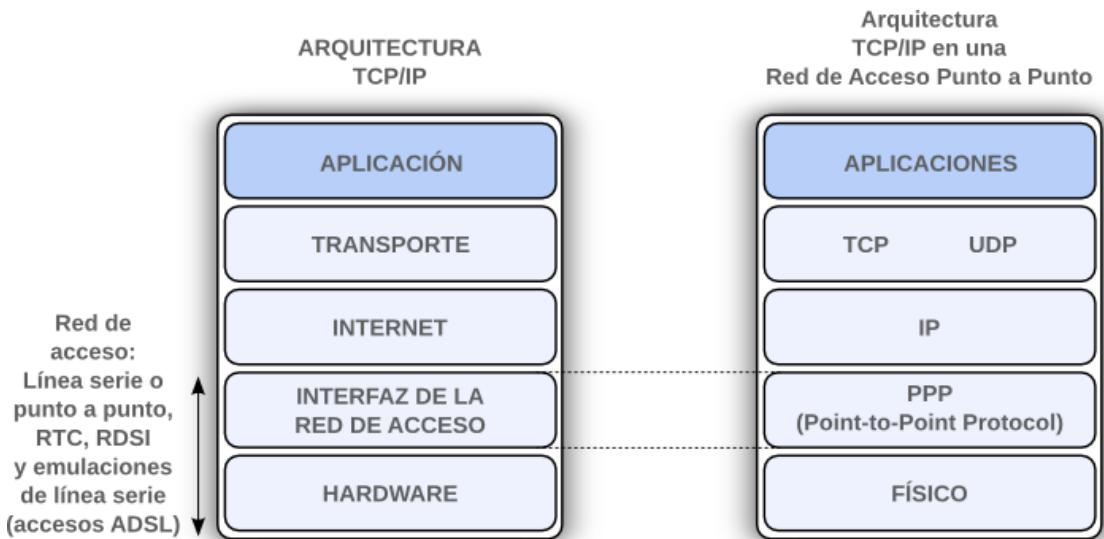


Las principales características de este protocolo son las siguientes:

- Una fase: Transferencia de datos.
- Hay control de errores y flujo. Detecta errores físicos o de transmisión (bits cambiados), eliminando y recuperando dichas tramas.
- Ofrece un servicio no orientado a conexión con confirmaciones y, por tanto, fiable debido a que el entorno inalámbrico es propenso a sufrir interferencias, ruido y otros efectos nocivos de propagación que pueden repercutir en la pérdida de un número significativo de tramas.
- Se basa fundamentalmente en un intercambio de dos tramas: Trama de datos y trama de confirmación (ACK o *ACKnowledgement*) asociada. Toda trama de datos tiene su temporizador asociado. Si el origen no recibe la confirmación a la trama enviada en un determinado periodo corto de tiempo, bien porque la trama de datos o la trama ACK resultó dañada, el origen retransmite la trama de datos.
- Sólo se envía una trama de datos (que tiene un temporizador asociado de espera de respuesta) y un ACK.

Protocolo PPP

El protocolo PPP (Point to Point Protocol) es el protocolo estándar del interfaz de una red de acceso cableada o alámbrica basada en una línea serie o punto a punto, es decir, entre únicamente dos sistemas. Dichos sistemas se encuentran ubicados en cada uno de los dos extremos del enlace o cable.



El protocolo PPP se ha convertido en un estándar en Internet (RFC-1661) para transportar datagramas IP encapsulados en tramas PPP a través de:

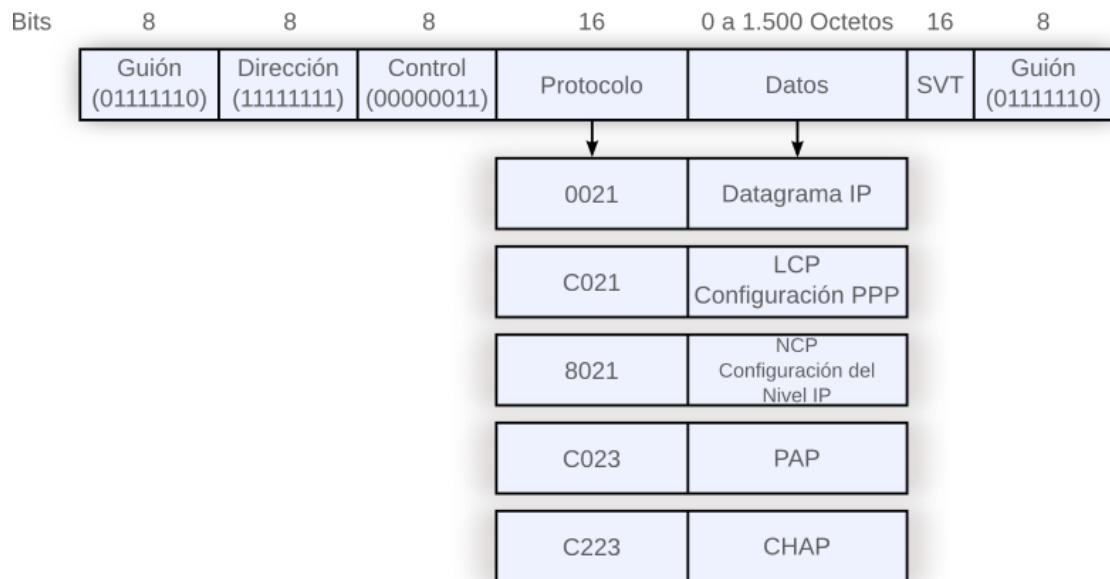
- Líneas serie o punto a punto, es decir, entre cables que conectan únicamente dos sistemas. Por ejemplo, líneas alquiladas o privadas entre dos *routers* contiguos en Internet. Se resalta que, actualmente, este tipo de enlace está prácticamente obsoleto ya que la mayoría de los sistemas TCP/IP suelen venir con una tarjeta *Ethernet* incorporada de 100 Mbps/1 Gbps; con lo cual, no hay más que conectarlos a través de un cable *Ethernet*.
- Líneas telefónicas analógicas con módem básico (RTC/RTB) y digitales (RDSI) entre usuarios y proveedores del servicio de acceso a Internet (también denominados ISP). En este caso, los usuarios se comunican con su ISP a través de un módem básico y tradicional (no ADSL), haciendo uso de una línea conmutada o línea serie o punto a punto telefónica que les conecta directamente con su central telefónica local. Se destaca que este tipo de enlace está cada vez más en desuso por la poca velocidad de transmisión que se alcanza con este clase de módems básicos.
- Emulaciones de líneas serie sobre redes *Ethernet* o PPPoE (PPPoE over Ethernet) y ATM o PPPoA (PPPoA over ATM). Actualmente, las emulaciones de líneas serie, especialmente, sobre la red ATM es uno de los usos más mayoritarios del protocolo PPP. Dicho protocolo está especialmente adaptado a los accesos ADSL para la asignación de direcciones IP públicas e información de configuración TCP/IP a los *routers* ADSL de los usuarios.

Las principales características de este protocolo son las siguientes:

- Ofrece un servicio orientado a conexión con confirmaciones o fiable (protocolo PPP con negociación previa de fiabilidad) o sin confirmaciones o no fiable (protocolo PPP con configuración por omisión). Este último servicio (no fiable) es el que proporciona, por omisión, en accesos ADSL.
- Soporta la asignación dinámica de direcciones numéricas (negociación de opciones de nivel de red mediante tramas especiales PPP).

- Opcionalmente, soporta distintos mecanismos de autenticación (protocolos CHAP, PAP, etc.) cuyos mensajes se encapsulan en tramas PPP. Esto resulta especialmente útil en el caso de conexiones RTC como es el caso de los ISP que han de facturar a los usuarios en función del tiempo de conexión.
- Opcionalmente, soporta mecanismos (mediante tramas especiales PPP) para probar el enlace y medir la calidad de la línea.

El formato de trama del protocolo PPP comienza y terminan con un campo **guión** de 8 bits (01111110) que indica el comienzo y final de la trama.



Luego siguen los siguientes campos:

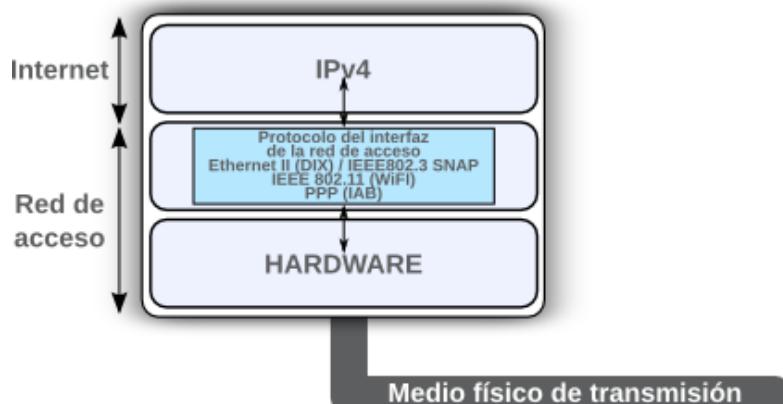
- **Dirección** (8 bits): No se utiliza y, por tanto, siempre contiene el valor binario 11111111 para indicar que todos los sistemas deben aceptar la trama. El empleo de este valor evita tener que asignar direcciones específicas de enlace a las dos estaciones implicadas.
- **Control** (8 bits): Indica si PPP ofrece un servicio orientado a conexión (fiable) o sin conexión (no fiable). Por omisión, el valor es 00000011 que indica que el servicio es no orientado a conexión. Salvo que se negocie una transmisión fiable, los campos de Dirección y Control contienen siempre la secuencia 111111100000011. Para no tener que transmitir estos dos octetos de información inútil en todas las tramas, generalmente, LCP negocia la supresión de dichos octetos al inicio de la sesión siempre y cuando el servicio ofrecido sea no orientado a conexión.
- **Protocolo** (16 bits o 32 bits): Indica el tipo de paquete contenido en el campo de datos.
- **Datos** (variable): Es la carga útil transportada por la trama PPP. Dicha carga es de longitud variable hasta un máximo negociado. Si la longitud no se negocia con LCP durante el establecimiento de la línea, se usa una longitud por omisión de 1500 octetos.

- **SVT** o Secuencia de Verificación de Trama o Suma de comprobación (16 bits o 32 bits): Suma aritmética binaria o en módulo 2 sin acarreos (suma XOR o OR-exclusivo) de todos los octetos de la trama. El emisor realiza la suma antes del envío e inserta el resultado en dicho campo. El receptor lleva a cabo el mismo procedimiento y comprueba si el resultado que ha obtenido coincide con el contenido de dicho campo insertado por el emisor. Si no es así, se elimina la trama y se retransmite ésta en caso de que PPP ofrezca un servicio orientado a conexión.

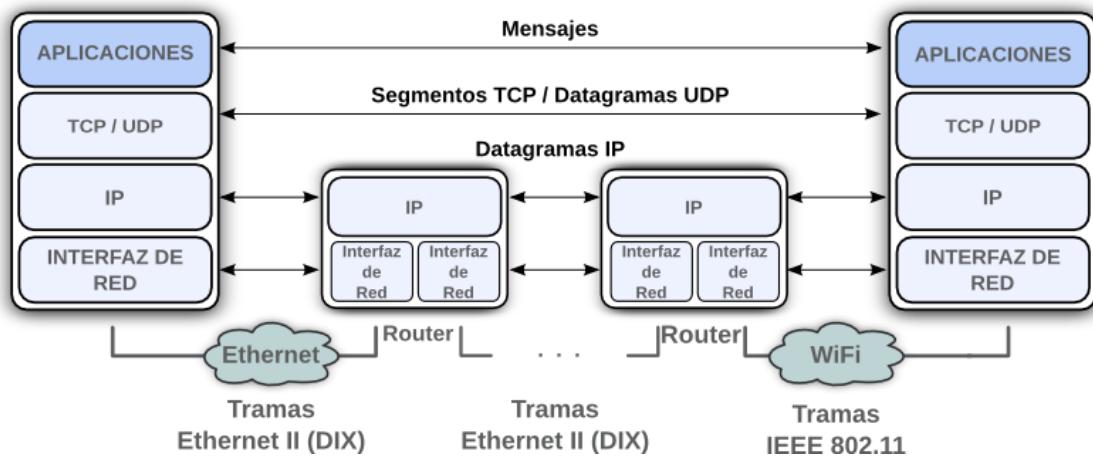
Nivel de Red: Internet Protocol (IP)

Introducción y generalidades

El **nivel de red** TCP/IP o nivel de Internet o, también llamado, nivel IP es el responsable de determinar la ruta o el camino que han de seguir los datagramas IP por Internet desde un sistema origen a otro destino. A los algoritmos que determinan dichas rutas se les denomina algoritmos de encaminamiento.



En dicho nivel de la arquitectura TCP/IP se ejecuta una entidad o proceso software IP que se rige bajo el protocolo IP (*Internet Protocol*). En un sistema final origen, la entidad IP va aceptando segmentos TCP o datagramas UDP del nivel de transporte y les va añadiendo a cada uno de ellos una cabecera IP. Cada segmento TCP o datagrama UDP se encapsula en un único datagrama IP. A la unidad de datos resultante se la denomina paquete o **datagrama IP**. Posteriormente, la entidad IP encamina cada datagrama IP a través de Internet usando un **algoritmo** y una **tabla de encaminamiento** para saber si un determinado paquete lo debe enviar directamente a su propia red (en el caso de que el sistema final destinatario sea vecino) o a un *router* contiguo en la misma red (en el caso de que el sistema final destinatario no sea vecino). En este último caso, la entidad IP del *router* contiguo procede de manera similar. Este nivel, ofrece siempre un servicio no orientado a conexión, no se realiza ningún tipo de control de errores ni de flujo durante el encaminamiento. De ahí, que se diga que el protocolo IP utiliza un servicio de “mejor entrega posible” o “mejor esfuerzo” o “hago lo que puedo” (*best effort*). Asimismo, este nivel no es extremo a extremo ya que siempre que exista al menos un *router* entre los sistemas finales de origen y destino, se imposibilita el envío directo de datagramas entre ambos extremos ya que dichos datagramas tienen que ser procesados previamente por la entidad intermedia IP alojada en el nivel de red del *router*.



Versiones del protocolo IP

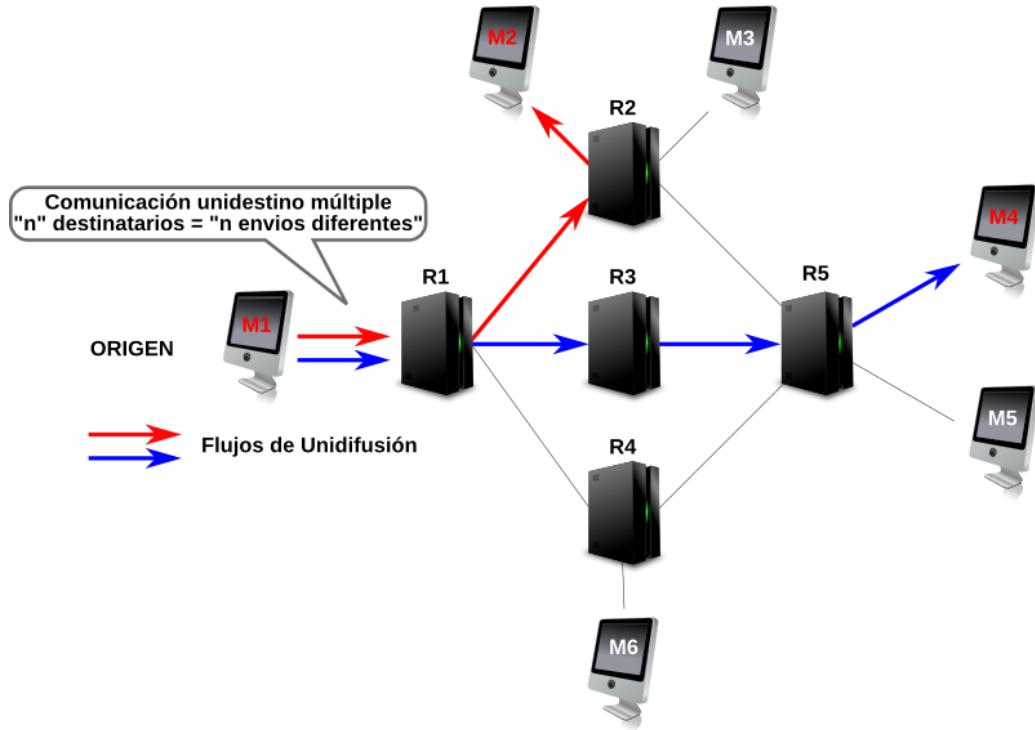
Actualmente, existen dos versiones del protocolo IP:

- **IPv4:** Versión 4 del protocolo IP de encaminamiento actual en Internet para todo tipo de usuarios
 - **IPv6:** Versión 6 del protocolo IP de encaminamiento futuro en Internet, también, para todo tipo de usuarios. Se destaca que IPv6 es un IPv4 mejorado con las siguientes diferencias básicas:
 - Direccionamiento: 16 octetos de la versión 4 de IP frente a los 4 octetos de la versión 4 de IP.
 - Flexibilidad y rapidez: En la versión 6 de IP se agiliza el proceso de encaminamiento. Por ejemplo, la cabecera de información de control es más simple ya que dispone de la mitad de campos y un nuevo formato flexible de cabeceras de extensión opcionales para utilizar los servicios adicionales cuando se necesiten (fragmentación y reensamblado, seguridad, etc.).
 - Seguridad: En la versión 6 de IP se han implementado diversos mecanismos de seguridad en una serie de cabeceras de extensión opcionales con el objetivo, por ejemplo, de asegurar la autenticación y confidencialidad de los datos. Se resalta que en IPv4 no se ha implementado ningún mecanismo de seguridad en su cabecera de información de control.

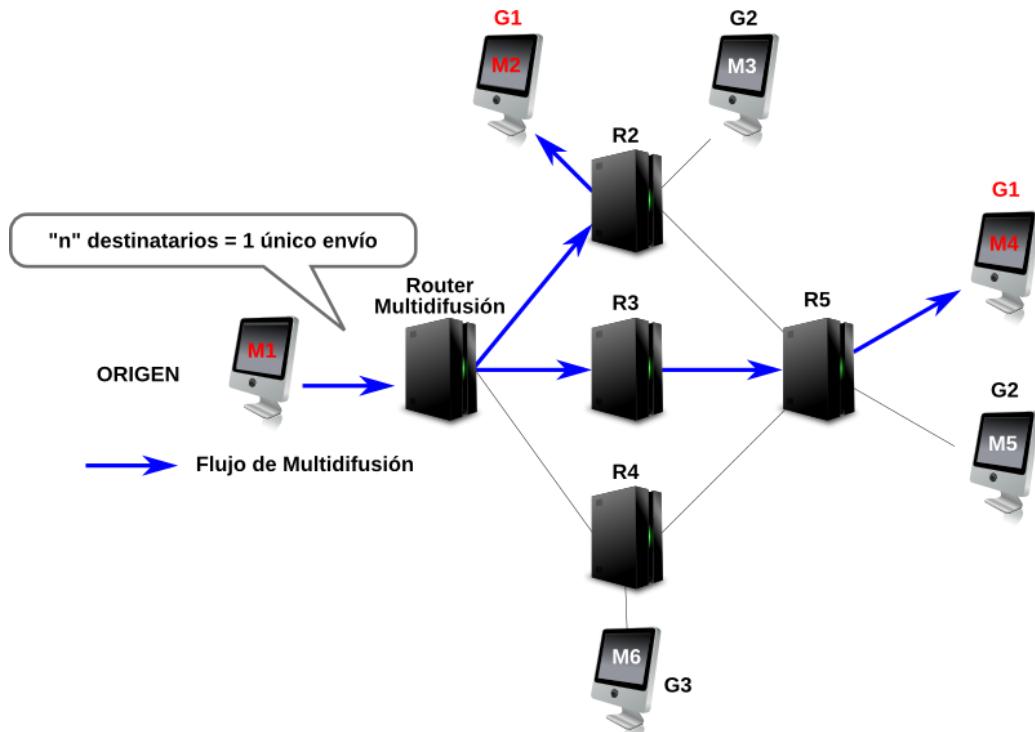
Tipos de transmisiones IPv4

Existen tres tipos de transmisiones IPv4:

- **Unidifusión (Unicast):** Se basa en una transmisión punto a punto desde un sistema final origen a un sistema final destinatario. Si hay “n” destinatarios hay que transmitir “n” copias (“n” transmisiones) de la misma información desde el sistema origen. En el siguiente ejemplo se desean realizar dos transmisiones de unidifusión basadas en el envío de un datagrama IP desde M1 a M2 y M4. Por consiguiente, hay que enviar una copia de la misma información desde M1 a M2 y otra más desde M1 a M4 todo ello, a través de los pertinentes routers R1-R2 y R1-R3-R5.

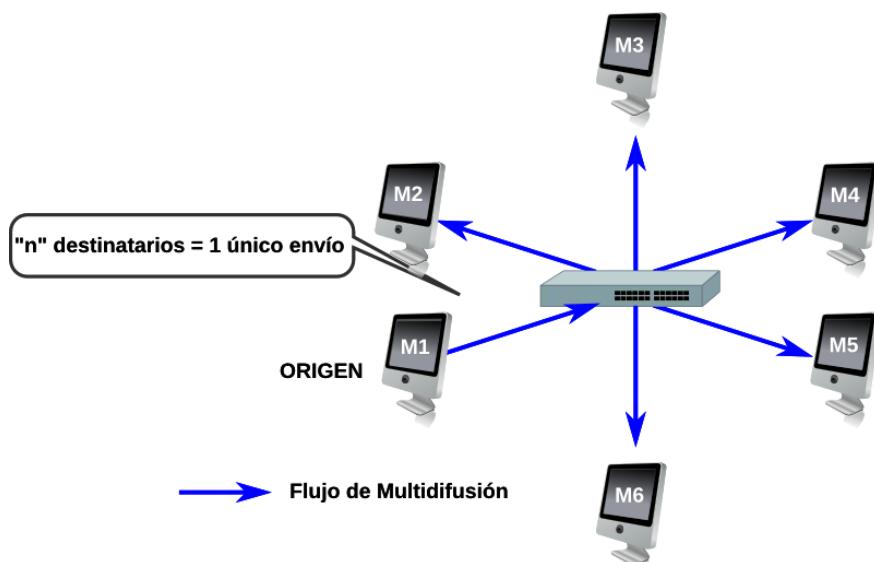


- **Multidifusión (Multicast):** Transmisión, en un solo envío, desde un sistema final origen a un grupo de sistemas destinatarios que forman un grupo de multidifusión por Internet y que comparten una misma dirección IP de multidifusión de grupo. Si hay “n” destinatarios en el grupo, sólo se transmite una vez la información desde el sistema origen. En este escenario, los routers de multidifusión por Internet tienen que poseer previamente la capacidad necesaria para hacer las copias de la información transmitida, desde el origen a los correspondientes sistemas destinatarios.
- En el siguiente ejemplo se desea realizar una transmisión de multidifusión basada en el envío de un único datagrama IP desde M1 a todas las máquinas (M2 y M4) que forman el grupo de multidifusión G1. Se resalta que R2 no transmite una copia a R5 porque tiene constancia (a través de un protocolo de encaminamiento dinámico de multidifusión) de que existe otro envío de la misma información por otro camino (R1-R3-R5).



- Difusión (Broadcast):** Es una transmisión, en un solo envío, desde un sistema final origen a todos los sistemas conectados a una misma red de difusión (*Etherneto WiFi*). Todo ello, sin necesidad de transmitir desde el origen una copia de la misma información, por separado, a cada uno de dichos sistemas. El problema de este tipo de difusión es que aparte de aumentar el tráfico por la red, la información transmitida llegará posiblemente a ciertos sistemas que no tienen el más mínimo interés por la información en cuestión. En la difusión, el envío de la información se realiza de una manera indiscriminada a todos los destinos, mientras que en la multidifusión, sólo se difunde la información a los destinos que hayan manifestado expresamente su interés en recibirla.

En el siguiente ejemplo se desea realizar una transmisión a todas las máquinas (M1, M2, M3, M4, M5 y M6). Al igual que con la multidifusión, pero de una forma menos selectiva, sólo se envía un datagrama IP desde M1 al resto de sistemas.



Direccionamiento IPv4

Todos los sistemas en Internet tienen que regir sus acciones de encaminamiento según el formato de direccionamiento IP indicado en el diseño del protocolo IP. El formato de una dirección IP manejada por el protocolo Ipv4 representa un modelo jerárquico de direccionamiento en dos niveles: redes y sistemas.



Las **direcciones IP**, también denominadas direcciones numéricas, se dan bajo la forma de cuatro octetos (32 bits) y cada octeto es un número entero decimal separado del anterior por un punto, de la forma siguiente:

octeto-1·octeto-2·octeto-3·octeto-4

Por ejemplo, 138.100.12.16 puede ser la dirección de un sistema en Internet o en una red privada TCP/IP.

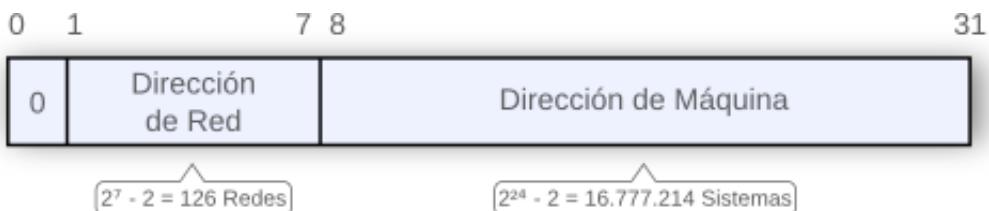
Clases de direcciones IPv4

En función de las tres clases de transmisiones y del número de bits utilizados para identificar redes y máquinas, las direcciones IP o direcciones numéricas se clasifican en cinco clases:

- Clase A: Unidifusión y difusión.
- Clase B: Unidifusión y difusión.
- Clase C: Unidifusión y difusión.
- Clase D: Multidifusión.
- Clase E: Experimental o reservada.

Direcciones IP de la clase A

Las **direcciones IP de la clase A** permiten definir $2^7 - 2 = 126$ redes diferentes (los números en decimal 0 y 127 están reservados) y $2^{24} - 2 = 16.777.214$ máquinas en cada una (los números en decimal 0 y 16.777.215 están reservados). El identificador 0 de la clase A se corresponde con el bit más significativo en la secuencia de 8 bits de los números en decimal del 0 al 127. Se utiliza para redes “grandes” capaces de conectar un gran número de máquinas.

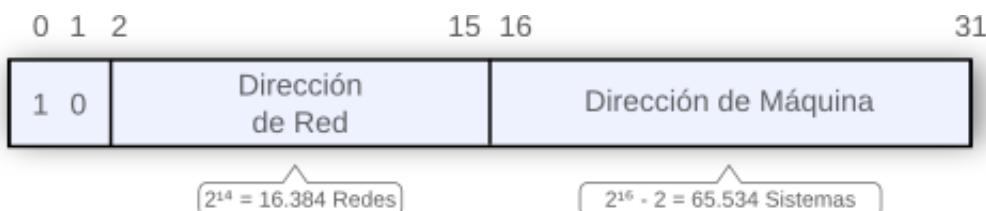


En los 7 bits de la dirección de red de la clase A no puede aparecer dos combinaciones, ni “todo a ceros” en binario (0 en decimal), ni “todo a unos” en binario (127 en decimal)” porque son dos direcciones de red reservadas. A su vez, en los 24 bits de la dirección de máquina no puede aparecer dos combinaciones, ni “todo a ceros” en binario (0.0.0 en decimal), ni “todo a unos” en binario (255.255.255 en decimal)” porque son dos direcciones

de máquinas reservadas. Por tanto, el rango práctico de direcciones de redes de la clase A es del 1.0.0.0 al 126.0.0.0.

Direcciones IP de la clase B

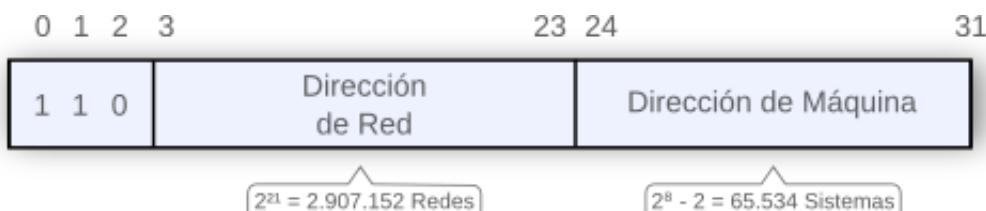
Las **direcciones IP de la clase B** permiten identificar $2^{14} = 16.384$ redes y $2^{16} - 2 = 65.534$ máquinas en cada una (los números en decimal 0 y 65.535 están reservados). El identificador 10 de la clase B se corresponde con los dos bits más significativos en la secuencia de 8 bits de los números en decimal del 128 al 191. Se utiliza para redes de “tipo medio”.



En los 16 bits de la dirección de máquina de la clase B no puede aparecer dos combinaciones, ni “todo a ceros” en binario (0.0 en decimal), ni “todo a unos” en binario (255.255 en decimal) porque son dos direcciones de máquinas reservadas. Por tanto, el rango práctico de direcciones de redes de la clase B es del 128.0.0.0 al 191.255.0.0.

Direcciones IP de la clase C

Las **direcciones IP de la clase C** permiten definir $2^{21} = 2.097.152$ redes y $2^8 - 2 = 254$ máquinas en cada una (los números en decimal 0 y 255 están reservados). El identificador 110 de la clase C se corresponde con los tres bits más significativos en la secuencia de 8 bits de los números en decimal del 192 al 223. Se utiliza para redes “pequeñas”.



Finalmente, en los 8 bits de la dirección de máquina de la clase C no puede aparecer dos combinaciones, ni “todo a ceros” en binario (0 en decimal), ni “todo a unos” en binario (255 en decimal) porque son dos direcciones de máquinas reservadas. Por tanto, el rango práctico de direcciones de redes de la clase C es del 192.0.0.0 al 223.255.255.0.

Direcciones IP de la clase D y E

Las direcciones IP de la clase D (RFC-2365) se utilizan en comunicaciones de multidifusión (o multicast) con el objetivo de enviar una misma información sin copias desde la máquina origen a todos los miembros del grupo (posiblemente, dispersos geográficamente en múltiples redes por Internet). Todos los miembros del grupo comparten una misma dirección IP de la clase D, independientemente de su ubicación geográfica en Internet, ya que a todos ellos les identifica individualmente su dirección IP clase A, B o C de unidifusión.

1 1 1 0	Dirección del Grupo de Multidifusión
---------	--------------------------------------

Los primeros 4 bits de una dirección de multidifusión (1110) se corresponden con el identificador de la clase. Los 28 bits restantes especifican un grupo de multidifusión en particular sin contener una dirección de red como en las direcciones de la clase A, B y C. El rango completo de direcciones de multidifusión es: 224.0.0.0---239.255.255.255.

Conviene resaltar que las direcciones de multidifusión sólo pueden emplearse como direcciones de IP de destino. En la dirección IP de origen aparece la dirección IP clase A, B o C de unidifusión de la máquina origen de la multidifusión.

Finalmente, las **direcciones IP de la clase E** son direcciones experimentales para un uso futuro. El rango completo de este tipo de direcciones IP es: 240.0.0.0---255.255.255.255.

0 1 2 3 4	31
1 1 1 1	Uso Futuro

Direcciones IP reservadas

Existen tres tipos de direcciones IP reservadas:

- **0.0.0.0:** Dirección reservada de red de la clase A. Tiene los siguientes usos reservados:
 - Ruta por omisión en una tabla de encaminamiento IP. El datagrama IP se reenvía al siguiente *router* especificado en dicha tabla.
 - Ruta directa a través del propio *router* en una tabla de encaminamiento. El datagrama IP se encamina directamente a una máquina vecina por la red de acceso.
 - Solicitud de información TCP/IP de configuración de un cliente DHCP a su servidor DHCP. Dicha información incluye una dirección IP temporal en dicha red.
- **127.0.0.0:** Dirección reservada de red de la clase A. Tiene el siguiente uso reservado:
 - Dirección de bucle (*loopback*) en la propia máquina local para pruebas de procesos servidores locales y desarrollo de aplicaciones cliente y servidor.
- **Todo a ceros y todo a unos en la parte local de máquina:** Direcciones reservadas de máquinas de la clase A, B y C. Tienen los siguientes usos reservados:

red.0.0.0: Identifica a una red de la clase A.

red.red.0.0: Identifica a una red de la clase B.

red.red.red.0: Identifica a una red de la clase C.

red.255.255.255: Identifica una difusión dirigida a una red de la clase A.

red.red.255.255: Identifica una difusión dirigida a una red de la clase B.

red.red.red.255: Identifica una difusión dirigida a una red de la clase C.

Direcciones IP especiales

Existen tres tipos de direcciones especiales:

- **0.0.0.0:** Dirección reservada y, además, *especial* para el protocolo DHCP. Esta dirección la utiliza un cliente DHCP para solicitar información de configuración TCP/IP a su servidor DHCP. Dicha información incluye una dirección IP temporal en dicha red. En concreto, un cliente DHCP en su mensaje de solicitud pone “*todo a ceros*” en el campo dirección IP del cliente y todo a “*unos*” en el campo dirección IP del servidor.
- **Difusiones a una red:** Existen dos tipos de direcciones especiales para transmisiones de difusión:
 - **Difusiones dirigidas (Directed Broadcast):** Se pone todo a “*unos*” (en binario) en la parte de máquina de la dirección IP destino:
red.255.255.255: Difusión dirigida a una red de la clase A.
red.red.255.255: Difusión dirigida a una red de la clase B.
red.red.red.255: Difusión dirigida a una red de la clase C.
 - **Difusiones limitadas (Limited Broadcast):** Se pone todo a “*unos*” (en binario) en los 32 bits de la dirección IP destino:
255.255.255.255: Difusión limitada a la red de acceso independientemente de la clase de dirección IP.
- **127.x.x.x:** Dirección de bucle (*loopback*) reservada y especial para autochequeos, pruebas y comprobaciones de procesos servidores. Asimismo, también es una dirección reservada y especial para el desarrollo de aplicaciones cliente y servidor sin necesidad de una red de comunicaciones que conecte al proceso cliente con el proceso servidor de la aplicación. La red de bucle, 127.0.0.0 es una red que no existe físicamente, se asume que está dentro de la propia máquina en donde se ejecutan los procesos servidores y/o se está desarrollando tanto la parte cliente como la parte servidora. El proceso cliente tendrá una dirección en esa nube ficticia (por ejemplo, 127.x.x.x) y el proceso servidor tendrá la misma u otra dirección en esa red (por ejemplo, 127.y.y.y). De esta forma, se prueba la interacción entre los dos procesos sin penalizar el tráfico de una red real.

Direcciones simbólicas: DNS

Aparte de su dirección IP, una máquina puede disponer de una **dirección simbólica**. Las direcciones simbólicas no se corresponden con los nombres de las máquinas por las que deben pasar los datagramas IP hasta llegar al destino; sino que se basan en una jerarquía de dominios nemotécnicos que se corresponden con las organizaciones relacionadas con la situación geográfica-administrativa de la máquina destinataria y que la identifican simbólicamente (por ejemplo, departamento, universidad, país).

De forma general, la dirección simbólica de una máquina se representa mediante una cadena de dominios con el siguiente formato:

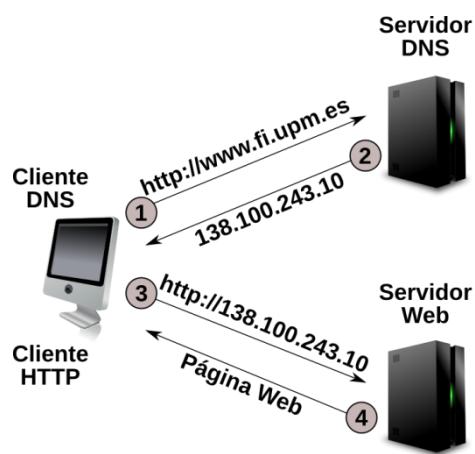
máquina·dominio(n)·dominio(n-1)·...·dominio(1)·dominio

Los dominios se ordenan de derecha a izquierda, del más general al más particular, separando cada nombre simbólico del anterior mediante un punto. A su vez, en el nombre de máquina se pone la dirección simbólica del sistema en cuestión.

El **sistema DNS** (*Dominio Name System*) o *Sistema de Nombres de Dominio* de Internet consiste en una jerarquía de nombres simbólicos o de dominios nemotécnicos de máquinas, repartida en una base de datos (BD) distribuida mediante servidores DNS por toda la red Internet. Esta base de datos distribuida es consultada por las aplicaciones de usuario para llevar a cabo la traducción entre los nombres simbólicos y las correspondientes direcciones IP. De esta forma se permite que un usuario escriba la dirección de un sistema destinatario en formato simbólico y no en formato IP o numérico que es más difícil de recordar.

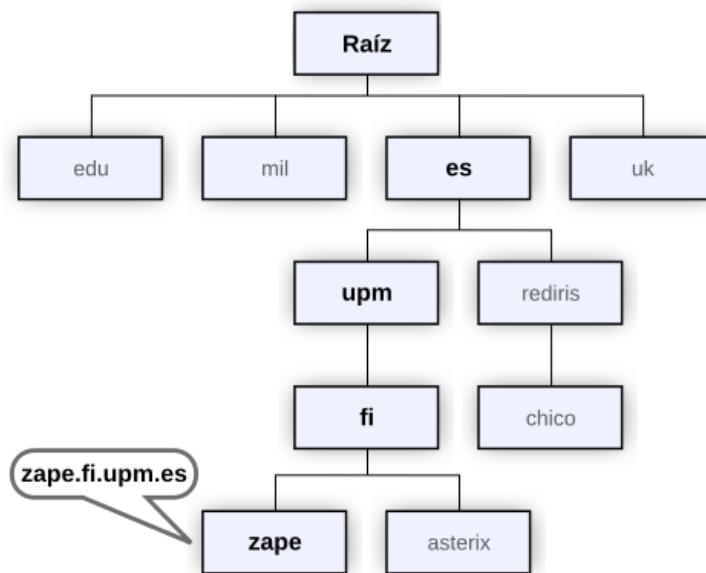
El procedimiento de actuación descansa en unos sistemas específicos denominados **servidores DNS** (*name servers*) o servidores de nombres. Generalmente, cada organización conectada a Internet dispone de su propio servidor DNS que hace la traducción de una dirección simbólica de una máquina de dicha organización en su correspondiente dirección IP que, por otro lado, es la dirección con la que trabajan, finalmente, todos los sistemas TCP/IP. Los típicos tres pasos son los siguientes:

1. El usuario escribe, en su navegador web (Internet Explorer, Mozilla Firefox, etc.), la dirección simbólica, por ejemplo, www.fi.upm.es. A continuación, el cliente DNS de la máquina del usuario consulta a su servidor DNS.
2. El servidor DNS contesta con una dirección IP: 138.100.243.10.
3. El navegador utiliza la anterior dirección IP para localizar el sitio web de la Facultad de Informática de la UPM.



Se destaca que ningún servidor DNS contiene la BD completa; cuando un servidor DNS no conoce la dirección simbólica, se comunica inmediatamente con otro servidor DNS de superior jerarquía y éste a su vez con el siguiente y así hasta encontrar o no la dirección IP asociada a la dirección simbólica de partida.

La distribución jerárquica DNS permite crear diferentes niveles o dominios de gestión o de responsabilidad para facilitar dicha gestión y garantizar la unicidad de nombres.



Así, en el nivel superior se encuentran los **TLD** (*Top Level Domain*) o dominios de primer nivel, clasificados por IANA en los siguientes dos tipos:

- **TLD genéricos** (*gTLD: generic Top Level Domains*). Están formados por tres o más caracteres. Algunos gTLD se corresponden con los dominios originales que se diseñaron inicialmente en Internet antes de que esta red traspasara las fronteras de los EE UU. Este es el caso de los siguientes seis TLD originales:
 - .edu**: Instituciones educacionales o docentes.
 - .mil**: Organizaciones militares asociadas al ejército de los EE UU.
 - .gov**: Organizaciones gubernamentales de los EE UU.
 - .org**: Organizaciones no lucrativas.
 - .com**: Organizaciones comerciales.
 - .net**: Centros de red y apoyo en Internet
- **TLD de códigos de países** (*ccTLD: country code Top Level Domains*) o **TLD geográficos**. Están formados por dos caracteres. Por ejemplo, **.es** para [España](#).

IANA es el gestor de la raíz DNS y responsable de coordinar a las distintas delegaciones en función de sus políticas y procedimientos. IANA ofrece un sitio web que incluye un motor de búsqueda para realizar consultas relacionadas con cualquier TLD existente en la actualidad: <http://www.iana.org/domains/root/db/> .

Por debajo del nivel superior formado por los TLD, se encuentran los dominios correspondientes a las distintas organizaciones conectadas a Internet dentro de cada país. El administrador de cada dominio, que mantiene su parte correspondiente de la BD distribuida, es responsable del registro de nombres de dominio dentro de su nivel, garantizando que éstos sean únicos.

En España, aparte del ESNIC (<http://www.nic.es>), existe un conjunto numeroso de empresas o agentes registradores acreditados por Red.es e ICANN y autorizados por ESNIC-Red.es. Los agentes registradores son intermediarios en los procedimientos relacionados con el registro de

nombres de dominio directamente bajo .es y, además, permiten “enganchar” una dirección, asimismo, bajo otros dominios como .com, .net, .org, etc.

Asimismo, conviene destacar que IANA (junio de 2008) ha liberalizado la creación de nuevos dominios. Tal es el caso de .madrid, .africa o extensión.apellido. El único problema que tiene esta novedad es que el coste de creación de un nuevo dominio tiene, de momento, un precio de partida de 60.000 euros.

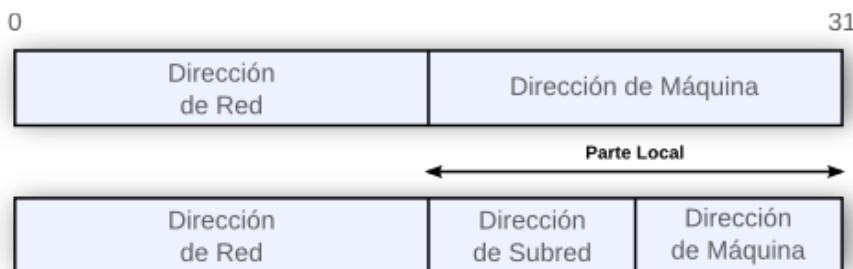
Subredes y máscaras

Para empezar es importante tener muy claro el concepto de subred y el porqué de su uso. Las características más relevantes asociadas a una subred son:

- Cuando no se desean tener todas las máquinas conectadas a la misma red de una organización y, por ejemplo, se desea una red por departamento u oficina; entonces, se crean tantas subredes o subconjuntos de dicha red como departamentos u oficinas existan.
- Una subred es un “red de comunicaciones más pequeña que la red original” y se crea a partir de la dirección IP asignada a la red de una organización.
- Una subred es un subconjunto de la red de comunicaciones, clase A, B o C, de una organización y, por tanto, es un subconjunto de la dirección IP asignada a la red de una organización.

Un administrador crea sus propias subredes, y asigna direcciones IP a dichas subredes, a partir de la dirección IP de la red de dicha organización y del número de ceros de la máscara asociada a dicha dirección IP. Para ello, divide la parte local o dirección de máquina en dos partes:

- Dirección de red: Para las subredes que se van a crear.
- Dirección de máquina: Para las máquinas que se van a conectar a dichas subredes.



En función de lo anterior, si se desean crear más subredes que máquinas conectadas a dichas subredes, se utilizarán más bits para la parte de dirección de subred. A su vez, si se quieren conectar más máquinas que subredes creadas, se utilizarán más bits para la parte de dirección de máquina.

En este contexto, se establece un criterio de distribución de bits en función:

- De la clase a la que pertenece la dirección de Internet asignada oficialmente a la red de la organización y la máscara asociada para obtener la parte local de dicha dirección.
- Del número de subredes que se desean crear y número de máquinas que se quieran conectar a dichas subredes.

- De las direcciones reservadas para la parte local de la dirección IP (“todo a ceros y unos”) que incluye tanto la parte local de subred como la parte local de máquina (RFC-950).

A continuación se muestran las restricciones existentes a la hora de crear subredes. Si se parte del hecho de que “todo a ceros” en binario (dirección de red) y “todo a unos” en binario (difusión dirigida) son dos direcciones de máquinas reservadas; dichas direcciones también afectan a las subredes en la parte local de subred y máquina. Por tanto, en principio, para seguir manteniendo la compatibilidad, el estándar RFC-950 recomienda, en la parte local (subred y máquina), tener en cuenta lo siguiente:

- Dirección de subred: No se debe poner “todo a ceros” (en binario) para identificar a una subred ya que en la parte local de máquina podría aparecer, también, “todo a ceros” y sería una dirección de red. Igualmente, no se debe poner “todo a unos” (en binario) para identificar a una subred ya que en la parte local de máquina podría aparecer, también, “todo a unos” y sería una difusión dirigida a una red. Se resalta que si no se usa el “todo a ceros” y “todo a unos” en la parte local de dirección de subred se pierden direcciones para identificar máquinas.
- Dirección de máquina: No se debe poner “todo a ceros” (en binario) para identificar a una máquina, ya que en la parte local de dirección de subred se podría haber tecleado también “todo a ceros” y, por tanto, sería una dirección de red. En el caso de que en la dirección de subred no apareciera “todo a ceros” y en la dirección de máquina sí, se estaría ante el caso de una dirección de subred. Igualmente, no se debe poner “todo a unos” (en binario) para identificar a una máquina, ya que en la parte local de dirección de subred podría aparecer también “todo a unos” y sería una difusión dirigida a una red. Asimismo, en el caso de que en la dirección de subred no apareciera “todo a unos” y en la dirección de máquina sí, se estaría ante el caso de una difusión dirigida a una subred.

Como recomendación para evitar cualquier tipo de problema, especialmente, con las difusiones dirigidas, se sugiere lo siguiente:

- Restar, siempre que se pueda, las dos direcciones reservadas (“todo a ceros” y “todo a unos”) en la parte local de la dirección de subred, según recomienda el estándar RFC-950 en Internet. Si al restar estas dos direcciones reservadas, se comprueba que no se dispone del suficiente rango de direcciones IP para identificar a todas las máquinas que se deseen conectar; entonces, no se sigue la recomendación RFC-950 y se hace uso de las dos direcciones reservadas (“todo a ceros” y “todo a unos”) en la parte local de la dirección de subred. Se resalta que el estándar RFC-950 es una recomendación y no una obligación.
- Obligatoriamente, restar siempre (tanto para redes como para subredes) las dos direcciones reservadas (“todo a ceros” y “todo a unos”) en la parte local de la dirección de máquina. Una máquina no puede tener nunca una dirección de subred ni de difusión dirigida a una subred.

Máscara de subred

Una **máscara de subred** es un número de 32 bits que contiene “unos” en los bits que identifican a la dirección de subred y “ceros” en los bits que identifican a la dirección de máquina en dicha subred.

Seguidamente se muestran los conceptos más importantes asociados a una máscara de subred:

- Los “unos” de una máscara de subred indican los bits de la dirección IP que no se pueden “tocar”
- Los “ceros” de una máscara de subred indican los bits de la dirección IP que se pueden “tocar” para direccionar máquinas (o más subredes y máquinas en dichas subredes)
- Una dirección de subred también se puede utilizar para crear más subredes a partir de su máscara
- El estándar RFC-950 recomienda para una mayor comprensión de las máscaras y las tablas IP de encaminamiento que los bits a “unos” que identifican a las direcciones de subred sean contiguos
- Toda máscara de subred, clase A, B y C, tiene siempre más “unos” que la correspondiente máscara por omisión, clase A, B, y C, de la dirección IP de partida
- Toda máscara de subred (registrada en la tabla de encaminamiento) facilita las labores de encaminamiento mediante la aplicación de la operación lógica AND a la dirección destino y máscara correspondiente

Las máscaras de red, por omisión, para las direcciones de la clase A, B y C son:

Clase de dirección	Máscara por omisión (Decimal)
A	255. 0.0.0
B	255.255. 0.0
C	255.255.255. 0

Máscaras de subred de longitud variable

Hasta el momento, se ha estado asignando una misma cantidad máxima de máquinas a todas las subredes que se han ido creando; pero también es posible asignar un número variable. Según esto, se pueden distinguir dos tipos de máscaras de subred:

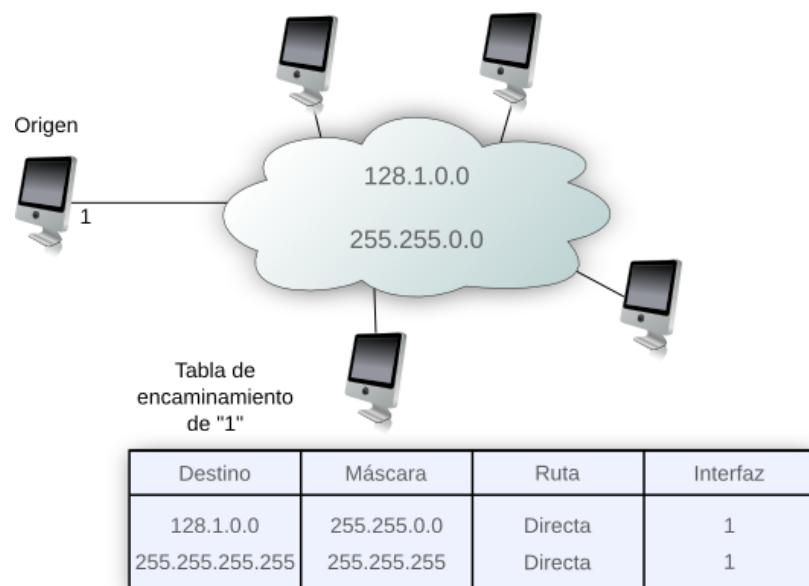
- **De longitud fija:** Se basa en una división estática de subredes (subredes estáticas). Todas las redes, lo necesiten o no, disponen del mismo número de direcciones para máquinas. Para ello, se aplica una misma máscara de subred a todas las subredes creadas, con lo cual se asigna la misma cantidad de máquinas a dichas subredes.
- **De longitud variable (VLSM: Variable Length Subnet Masks):** Se fundamenta en una división dinámica o variable de subredes (subredes dinámicas). Proporciona al administrador una forma más óptima y flexible de asignar direcciones numéricas, permitiéndole asignar cantidades variables de máquinas en las subredes creadas. Para

ello, se aplican diferentes máscaras a las subredes creadas, con lo cual se asigna un número diferente de máquinas a dichas subredes.

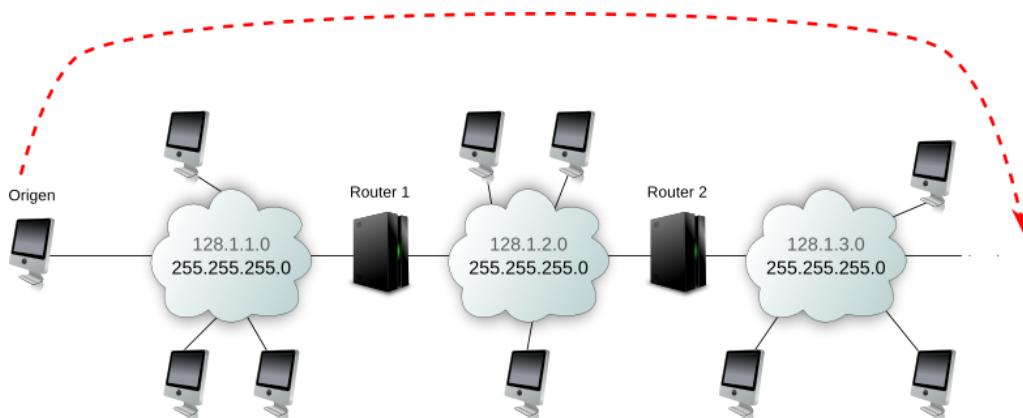
Tipos de difusiones a redes y subredes

Las difusiones se emplean en una RAL de difusión ya sea *Ethernet* o *WiFi*. Existen dos tipos de difusiones:

- **Difusión limitada (broadcast):** permite enviar un mismo mensaje de difusión a todas las máquinas de la red o subred de acceso a la cual está conectada la máquina origen de la difusión. Consiste en poner “todo a unos” en los 32 bits de la dirección IP destino. Una difusión limitada, por ejemplo, a la red 128.1.0.0/255.255.0.0, sería: 255.255.255.255/255.255.255.255.



- **Difusión dirigida:** permite enviar un mismo mensaje de difusión progresivamente a todas las máquinas y subredes de una red o subred. Consiste en poner “todo a unos” en la parte local de máquina de la dirección IP destino. Se resalta que la difusión dirigida puede ser local (directamente a través de la red de acceso) o remota (indirectamente a través de uno o más routers); a diferencia de la difusión limitada que es siempre local.



Para terminar con las difusiones dirigidas es importante tener en cuenta los siguientes puntos:

- Las difusiones dirigidas están diseñadas, fundamentalmente, para las subredes, aunque se puede enviar una difusión dirigida exclusivamente a una red o a una subred.
- Una difusión dirigida a subredes exige una difusión progresiva desde la primera subred al resto de las subredes conectadas.
- En el caso de subredes anidadas, la máquina origen y todos los *routers* deben tener una misma entrada en la tabla de encaminamiento.
- Una difusión dirigida exclusivamente a una red o subred (sin subredes anidadas) tiene el mismo efecto que una difusión limitada.

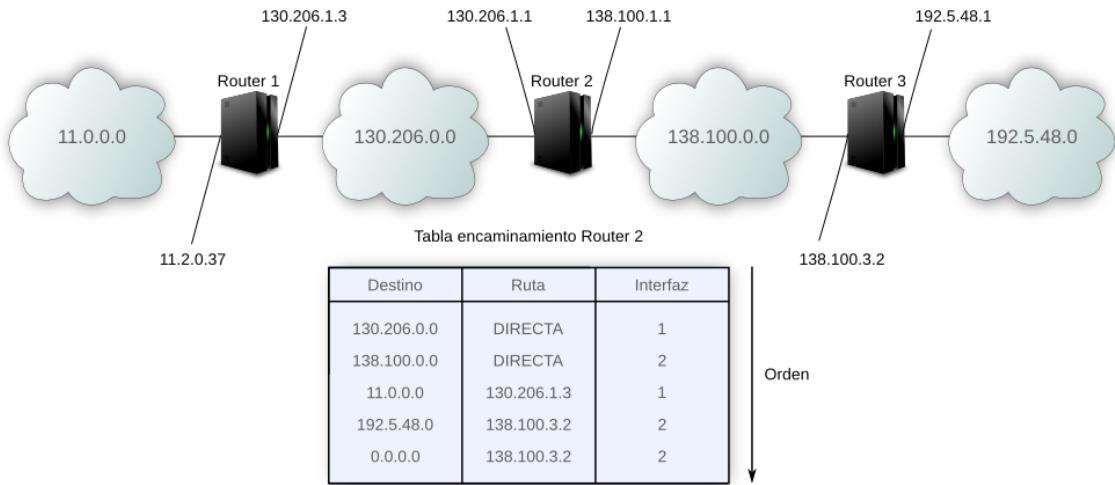
Actualmente, las difusiones dirigidas no se suelen utilizar. Los administradores suelen eliminar sus entradas en los *routers* correspondientes, por cuestiones de seguridad, para no hacer progresar dichas difusiones por todas las subredes y máquinas de la organización. Prácticamente, las únicas difusiones que se utilizan son las limitadas. Quiere decir esto, que si no se utilizan las difusiones dirigidas se puede utilizar todo el rango de direcciones en la parte local de subred y, por tanto, no seguir la recomendación RFC-950.

Tipos de encaminamiento

Existen tres tipos posibles de encaminamiento:

- **Directo:** Cuando la máquina destino está conectada a la misma red de acceso y la dirección de red de dicha máquina destinataria es conocida al aplicar la máscara. En este caso, no se transmite el datagrama IP a un *router* vecino (cuya dirección está registrada en el campo de Ruta de la tabla de encaminamiento) ya que la propia máquina es capaz de efectuar dicho encaminamiento. Una máquina es “vecina” de otra, cuando está conectada a la misma red que la considerada.
- **Indirecto:** Cuando la máquina de destino no está conectada a la misma red de acceso. Además, la dirección de red de dicha máquina destinataria es conocida y hay que pasar por el *router* vecino (campo Ruta) indicado en la tabla.
- **Por omisión:** Cuando la máquina de destino no está conectada a la misma red de acceso. Además, la dirección de red de dicha máquina destinataria no es conocida y hay que transmitir el datagrama IP a un *router* vecino (campo Ruta).

En la siguiente figura se muestra un ejemplo de 4 redes de comunicaciones y 3 *routers*, así como la tabla de encaminamiento de uno de ellos. Dicha tabla dispone de los tres tipos posibles de encaminamiento.



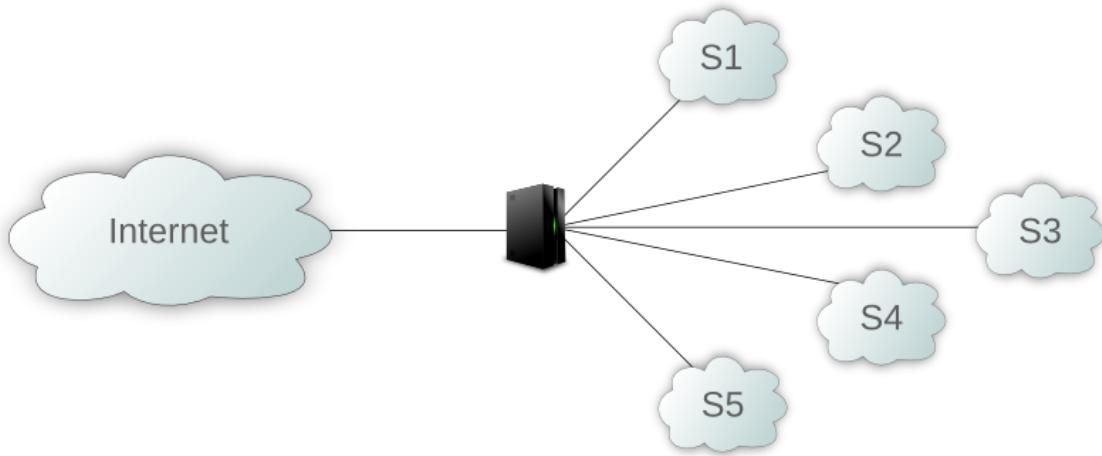
Teniendo en cuenta el escenario descrito, es importante resaltar lo siguiente:

- Cada *router* en su tabla de encaminamiento tiene registradas las direcciones IP de sus *routers* vecinos (para el *router 2* del ejemplo: 130.206.1.3 y 138.100.3.2). Mediante este procedimiento se van concatenando todas las redes de comunicaciones que forman la red Internet.
- Cada *router* dispone de dos o más direcciones numéricas (para el *router 2* del ejemplo: 130.206.1.1 y 138.100.1.1) en función del número de redes (130.206.0.0 y 138.100.0.0) a las que esté conectado.
- Un *router* puede saber si la máquina destinataria pertenece a su misma red de acceso comparando la dirección de red de dicha máquina con la suya. Si es la misma, transmite directamente y si no, indirectamente o por omisión a un *router* vecino. Un *router* excluyendo la máscara asociada al interfaz de entrada por donde ha llegado el datagrama IP, va aplicando, en un determinado orden, las máscaras de cada interfaz de salida hasta obtener o no una dirección incluida en la tabla. En el caso de no obtener una dirección conocida, se actúa por omisión si la entrada 0.0.0.0 está registrada en la tabla.

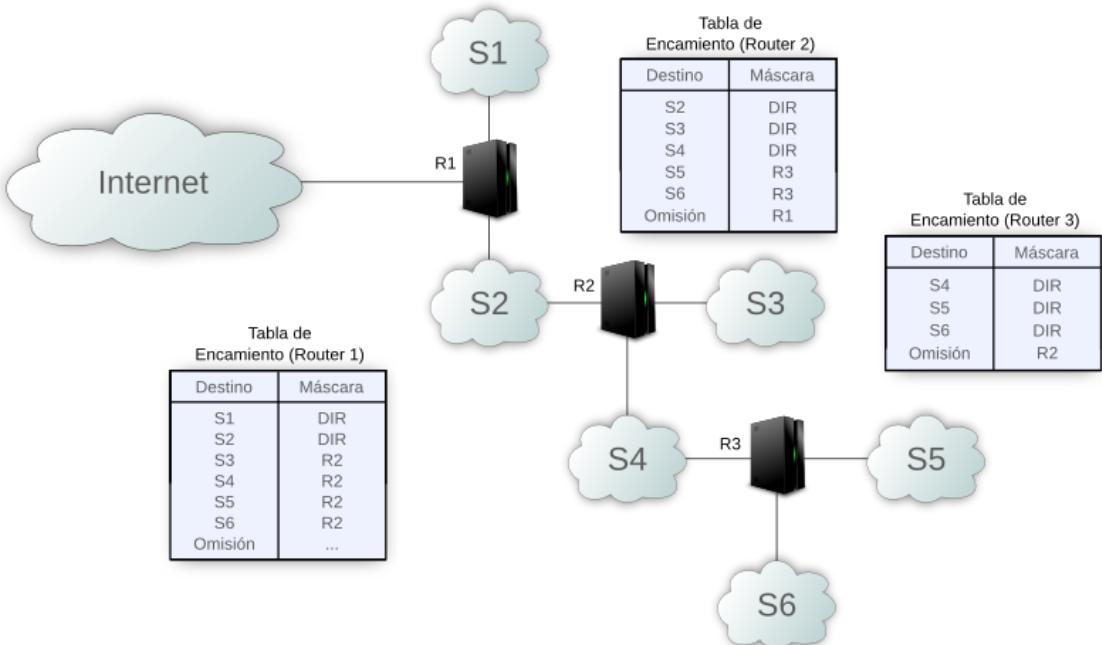
Asimismo, es importante resaltar que:

- Un *router* encamina los datagramas IP según la dirección de red del destino.
- La cantidad de información almacenada en un *router* es proporcional al número de redes conocidas por dicho *router*.
- Un *router* es transparente al usuario final.

Finalmente, es importante resaltar dentro del contexto de las tablas de encaminamiento, que cuando se crean subredes en una organización, el *router* de dicha organización debe tener tantas líneas, direcciones IP y entradas en la tabla de encaminamiento como subredes se hayan creado.



Una alternativa al escenario comentado es a través de un anidamiento de *routers* y subredes. Consecuentemente, para evitar depender de un solo *router* y que éste se encargue de todas las tareas del encaminamiento de la organización, se puede emplear más de uno, distribuyendo entre ellos todas las subredes. Cada *router* en su tabla contendrá, como informaciones más significativas, las direcciones de las subredes que conoce (Destino) y el tipo de acceso directo, indirecto o por omisión. Se asume que el *router* más externo (R1) contendrá más información que el más interno (R3) y una dirección por omisión a otro *router* vecino y superior en la jerarquía Internet para cualquier dirección externa a la organización.



Direccionamiento privado y traducción de direcciones (NAT)

El concepto del término **NAT** (*Network Address Translation*) o de traducción de direcciones de red también conocido como IP *masquerading* (RFC-3022) se basa en una traducción entre las direcciones IP privadas o internas de una organización y las direcciones IP públicas o externas asignadas de forma oficial y global en Internet. La idea es sencilla, si un número indeterminado de máquinas pertenecientes a una organización desean conectarse con el exterior y a cada una

de esas máquinas se le asigna de forma oficial y permanente una dirección IP pública en Internet, y si este mismo procedimiento se repitiera en todas las organizaciones conectadas a Internet; llegaría el momento que se agotaría el espacio oficial de direcciones IP públicas asignables. Debido al crecimiento exponencial de Internet cada vez resulta más complejo obtener direcciones IP públicas del ISP correspondiente. Asimismo, en muchas ocasiones no se desea disponer de un acceso directo completo a Internet por razones, fundamentalmente, de seguridad. Se han reservado, para una compartición común en redes privadas, tres bloques de direcciones privadas del espacio oficial de direcciones IP públicas:

- 10.0.0.0 hasta 10.255.255.255 (una única dirección de red de clase A)
- 172.16.0.0 hasta 172.31.255.255 (16 direcciones de red contiguas de clase B)
- 192.168.0.0 hasta 192.168.255.255 (256 direcciones de red contiguas de clase C)

Por consiguiente, la mayoría de las necesidades de conectividad de las organizaciones encajan en las siguientes categorías:

- **Conectividad global:** Todas las máquinas y redes de la organización disponen de direcciones IP públicas. Por tanto, las máquinas dentro de una organización tienen acceso tanto a máquinas internas como a máquinas externas de Internet. La organización que requiera conectividad global deben solicitar direcciones IP públicas a su ISP.
- **Conectividad privada:** Todas las máquinas y redes de la organización disponen de direcciones IP privadas que se pueden compartir con otras organizaciones y que por seguridad y privacidad se traducen por direcciones públicas cuando se accede al exterior.

Consecuentemente, existen 2 tipos de direcciones:

- **Públicas:** aquellas direcciones que son propias de Internet (oficiales o debidamente registradas para una conectividad global) y, por tanto, no pueden repetirse
- **Privadas:** aquellas direcciones que admiten su compartición en redes privadas diferentes.

Las máquinas de la red privada de una organización aunque internamente se pueden conectar sin problemas, no pueden, en principio, intercambiar datagramas directamente con el exterior porque podría haber direcciones repetidas. En este caso han de utilizar un dispositivo intermediario o *router* que haga de representante (proxy) y que se ocupe de realizar la traducción de direcciones o NAT. Un **router NAT** se coloca en la frontera del dominio de una organización, es decir, es el *router* más externo de dicha organización. Su misión es traducir las direcciones privadas en direcciones públicas y viceversa cuando las máquinas internas necesitan comunicarse con destinos en Internet. Una restricción que impone la tecnología NAT es que sólo puede haber un punto de comunicación entre la red privada y el exterior, es decir, la conexión sólo puede hacerse en un *router*.

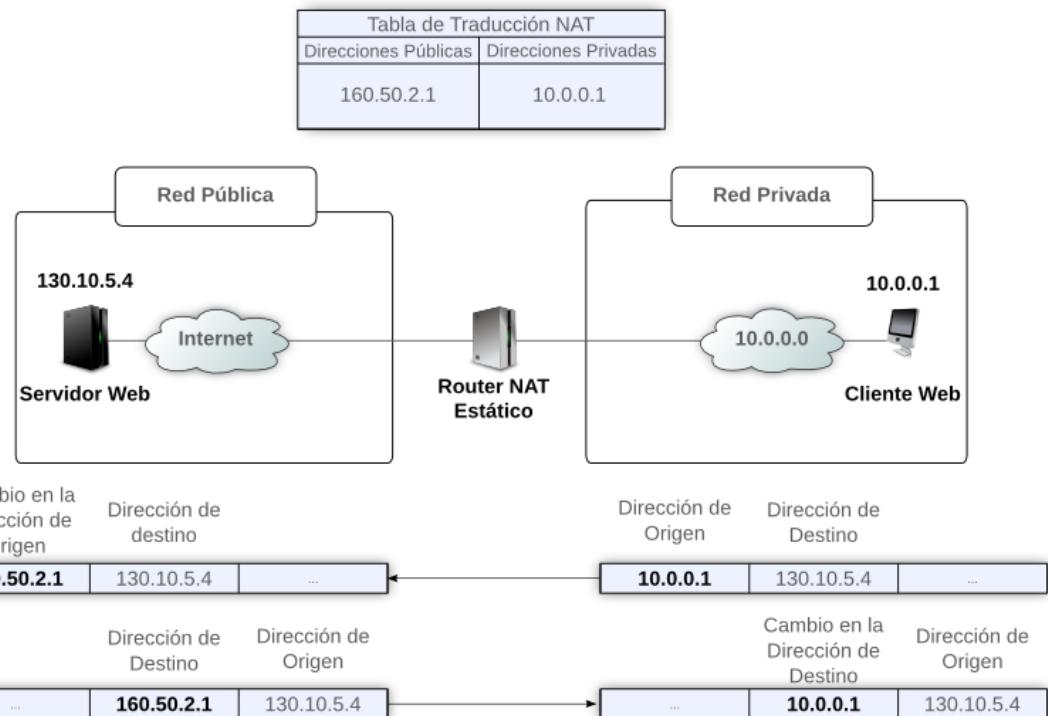
Teniendo en cuenta que muchas organizaciones que construyen redes privadas pueden utilizar las mismas direcciones IP privadas, pocas direcciones IP públicas únicas necesitan ser asignadas. En este escenario, las máquinas que tengan direcciones privadas pueden coexistir

con máquinas que tienen direcciones públicas. Las organizaciones pueden elegir convertir en privadas la mayoría de sus máquinas y mantener, a su vez, otras máquinas con direcciones públicas.

A la operación de traducción de direcciones IP se le denomina **NAT básica** o tradicional. Mediante una traducción NAT las diferentes máquinas privadas de una organización pueden acceder a Internet mediante una traducción del direccionamiento privado en direccionamiento público y viceversa.

El procedimiento de actuación del mecanismo de traducción NAT realiza un tratamiento diferente según que el datagrama IP sea de salida, es decir, desde la red privada hacia Internet; o de entrada, es decir, desde Internet hacia la red privada.

- Datagramas IP de salida (desde la red privada hacia Internet): Para cada datagrama IP de salida, el *router* NAT comprueba el campo de dirección de origen y analiza si dicha dirección coincide con alguna dirección privada de la organización, la cual tiene que estar registrada previamente en su tabla de traducción. En el caso afirmativo, el *router* NAT cambia la dirección de origen del datagrama IP de salida por la dirección pública asociada en su tabla.
- Datagramas IP de entrada (desde Internet hacia la red privada): Para cada datagrama IP de entrada, el *router* NAT analiza el campo de dirección de destino y comprueba si la dirección de destino coincide con alguna dirección pública de la organización, la cual tiene que estar registrada en su tabla de traducción. En el caso afirmativo, el *router* NAT cambia la dirección de destino del datagrama IP de entrada por la dirección privada asociada en su tabla.



Tipos de traducciones de direcciones IP

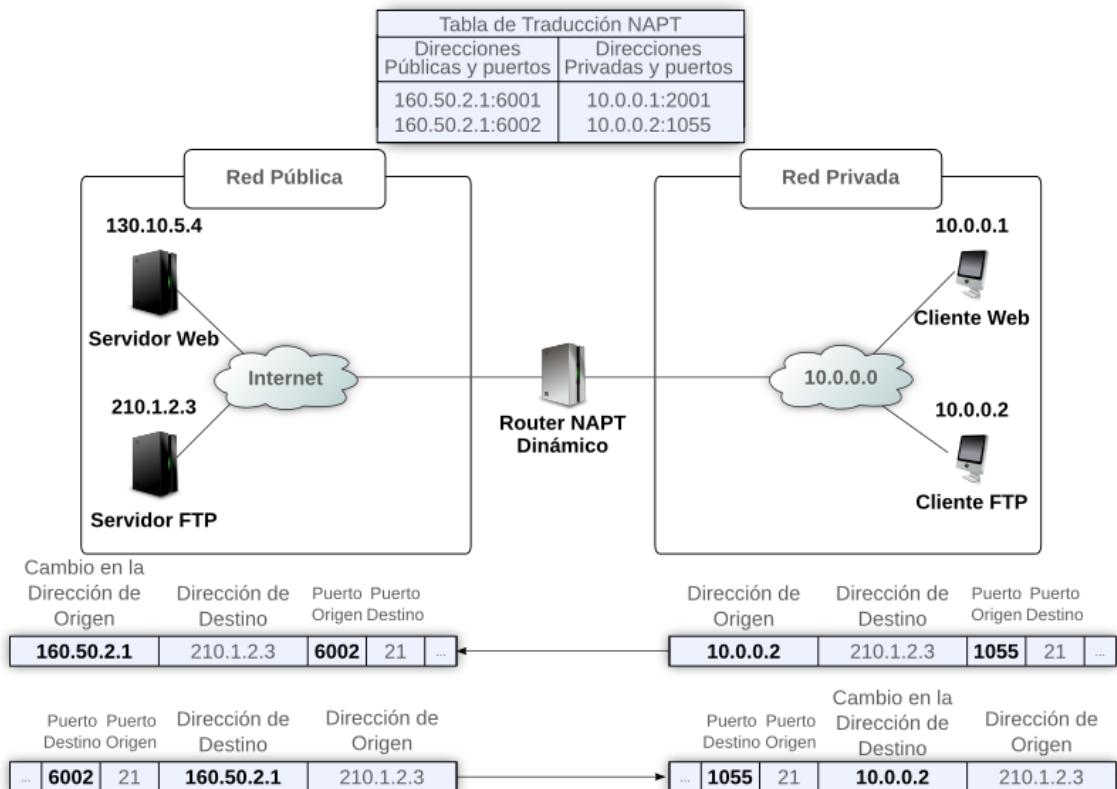
Un *router NAT* trabaja, únicamente, en el nivel de red realizando estáticamente la traducción de las direcciones de origen de los datagramas IP salientes y las direcciones de destino de los datagramas IP entrantes. En este contexto, existen tres tipos de traducciones de direcciones IP:

- Por el modo de funcionamiento:
 - **Traducción unidireccional:** Comunicaciones unidireccionales salientes para clientes internos
 - **Traducción bidireccional:** Comunicaciones bidireccionales para ofrecer al exterior servidores internos
- Por el modo de traducción:
 - **Traducción estática:** Asociación permanente y manual, efectuada previamente por el administrador, con un número de direcciones públicas igual al de direcciones privadas
 - **Traducción dinámica:** Asociación temporal y automática con reutilización de direcciones públicas según se vayan liberando
- Por el nivel de comunicaciones:
 - **Traducción básica de dirección o NAT:** Nivel IP o nivel de Red
 - **Traducción de dirección y puerto o NAPT (Network Address Port Translation):** Nivel IP y Nivel de Transporte (números de puerto). Con una dirección pública se pueden representar hasta 65.535 sistemas privados asociando dinámicamente

En función de todo lo anterior, es decir, del modo de funcionamiento, modo de traducción y nivel de comunicaciones, se obtienen varios modos operativos NAT. A continuación se describen los dos modos más utilizados:

- **NAPT dinámica unidireccional:** Combina las funcionalidades de la traducción NAPT, traducción NAT dinámica y traducción NAT unidireccional para los procesos clientes en máquinas privadas de la organización. Un *router* NAPT dinámico unidireccional trabaja, también, en el nivel de red y transporte al igual que un *routerNAPT* estático. La traducción NAPT dinámica es útil, cuando se dispone de una única dirección pública para una red privada y se desean conectar a Internet todas las máquinas de dicha red privada (“relación de muchos a uno”).

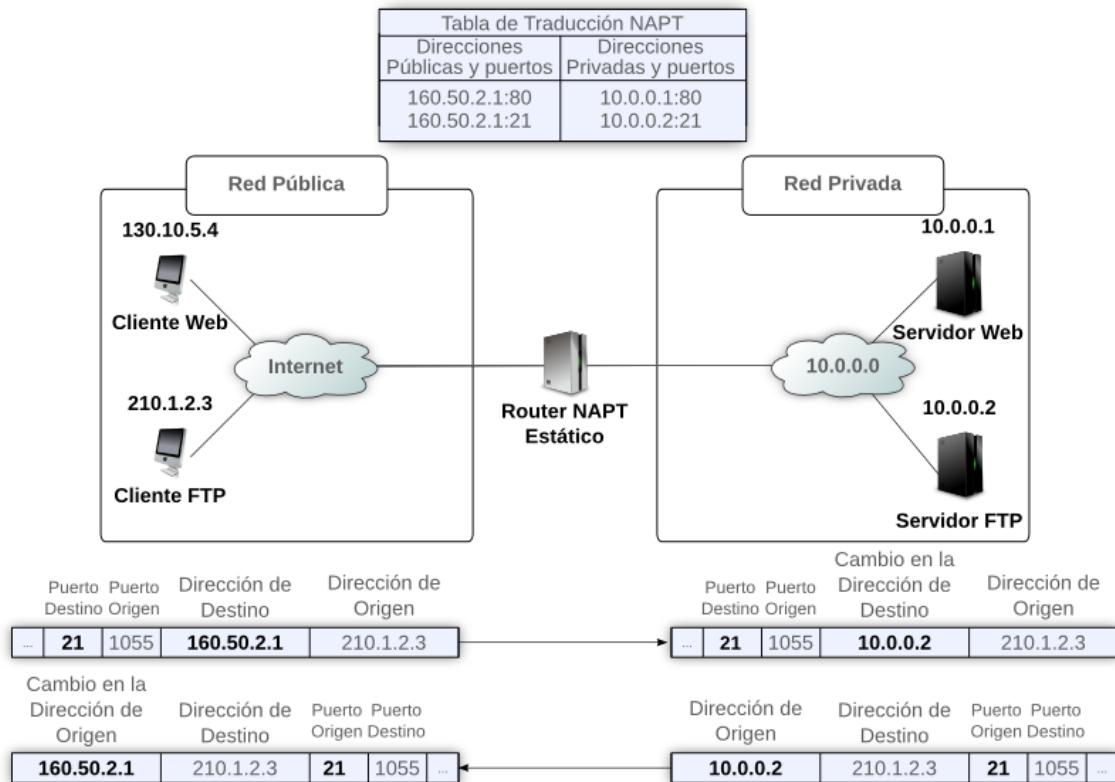
En el ejemplo se muestran dos máquinas privadas (10.0.0.1 y 10.0.0.2) compartiendo una única dirección pública (160.50.2.1).



- **NAPT Estática bidireccional:** Combina las funcionalidades de la traducción NAPT, traducción NAT estática y traducción NAT bidireccional para ofrecer servidores internos. Por tanto, las entradas en la tabla de traducción NAPT incluyen no sólo la dirección IP sino también el número de puerto (TCP o UDP). Por consiguiente, un *routerNAPT* estático trabaja en el nivel de red y transporte y es, especialmente útil, cuando se desean hacer accesibles con una única dirección pública aquellas máquinas servidoras privadas de la organización que necesitan estar visibles para el exterior. Para ello, se asigna de manera previa y permanente, por un lado, una dirección pública a una privada de la organización y, por otro lado, un mismo número de puerto público al número de puerto privado. Se resalta que el número de puerto privado tiene que ser conocido en Internet y, por tanto, tiene que ser el mismo que el número de puerto público.

En el ejemplo siguiente existen dos máquinas servidoras privadas (10.0.0.1 y 10.0.0.2) y en cada una de ellas un proceso servidor diferente. Consecuentemente, se necesita una única dirección pública (160.50.2.1) que han de compartir, con diferente número de puerto, las dos máquinas privadas (10.0.0.1 y 10.0.0.2). Asimismo, se muestra el

intercambio de datagramas IP entre el cliente externo FTP y el servidor privado FTP de la organización.



Asimismo, un *router NAPT* tiene que hacer las siguientes modificaciones:

- Cabecera IP: Al modificar las direcciones de origen y/o destino, el valor del campo suma de comprobación cambia en la cabecera del datagrama IP y por tanto, debe recalcularse y cambiar dicho campo de la cabecera por el nuevo valor.
- Cabecera de transporte (TCP/UDP): Asimismo, al modificar las direcciones de origen y/o destino, el valor del campo suma de comprobación en la cabecera de un segmento TCP o datagrama UDP también cambia con lo dicho campo debe calcularse de nuevo ya que la pseudocabecera incluye las direcciones IP de origen y destino.
- Mensajes ICMP: Como un mensaje ICMP encapsula la cabecera del datagrama IP y el comienzo de la cabecera TCP/UDP que originó dicho mensaje ICMP; NAT debe localizar en el mensaje ICMP la dirección IP y modificarla. Asimismo, debe cambiar la suma de comprobación de la cabecera IP encapsulada. A su vez, NAPT ha de modificar también el número de puerto TCP o UDP incluido en la cabecera encapsulada.

Por último, aunque NAPT representa un mecanismo muy útil, a veces se plantean problemas de difícil solución que hacen que determinadas aplicaciones no funcionen a través de un *router* configurado con alguna variante de este modo operativo. En general, cualquier protocolo del nivel de aplicación, que incluya en la parte de datos información sobre direcciones IP o números de puerto TCP/UDP, supone un reto para NAPT ya que la detección y modificación de dichas informaciones requiere un análisis de dicho nivel de aplicación.

Los *routers* actuales de las organizaciones, incluyendo los *routers* ADSL de los usuarios, permiten que en una misma tabla de traducciones pueden convivir las traducciones efectuadas en el nivel de red y transporte (NAPT) para:

- Clientes internos mediante traducciones NAPT dinámicas unidireccionales
- Servidores internos mediante traducciones NAPT estáticas bidireccionales

Superredes

La experiencia ha demostrado que la división del espacio de direcciones numéricas en clases A, B y C ha resultado ser bastante inflexible e ineficiente en muchos casos. Para evitar terminar con el espacio de direcciones de IP de la clase B y poder hacer un uso más óptimo del espacio de direccionamiento en función del número de máquinas que, en realidad, se desea conectar; se ha creado el concepto de **superred** o **CIDR** (*Classless Internet Domain Routing*: Encaminamiento entre Dominios sin Clase) que se basa en una técnica que permite resumir un conjunto variable de direcciones IP contiguas de red de una clase (en la práctica de la clase C) en una misma dirección de IP de red de esa clase para, por un lado, disponer de un espacio de direccionamiento superior sin necesidad de solicitar una dirección de rango superior (en la práctica de la clase B); y, por otro lado, evitar que las tablas de encaminamiento, y los mensajes para una actualización dinámica de éstas entre *routers* contiguos, crezcan demasiado.

Por todo lo anterior, en 1993 se eliminó en Internet la restricción del espacio de direcciones con clase, adoptándose un esquema o notación en el que se utiliza una longitud de prefijo común arbitraria para indicar la dirección común de red de un bloque de direcciones de red contiguas que se quieren resumir en una sola dirección de red. Este esquema es lo que se conoce como formato CIDR o de superred y que representa una alternativa al direccionamiento IP con clase.

El **formato CIDR** resume un grupo de direcciones contiguas, básicamente, de la clase C en una sola dirección de red de la clase C y, por tanto, en una sola entrada en la tabla de encaminamiento.

(dirección de red/longitud)

Donde:

- Dirección de red: Representa la dirección más baja del bloque o grupo contiguo de direcciones de IP que se quieren resumir en una única dirección de destino o entrada en la tabla de encaminamiento.
- Longitud: Indica el número de bits que delimitan la máscara CIDR, que definen el prefijo común que comparten todas las direcciones de red del bloque de direcciones adyacentes que se desean resumir en una sola dirección.

El **bloque CIDR** es el número de direcciones de red contiguas que conforman el bloque de direcciones que se desea resumir en una sola dirección de red.

Bloque CIDR = máscara por omisión – máscara CIDR

Donde la máscara CIDR es un número de 32 bits que contiene “unos” en los bits que identifican la dirección o prefijo común de red del bloque de direcciones contiguas que se desea resumir.

Si el Bloque CIDR = 0, es porque la máscara por omisión es igual a la máscara CIDR y, por tanto, la máscara está denotando una sola dirección de red.

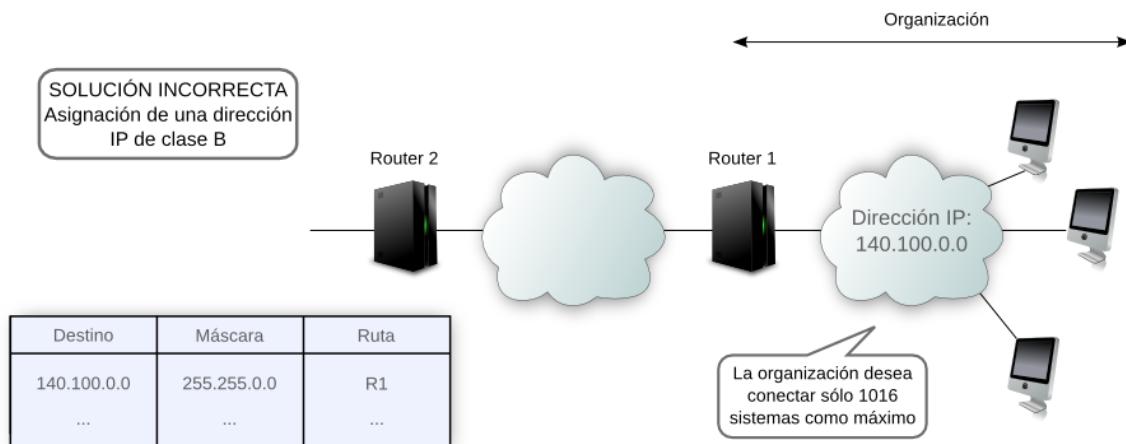
Un ejemplo de formato CIDR es:

$$\text{Formato CIDR} = (220.1.0.0/22) = (220.1.0.0, 255.255.252.0)$$

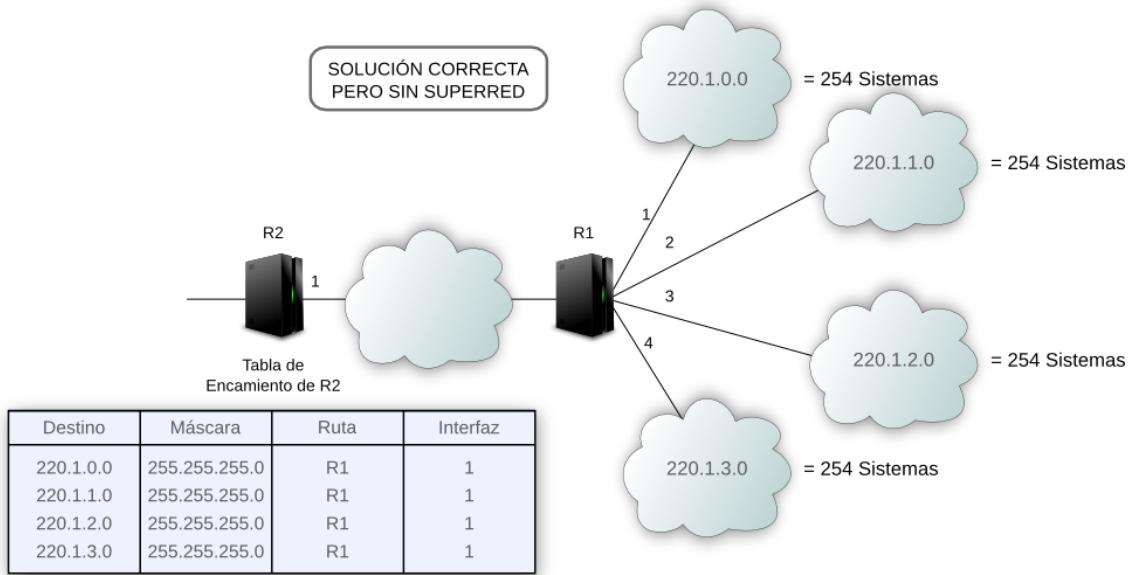
Se utiliza una longitud de prefijo de 22 bits (/22) que son los 22 bits a unos de la máscara de CIDR que definen un prefijo común de 22 bits y que es equivalente en tamaño a cuatro redes de la clase C.

El Bloque CIDR = 24 bits (clase C) – 22 bits = 2 bits = $2^2 = 4$ direcciones de redes contiguas. Asimismo, todas ellas tienen como prefijo común: 11011100.00000001.000000xx.

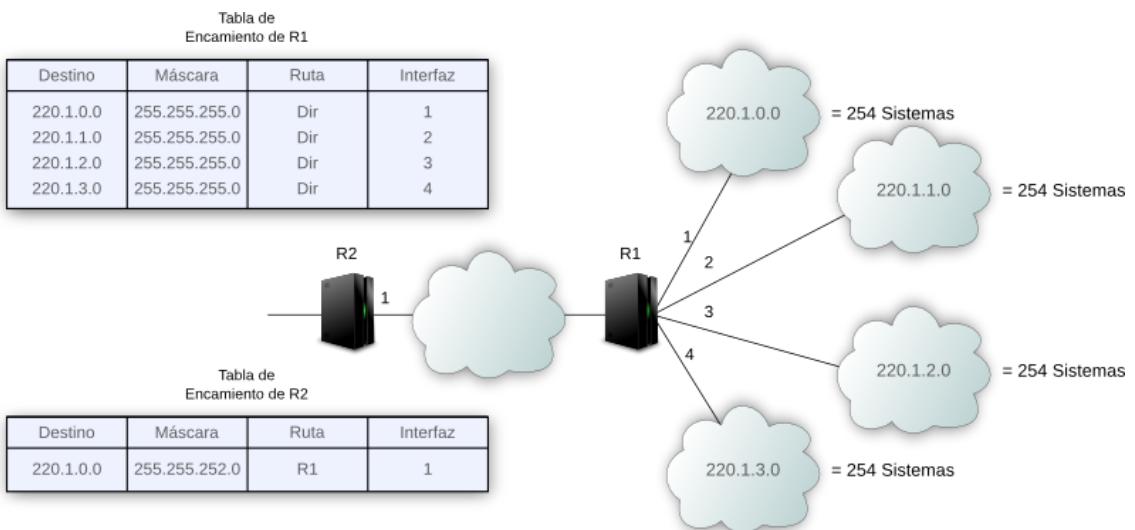
En la siguiente figura se muestra la ineficiencia y falta de flexibilidad en la división del espacio de direcciones de IP en clases A, B y C. Una organización desea conectar 1016 máquinas y se le ha asignado la dirección de IP oficial de la clase B: 140.100.0.0. Con una dirección de red de la clase B se pueden conectar 65.534 máquinas como máximo. Teniendo en cuenta que se desean conectar como máximo 1016 máquinas, sobran pues 64.510 potenciales direcciones, lo cual supone todo un desperdicio.



Utilizando el concepto de superred, se asigna a la organización cuatro direcciones contiguas de red de la clase C ($4 \times 254 = 1016$). De esta forma, no se pierde una dirección de la clase B y el espacio de direccionamiento (mediante redes contiguas de la clase C) está más aprovechado.



Para solventar el problema del aumento de entradas en las tablas de encaminamiento se ha utilizado el formato CIDR que permite resumir un grupo de direcciones contiguas en una sola entrada en la tabla de encaminamiento.



Agotamiento del espacio de direcciones en Internet

La creciente demanda de direcciones numéricas ha supuesto un problema en el modelo con clase. La mayoría de las empresas que solicitan direcciones de red de la clase B han determinado que una dirección de clase B se ajustaría mejor a sus necesidades por el equilibrio entre el número de redes y el número de máquinas que ofrece. En este contexto, una dirección de clase A suele ser excesiva, con más de 16 millones de máquinas, y una de clase C tiene muy pocas máquinas por red.

Distintas soluciones para evitar el agotamiento de direcciones de clase B son:

- Asignación consecutiva de direcciones de red IP de la clase C:
- Menos de 256 direcciones = 1 red de clase C

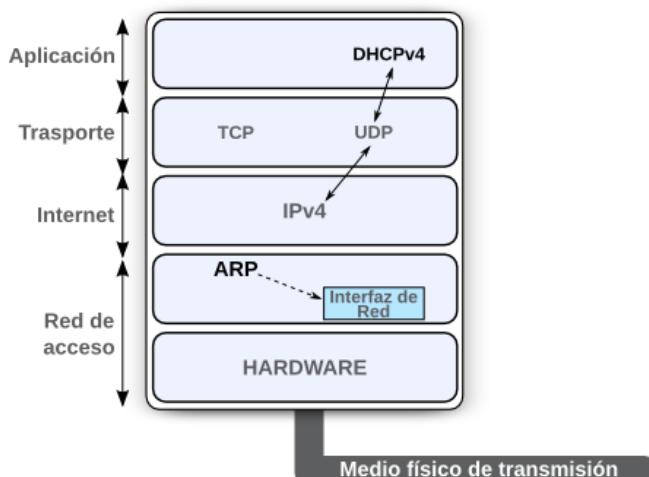
- Menos de 512, pero más de 256 direcciones = 2 redes contiguas de clase C
- Menos de 1.024, pero más de 512 direcciones = 4 redes contiguas de clase C
- Uso de superredes o CIDR: se fundamenta en una combinación de un conjunto de direcciones de IP contiguas de la clase C en una misma dirección de esa clase para disponer de un espacio de direccionamiento superior sin necesidad de solicitar una dirección de rango superior de la clase B para dicha organización.

Soluciones para evitar el agotamiento de cualquier dirección:

- Direccionamiento IP privado y Traducción de direcciones de red (NAPT): permite preservar el espacio de direccionamiento IP oficial, amén de proporcionar privacidad en las direcciones IP e incluso en los números de puerto utilizados.
- Protocolo IP versión 6 (direcciones IP de 16 octetos): conserva y adapta el protocolo IPv4 buscando un mayor espacio de direccionamiento (se pasa de 4 octetos a 16 octetos), una mayor calidad de servicio, una mayor seguridad y una mayor flexibilidad y rapidez en futuros encaminamientos por Internet.

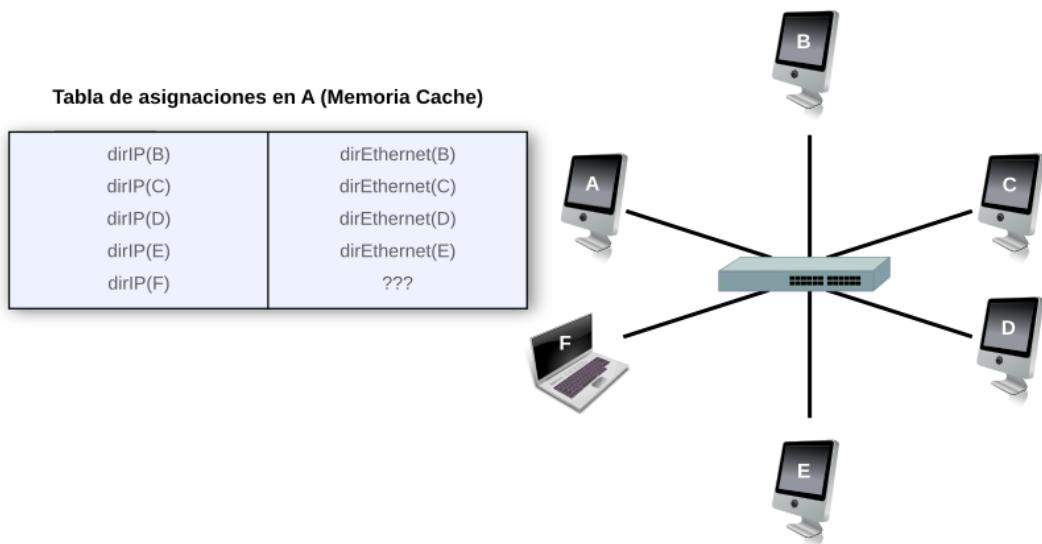
Protocolos relacionados con el direccionamiento IPv4

A continuación se van a tratar los niveles de comunicaciones que ocupan los protocolos de la arquitectura TCP/IP más relacionados con el direccionamiento IPv4.



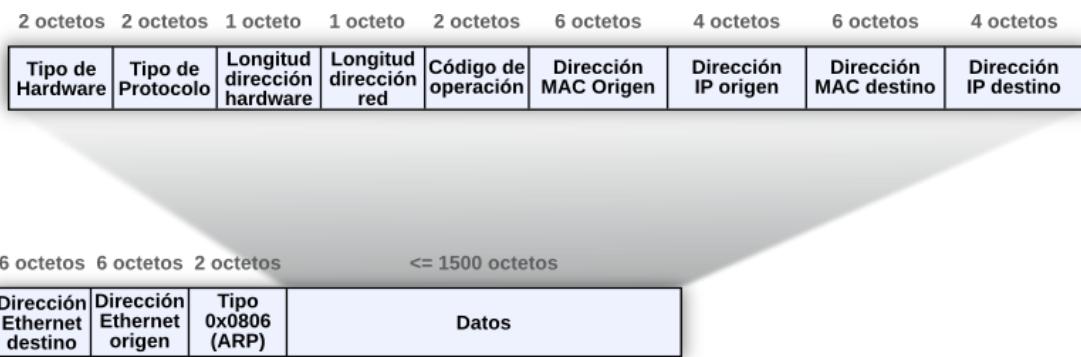
Protocolo ARP

El **protocolo de resolución de direcciones ARP** (*Address Resolution Protocol*) hace corresponder una dirección IP con una dirección MAC o de tarjeta de red. Si una máquina “A” quiere comunicarse con otra “F”, cuya dirección MAC no conoce, su proceso ARP envía a la red una trama de difusión que contiene: si es una solicitud o respuesta ARP, la dirección IP de la máquina “F” y las direcciones IP y MAC de la máquina “A”. A su vez, “F” reconocerá su dirección IP y enviará únicamente a la máquina “A” un mensaje de respuesta ARP encapsulado en una trama específica de información. Esta respuesta ARP transporta las direcciones MAC e IP del solicitante original (“A”) y de la máquina que responde (“F”).



Es importante resaltar que una dirección de IP indica cómo acceder a la red a la cual está conectada una determinada máquina. Pero con una dirección IP no se puede entrar físicamente por el hardware de la máquina en cuestión. Para ello, es necesario conocer la dirección MAC. Por otro lado, ARP utiliza una memoria caché (RAM) en donde existe una **tabla de asignaciones de direcciones IP y MAC**. Esta tabla permite almacenar vía ARP todas las direcciones MAC de máquinas con las que se ha conectado anteriormente. El hecho de que ARP maneje temporalmente esta información, es para evitar tener datos de una estación que falla o es reemplazada o no está operativa o encendida.

El formato de los paquetes ARP de solicitud y respuesta es el siguiente:



Todo paquete ARP se encapsula en el campo datos por ejemplo, de una trama *Ethernet* cuya estructura es la siguiente:

- Cabecera de control (*Ethernet*):
 - Dirección *Ethernet* destino (6 octetos): Dirección MAC de la máquina destino. En un paquete ARP de solicitud, este campo contiene una dirección de difusión con 48 bits “todos a unos” en binario.
 - Dirección *Ethernet* origen (6 octetos): Dirección MAC de la máquina origen.
 - Tipo de trama (2 octetos): Indica el contenido encapsulado en el campo de datos de una trama *Ethernet*. Este campo contiene el valor (hexadecimal) 0x0806 para una solicitud o respuesta ARP.

- Campo Datos (*Ethernet*): Puede tener una longitud variable entre 0 y 1500 octetos y encapsula, a su vez, los siguientes campos de un paquete ARP:
 - **Tipo de hardware** (2 octetos): Especifica el tipo de dirección hardware o MAC. Por ejemplo, el código 1 identifica una dirección *Ethernet*.
 - **Tipo de protocolo** (2 octetos): Especifica el tipo de protocolo del nivel de red utilizado o, más en concreto, el tipo de formato de dirección que se va a traducir en una dirección de hardware. Para direcciones IP su valor es 0x0800 (hexadecimal). Este campo contiene el mismo valor que el campo Tipo de trama cuando la trama *Ethernet* contiene un datagrama IP.
 - **Longitud de la dirección hardware** (1 octeto): Longitud en octetos (6 octetos) de la dirección MAC. También se puede utilizar en otras redes de difusión (por ejemplo, WiFi).
 - **Longitud de la dirección de red** (1 octeto): Longitud en octetos (4 octetos) de la dirección IP.
 - **Código de operación** (2 octetos): Indica si el paquete ARP es una solicitud (código 1) o una respuesta (código 2).
 - **Dirección MAC origen** (6 octetos): Dirección hardware de la máquina origen. Es una información duplicada que también se encuentra en el campo dirección *Ethernet* de origen de la cabecera de la trama *Ethernet*.
 - **Dirección IP origen** (4 octetos): Dirección IP de la máquina origen.
 - **Dirección MAC destino** (6 octetos): Dirección hardware de la máquina destino. Este campo, cuyo contenido es el objetivo del protocolo ARP, sólo se cumple en la correspondiente respuesta ARP.
 - **Dirección IP destino** (4 octetos): Dirección IP de la máquina destino.

Protocolo DHCPv4

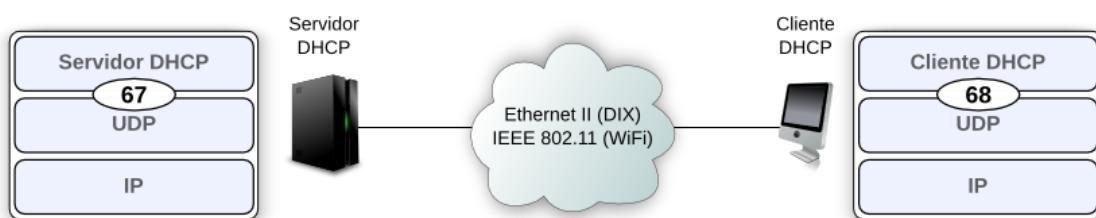
El **protocolo dinámico de configuración TCP/IP DHCPv4** (*Dynamic Host Configuration Protocol version 4*), permite automatizar completamente la asignación de direcciones IP temporales (y permanentes) y toda la información TCP/IP de configuración (dirección IP de la red, dirección IP del siguiente *router*, máscaras asociadas, direcciones IP de los servidores DNS, etc.) para que cualquier sistema pueda crear, inmediatamente, su tabla de encaminamiento IP y acceder a una red TCP/IP o a Internet. Por tanto, DHCP juega un papel fundamental, cuando un sistema portátil se mueve, se conecta o desconecta en una red *Ethernet* o WiFi.

El protocolo DHCP dispone de dos modos de configuración:

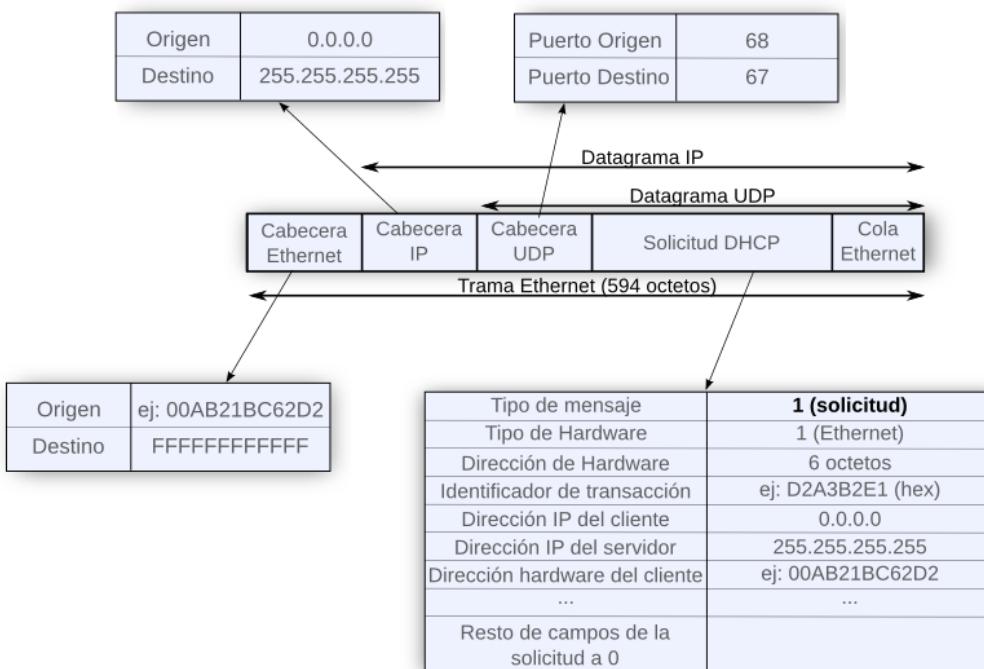
- **Configuración dinámica o temporal:** Automatiza, aparte del resto de la información de configuración TCP/IP, la asignación temporal de direcciones IP durante un tiempo limitado. El administrador, en un fichero de configuración del servidor DHCP, introduce un rango de direcciones IP, especificando su tiempo de uso, para que éste las vaya asignando dinámicamente (a las direcciones MAC recibidas) en función de las solicitudes de los clientes.
- **Configuración fija o permanente:** Automatiza, aparte del resto de la información de configuración TCP/IP, la asignación permanente de direcciones IP. El administrador, en un fichero de configuración del servidor DHCP, introduce las direcciones IP y las asocia de manera estática y de forma permanente a las correspondientes direcciones MAC de

los clientes. Este modo de actuación está pensado para facilitar las labores de un administrador y que éste pueda configurar de una sola vez todos sus sistemas sin necesidad de hacerlo, manualmente, sistema por sistema. Si se producen cambios en la red, en lugar de volver a configurar individualmente todos sus sistemas, el administrador modifica el fichero de configuración DHCP en el servidor para establecer los nuevos cambios. Una vez que se reinicie la red o rearanquen los clientes, se aplicarán los cambios.

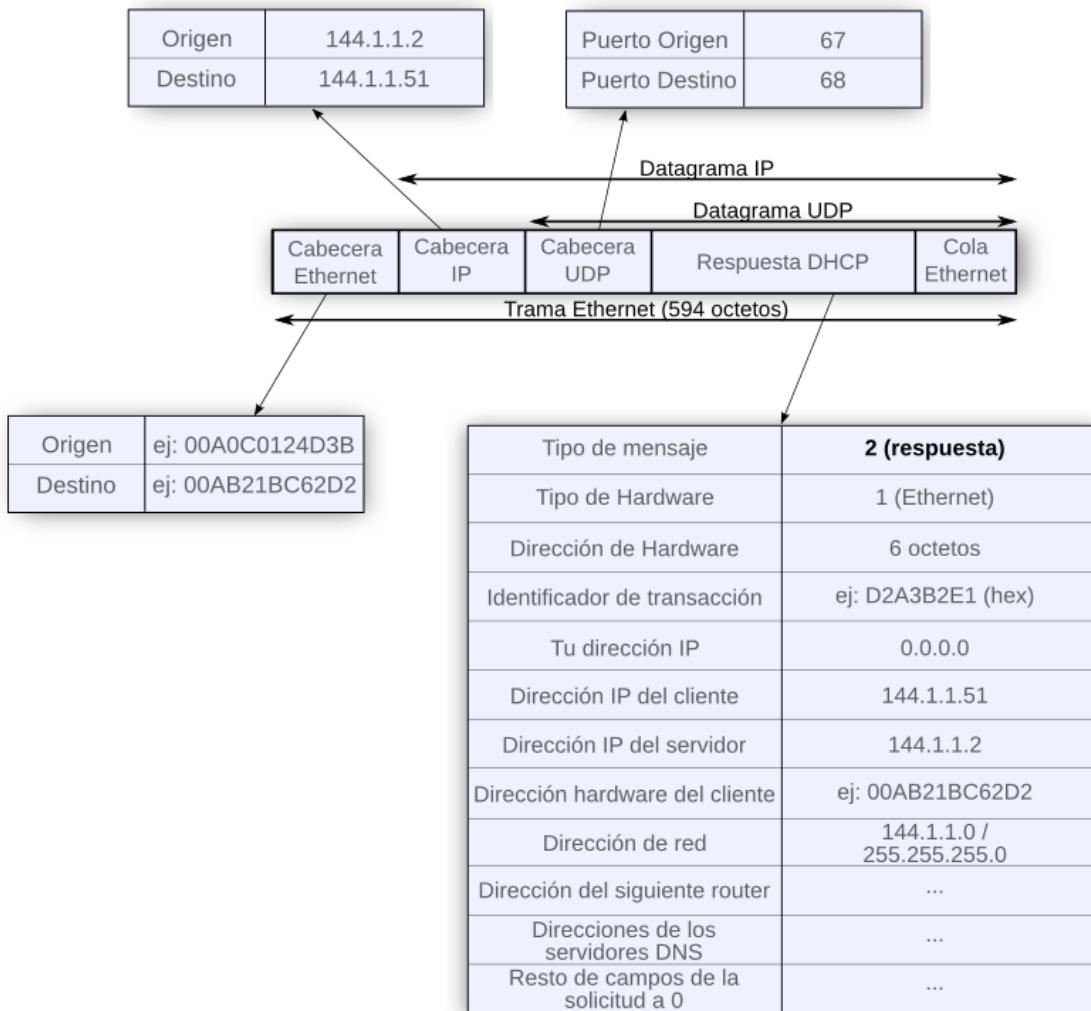
Independientemente, del modo de configuración DHCP, la forma de actuar siempre es la misma: En el sistema cliente se ejecuta un proceso cliente DHCP que vía UDP se comunica con un proceso servidor DHCP. Este proceso servidor DHCP se ejecuta, a su vez, en un sistema centralizado de la organización conectado, habitualmente, a la misma red de acceso que la del cliente. El cliente DHCP se identifica con el número de puerto 68 y el servidor DHCP con el 67.



El sistema cliente se conecta, por primera vez, a una red del tipo *Ethernet* y desea obtener tanto su dirección IP como el resto de la información de configuración TCP/IP. El proceso cliente DHCP envía una solicitud DHCP con 0.0.0.0 como dirección IP de origen y 255.255.255.255 (difusión limitada en el nivel IP) como dirección IP de destino. En caso de conocer la dirección IP del servidor (caso más habitual), puede incluir ésta (144.1.1.2). A su vez, en el nivel de trama *Ethernet* o WiFi, el cliente pone como dirección origen su dirección MAC (00AB21BC62D2) y como dirección MAC destino FF-FF-FF-FF-FF-FF.



Posteriormente, el servidor DHCP asocia una dirección IP libre a la dirección MAC recibida (00AB21BC62D2) y transmite al cliente dicha dirección IP junto con el resto de información de configuración TCP/IP.



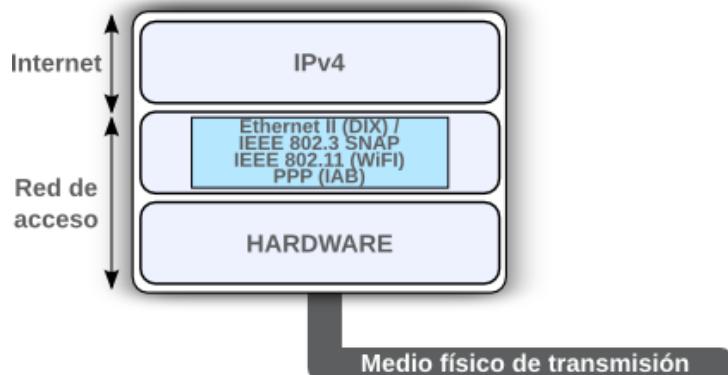
En las solicitudes y respuestas DHCP (594 octetos) aparecen, entre otros, los siguientes campos:

- **Tipo del mensaje:** Identifica si el mensaje es una solicitud (código 1) o una respuesta DHCP (código 2).
- **Tipo de hardware:** Especifica el tipo de dirección de hardware. Un valor 1 es para una red *Ethernet*.
- **Longitud de la dirección de hardware:** Longitud de la dirección de hardware en octetos. Un valor 6 es para una dirección *Ethernet*.
- **Identificador de transacción:** Se utiliza para asociar una solicitud con su correspondiente respuesta. Si el cliente rellena a ceros este campo, significa que va a ignorar dicho campo y no se va a molestar en analizar el valor que lleva la respuesta.

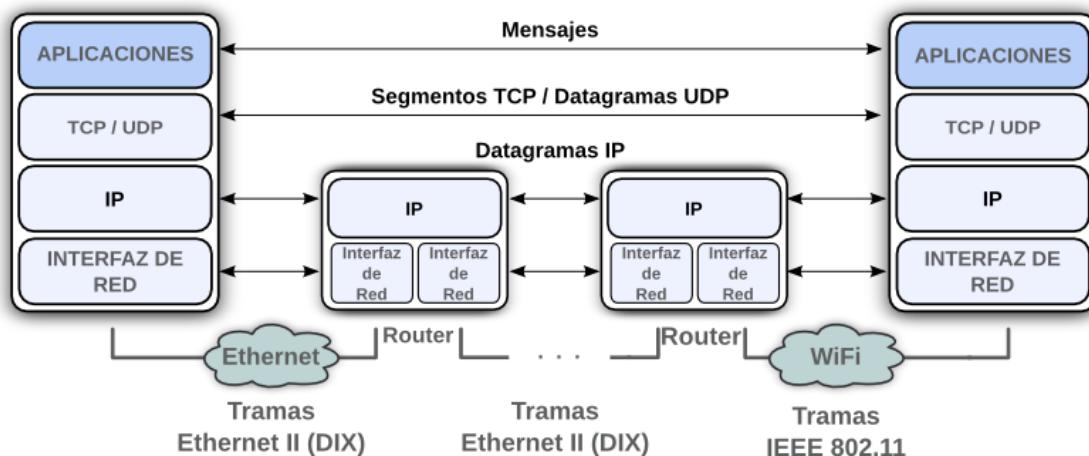
Protocolo IPv4

Fundamentos

El **protocolo IP** (*Internet Protocol*) es el protocolo responsable del encaminamiento de paquetes o datagramas IP entre un sistema origen y sistema destino por Internet.



IP ofrece siempre un servicio no orientado a conexión, es decir, cada paquete se encamina de forma independiente y dichos paquetes no tienen relación entre ellos. Por ser un servicio no orientado a conexión, no se mantiene ningún tipo de control de errores ni de flujo durante el encaminamiento. De ahí, que se diga que el protocolo IP utiliza un servicio de “mejor entrega posible” o “hago lo que puedo” (*best effort*).



Las características fundamentales del protocolo IPv4 son:

- Es un protocolo clave de la arquitectura TCP/IP. El protocolo IPv4 es el protocolo por excelencia del nivel de red; de ahí que también se denomine a este nivel como nivel IP. Su función primordial es encaminar paquetes IP por Internet.
- Ofrece un servicio no orientado a conexión y no fiable. No hay control de errores ni flujo (Sólo detección de errores físicos en la cabecera IP). Por omisión, para IPv4 todos los paquetes son iguales.
- Por omisión, no ofrece prioridades de tratamiento o procesamiento para que un paquete adelante a otro en la cola del *buffer* del correspondiente interfaz salida de un *router*.

- Por omisión, tampoco proporciona calidad de servicio o QoS (*Quality of Service*) en cuanto a caudal, retardos y pérdidas. Sólo prioridades de tratamiento y calidad de servicio en los *routers* de la red IP de un operador si se ha contratado previamente una determinada calidad de servicio con dicho operador.

El tratamiento de los paquetes IP dentro de un *router* depende de su configuración interna y en función de ésta, dispondrá de más o menos funcionalidad. La mayoría de los *routers* en Internet, salvo los *routers* de los operadores con los que se ha contratado previamente el servicio, disponen de una configuración mínima, por omisión, para el funcionamiento de la tradicional cola FIFO (*First-In-First-Out*) del *buffer* asociado a cada interfaz de salida. Es el servicio IP más simple, y por omisión, pero no el ideal.



Se descartan paquetes IP (**congestión de un router**) cuando se desborda la capacidad de almacenamiento de los *buffers* asociados a las líneas de salida, al superar las tasas de entrada las capacidades de salida. La congestión o perdida de paquetes IP en un *router* de acceso, es especialmente crítico en enlaces de entrada de alta capacidad y enlaces de salida de menor capacidad.



Formato de un datagrama IPv4

La estructura de un datagrama o paquete IPv4 es la siguiente:



- **Cabecera de control:** Tiene como mínimo, y por omisión, 20 octetos sin opciones de servicios adicionales.
- **Datos:** Encapsula cabeceras de información de control de niveles superiores y potenciales datos de usuario.

La longitud máxima de un datagrama IP (cabecera y datos) es de 65.535 octetos (64 KBytes).

Versión	Longitud Cabecera	Tipo de Servicio	Longitud Total						
Identificador			0	D	M	F	Desplazamiento		
Tiempo de Vida (TTL)	Protocolo		Suma de comprobación cabecera (XOR)						
Dirección de Origen									
Dirección de Destino									
Opciones	Relleno								
Datos									

Los distintos campos de la cabecera de control de un datagrama IPv4 son:

- **Versión** (4 bits): Indica la versión 4 del protocolo IP.
- **Longitud de la cabecera** (4 bits): Especifica el número de bloques de 4 octetos de que consta la cabecera. Este campo es necesario debido a que la longitud de la cabecera es variable. Como mínimo la longitud es de 20 octetos sin opciones de servicios adicionales (5 bloques de 4 octetos o un 5 en decimal o 0101 en binario). Como máximo una cabecera IP tiene 60 octetos con todas las opciones de servicios adicionales posibles (15 bloques de 4 octetos o un 15 en decimal o 1111 en binario).
- **Tipo de servicio del datagrama**(8 bits): El IETF cambió la interpretación y el nombre de este campo que, ahora, se denomina de servicios diferenciados. A continuación, se muestra el contenido original de este campo siguiendo la terminología tradicional:
 - **Prioridad**: Los tres primeros bits indican la prioridad de procesamiento del datagrama. Existen 8 niveles de prioridad entre 0 (000 en binario) o prioridad normal y 7 (111 en binario) o prioridad más alta. Cuanta más prioridad tenga un datagrama, antes se procesa éste. La prioridad del datagrama se utiliza en situaciones tales como la congestión. Si un *router* se encuentra saturado y necesita descartar algunos datagramas, se eliminarán primero aquéllos con menor precedencia.
 - **D (Delay)**: Bit de mínimo retardo de tránsito del datagrama desde un origen a un destino. Sólo hay dos posibles valores: Normal (0) o bajo (1). Se utiliza cuando se le da la máxima importancia al “tiempo de viaje” de un datagrama desde un sistema origen a un sistema destino.
 - **T (Throughput)**: Bit de máximo rendimiento en el transporte del datagrama desde un origen a un destino. Sólo hay dos posibles valores: Normal (0) o alto (1). Se utiliza cuando interesa transmitir siempre a la máxima velocidad posible, es decir, por los enlaces de mayor capacidad o mayor ancho de banda.

- **R (Reliability):** Bit de máxima fiabilidad en el transporte del datagrama desde un origen a un destino. Sólo hay dos posibles valores: Normal (0) o alta (1). Se utiliza cuando es importante tener alguna certeza de que los datos llegarán al destino sin necesidad de una retransmisión.

La aplicación puede activar, o no, los parámetros de la cabecera de control IP, de cada uno de los datagramas IP. Dicha aplicación pasa los valores asociados en una llamada al proceso IP. La aplicación sólo puede activar un bit de los tres (R, C, F). Seguidamente, se muestran algunas recomendaciones de uso:

Telnet	D = 1, T = 0, R = 0 (demora mínima)
FTP (control)	D = 1, T = 0, R = 0 (demora mínima)
FTP (datos)	D = 0, T = 1, R = 0 (rendimiento máximo)
SMTP	D = 0, T = 1, R = 0 (rendimiento máximo)
DNS	D = 0, T = 0, R = 1 (fiabilidad máxima)
HTTP	D = 0, T = 1, R = 0 (rendimiento máximo)
DHCP	D = 0 , T = 0, R = 0 (servicio normal)
ICMP	D = 0 , T = 0, R = 0 (servicio normal)

De esta manera, si un *router* dispone, de varios enlaces para alcanzar un determinado destino; entonces, elige aquél que más se ajuste al bit activado (D, T, R) de Tipo de servicio para el correcto encaminamiento del datagrama en cuestión.

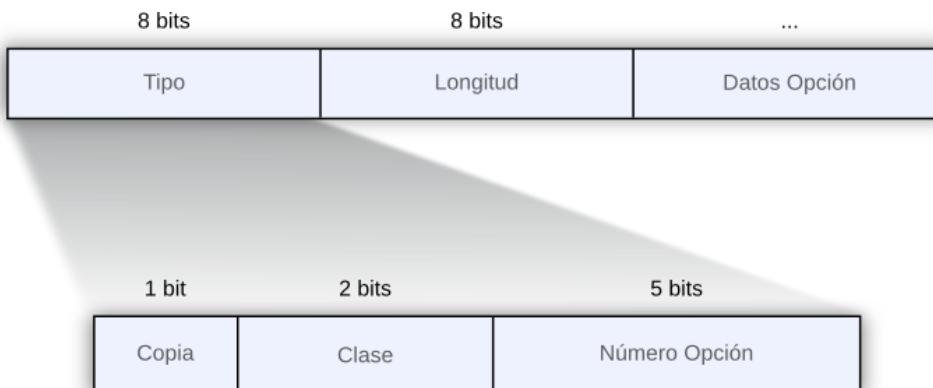
En la práctica, los *routers* en Internet no analizan el campo Tipo de Servicio, salvo los *routers* de un operador (ISP) que hacen uso de sus propios códigos para este campo.

- **Longitud total** (16 bits): Indica la longitud total en octetos de un datagrama (cabecera y datos). Dicha longitud como máximo es de 65.535 octetos. Para encontrar la longitud de los datos que vienen del nivel superior, se resta la longitud de la cabecera de la longitud total. Prácticamente, en la totalidad de los casos no se necesita este campo.
- **Identificador** (16 bits): Es el número asignado, por omisión, a todo datagrama se fragmente o no, e identifica a los potenciales fragmentos pertenecientes a un mismo datagrama.
- **Un bit a cero** (reservado para un uso futuro).
- **DF**: Bit de no fragmentar. Está activado cuando la aplicación no desea que se fragmente su datagrama por el camino ya que sólo el datagrama completo es útil.

- **MF:** Bit de más fragmentos. Si está activado indica que, a continuación, vienen más fragmentos pertenecientes al mismo datagrama. El último fragmento lleva este bit desactivado.
- **Desplazamiento** (13 bits): Indica el número de bloques de 8 octetos contenidos en el campo de datos en fragmentos anteriores.
- **Tiempo de vida (TTL):** Define el número máximo de *routers* (255) que el datagrama puede atravesar en Internet desde un origen a un destino. Cada *router* decrementa en una unidad el valor almacenado en este campo. Si el resultado es cero, se elimina el datagrama. Si el valor es diferente de cero se actualiza el campo con el nuevo valor. La entidad IP del sistema final destinatario no decrementa el contenido del campo TTL porque el paquete ya ha llegado al destino. Este campo es muy útil para evitar viajes en bucle y que el datagrama circule indefinidamente por una mala configuración de las tablas de encaminamiento de los *routers* implicados.
- **Protocolo** (8 bits): Es un número que identifica el protocolo superior (TCP = 6, UDP = 17, OSPF = 89, ICMP = 1, IGMP = 2) al cual IP debe entregar los datos transportados por el datagrama.
- **Suma de comprobación** (16 bits): Suma aritmética binaria o en módulo 2 sin acarreos (suma XOR o OR-exclusivo) de todos los bloques de 16 bits de que consta la cabecera. La suma de comprobación sólo se aplica a la cabecera y no a los datos. Hay dos razones. En primer lugar, todos los protocolos del nivel superior (TCP y UDP) que encapsulan datos en el datagrama IPv4 tienen un campo con la suma de comprobación que cubre todo el paquete. En segundo lugar, la cabecera del paquete IPv4 cambia al pasar por cada *router*, pero no los datos. Por tanto, la suma de comprobación incluye sólo la parte que cambia.
- **Dirección de origen** (32 bits): Los cuatro octetos que identifican al sistema origen del datagrama. El contenido de este campo no se modifica nunca en el trayecto del origen al destino.
- **Dirección de destino** (32 bits): Los cuatro octetos que identifican al sistema destino del datagrama. El contenido de este campo no se modifica nunca en el trayecto del origen al destino.
- **Opciones:** Es un campo de información de control de longitud variable para servicios adicionales. Puede haber 0, 1 o más opciones.
- **Relleno:** Son los bits que se añaden al campo de opciones para conseguir que la cabecera tenga una longitud total múltiplo de 4 octetos.

Opciones de servicio

La cabecera de un datagrama IPv4 consta de dos partes: una parte fija y otra variable. La parte fija tiene una longitud de 20 octetos. La parte variable comprende las opciones que pueden ocupar un máximo de 40 octetos en función de los siguientes campos:



- **Tipo:** Indica el tipo de la opción en función, a su vez, de los siguientes tres campos.
 - **Copia** (1 bit): Indica si el campo de opción se debe copiar en todos los fragmentos (bit activado) o sólo en el primero (bit desactivado).
 - **Clase** (2 bits): Indica la clase general de la opción. Básicamente, hay dos opciones:
 - Depuración y medición (gestión).
 - Control de red: Es la clase por excelencia y agrupa a todos los números de opciones que se van a ver a continuación.
 - **Número de opción** (5 bits): Entre las opciones más significativas se destacan las siguientes:
 - Encaminamiento desde origen: Especificación de las direcciones IP de los *routers* por donde debe pasar el datagrama.
 - Registro de ruta: Permite al sistema origen crear una lista vacía y con suficiente espacio (en el campo de Datos Opción) para que cada *router* que procese ese datagrama incluya su propia dirección IP.
 - Sello o marca de tiempo: Permite al sistema origen crear una lista vacía y con suficiente espacio (en el campo de Datos Opción) para que cada *router* que procese ese datagrama incluya su propia dirección IP y una marca de tiempo de 32 bits que indique el momento en que procesó el datagrama.
- **Longitud:** Es un campo de 8 bits que indica la longitud total en octetos de la opción.
- **Datos de la opción:** Información para poder procesar debidamente la opción.

Rutina de comprobación de la cabecera

Toda entidad IP, incluyendo las entidades intermedias en los *routers* y la entidad IP en la máquina destino, llevan a cabo una rutina de comprobación de la cabecera IP en función de los siguientes campos:

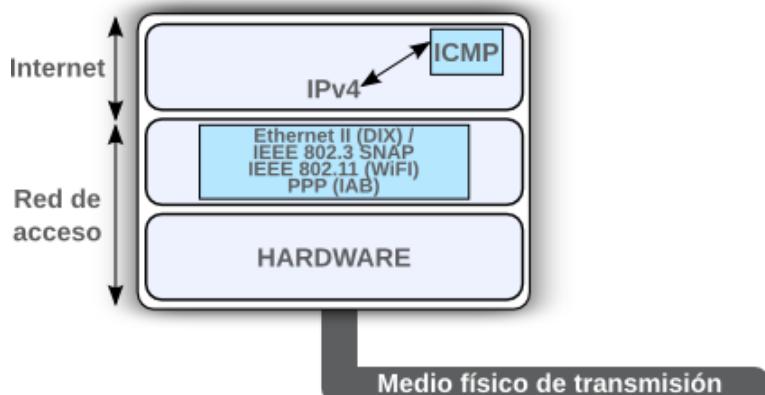
1. Suma de comprobación.
2. Versión.
3. Longitud de la cabecera.
4. Longitud total. Si alguno de los campos anteriores no es válido, el datagrama se desecha sin enviar ninguna notificación al origen.
5. TTL-1: Sólo las entidades IP en los *routers* (nunca la entidad IP en la máquina destino) decrementan dicho campo en una unidad y comprueban que el resultado no sea cero.

Si el resultado es diferente de cero entonces actualiza el campo TTL y se calcula y actualiza la suma de comprobación). Si por el contrario el resultado es cero, entonces elimina el paquete.

Posteriormente se analizan las opciones, aunque no suele ser habitual debido a que los *routers* se optimizan para encaminar datagramas sin opciones. Si todo lo anterior se lleva a cabo correctamente y no hay que fragmentar como se verá a continuación, la entidad IP procede a encaminar el datagrama aplicando las máscaras y la información de su tabla de encaminamiento.

Protocolo ICMPv4

El **protocolo ICMPv4** (*Internet Control Message Protocol*) es el protocolo de envío de mensajes de control en Internet (RFC-792) y está tan íntimamente ligado al protocolo IP, que de hecho se puede ver como un módulo más dentro del propio módulo IP.



El protocolo IP no tiene mecanismos de informe o corrección de errores. ¿Qué ocurre si se presenta un problema con un datagrama? Por ejemplo, ¿qué sucede si un *router* descarta un datagrama porque no tiene información en su tabla IP para encaminarlo hacia el destino?, ¿qué ocurre si un *router* obtiene un valor TTL igual a cero?, ¿qué sucede si vence el temporizador de reensamblado y el sistema destino debe eliminar los fragmentos que le han llegado?, ¿qué ocurre si una entidad IP no entiende algún parámetro de la cabecera de información de control? Éstas son situaciones en las que ha ocurrido un error y en donde el protocolo IP no tiene mecanismos para notificarlo al sistema origen. Resumiendo, se ha diseñado el protocolo ICMPv4 para solventar el envío de mensajes relacionados con las situaciones anteriormente descritas. Los mensajes de control ICMP están relacionados con informes de errores y consultas.

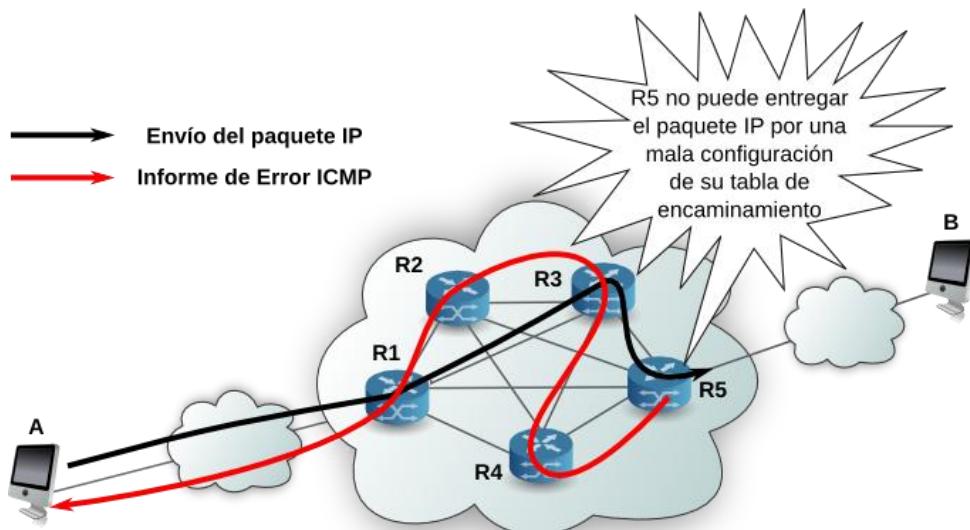
Se resalta que el destino de un mensaje ICMP es siempre el sistema origen que ha enviado el correspondiente datagrama. Se recuerda que un mensaje ICMP se utiliza para enviar mensajes de control y no para hacer más fiable al protocolo IP, el cual jamás podrá recuperar un datagrama.

Los tipos de mensajes ICMPv4 que existen son:

- **Informes de error:** Problemas que un *router*, o la máquina destino (o incluso la máquina origen), pueden encontrar al procesar un datagrama IP

- Destino inalcanzable (falta de información en la tabla IP)
- Tiempo excedido (TTL = 0 en un *router* o tiempo de reensamblado excedido en la máquina destino)
- Frenado en el origen (evitar una congestión en un *router* de Internet)
- Problemas con los parámetros (información ininteligible en la cabecera del datagrama IP)
- Redirección (actualización de la tabla IP de la máquina origen)
- **Consultas:** Información que permite que una máquina tenga información de otra
 - Solicitud y respuesta de eco (comprobación de si una máquina está conectada y responde)

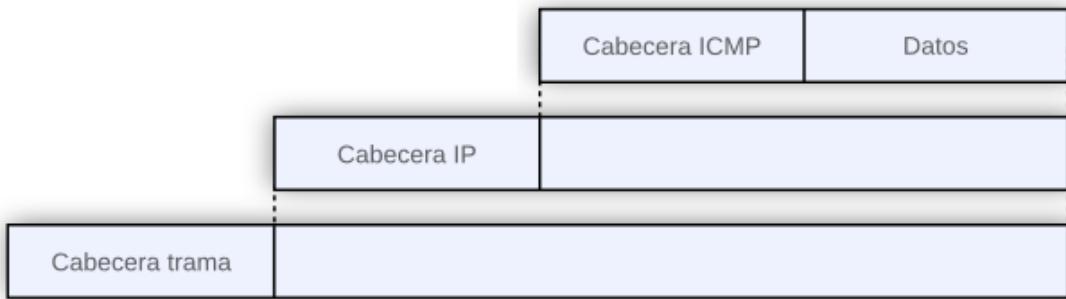
La siguiente figura muestra un ejemplo del envío de un datagrama IP desde el sistema “A” al sistema “B”. El *router* R3 no dispone de suficiente información en su tabla IP y es incapaz de encaminar dicho datagrama al destino “B” en cuestión. Consecuentemente, R3 elimina el datagrama y envía un mensaje ICMP al sistema origen “A”, notificando dicho suceso. El datagrama que encapsula dicho mensaje, incluso, no tiene porqué ir por el mismo itinerario de *routers* que el datagrama inicial.



Como el sistema origen a la cual va destinado un mensaje ICMP puede ser un sistema remoto por Internet, los mensajes ICMP se encapsulan siempre en datagramas IP. De ahí que ICMP ocupe un subnivel superior al ocupado por el protocolo IP en el mismo nivel Internet o de red de la arquitectura TCP/IP.

Formato y mensajes ICMPv4

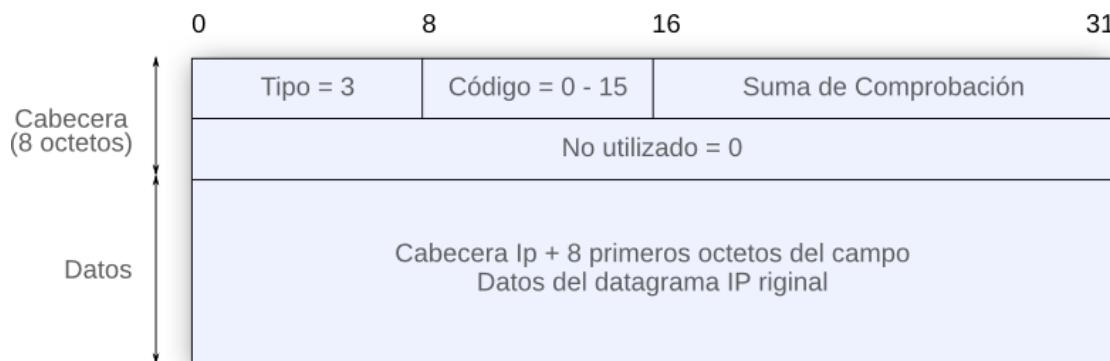
Todo mensaje ICMPv4 dispone de un formato en función de tres campos fundamentales:



- **Tipo** (1 octeto).
- **Código** (1 octeto): Una información adicional y específica sobre el tipo del mensaje ICMP.
- **Suma de comprobación** (2 octetos): Se aplica a todo el mensaje ICMP. Al igual que en IP sólo hay detección de errores físicos y en caso de error, eliminación del mensaje sin recuperación del mismo.
- **Parámetros opcionales** (4 octetos): Información opcional que unos mensajes ICMP utilizan y otros no (en este último caso los 4 octetos están "a cero").
- A los campos anteriores sigue un contenido variable dependiendo del tipo y código.

Seguidamente, se describe el formato concreto de los principales mensajes ICMPv4:

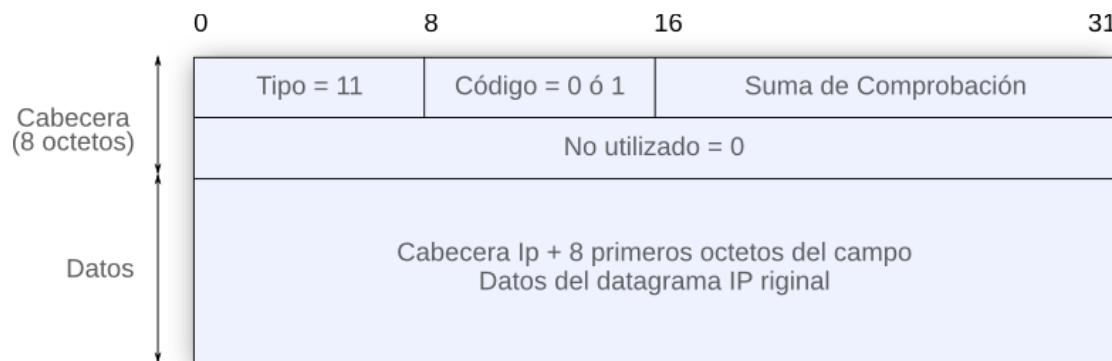
El formato del **mensaje de destino inalcanzable** se define por un Tipo = 3 y Códigos del 0 al 15 que indican más explícitamente la categoría de dicho mensaje. Asimismo, para una mayor información en la máquina de origen, se copia la cabecera IP y los 8 primeros octetos del campo de datos del datagrama original. Este mensaje lo envía tanto un *router* o la máquina origen (básicamente, cuando no sabe o no puede reenviar un datagrama a una red) como la máquina de destino (cuando el protocolo o puerto especificado no está activo).



- Código 0 - Red no alcanzable: Lo generan los *routers* o la máquina origen cuando hay un error en la dirección IP de destino o no disponen de suficiente información en la tabla de encaminamiento para encaminar el pertinente datagrama.
- Código 1 - Máquina destinataria no alcanzable (sin respuesta ARP): Sólo lo genera el *router* final (conectado a la misma red de acceso que la máquina destino) cuando no obtiene respuesta ARP.

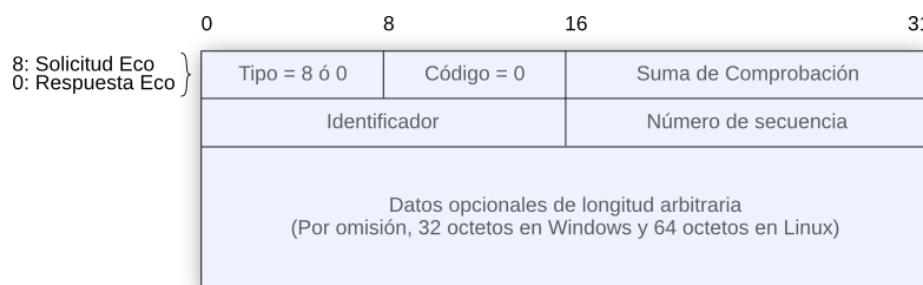
- Código 2 - Protocolo no alcanzable: Lo genera el nivel de red de la máquina destinataria cuando el protocolo superior (TCP, UDP, etc.) indicado en la cabecera de IP (campo de Protocolo) no está disponible.
- Código 3 - Puerto no alcanzable: Se genera a partir del nivel de transporte de la máquina destino cuando el puerto destino no se corresponde con algún proceso en uso.
- Código 4 - Fragmentación necesaria y no realizada: Lo genera un *router* cuando es necesario fragmentar (el tamaño del datagrama IP es superior a la MTU de la red) y el bit DF está activado en la cabecera de información de control IP.
- Código 5 - Fallo en el encaminamiento desde origen: Lo genera un *router* cuando es imposible transitar al *router* especificado en la pertinente opción IP (encaminamiento desde origen).

El formato del **mensaje de tiempo excedido** se define por un Tipo = 11 y Códigos del 0 al 1 que indican más explícitamente la categoría de dicho mensaje. Asimismo, para una mayor información en la máquina de origen, se copia la cabecera IP y los 8 primeros octetos del campo de datos del datagrama original. Dicho mensaje lo envía tanto un router (cuando se ha excedido el TTL del datagrama) como una máquina destino (cuando se ha excedido el tiempo de reensamblado de un datagrama fragmentado).



- Código 0 - Tiempo de Vida (TTL) del Datagrama Excedido.
- Código 1 - Tiempo de Reensamblado Excedido.

El formato de los **mensajes de solicitud y respuesta de eco** se definen por un Tipo = 8 ó 0 (solicitud o respuesta) y Código = 0. Estos mensajes los utiliza el comando ping para comprobar que una máquina está activa o en servicio en Internet o en una red privada TCP/IP. Esto último ocurre cuando se envía una solicitud y se recibe el eco (copia) de lo transmitido.

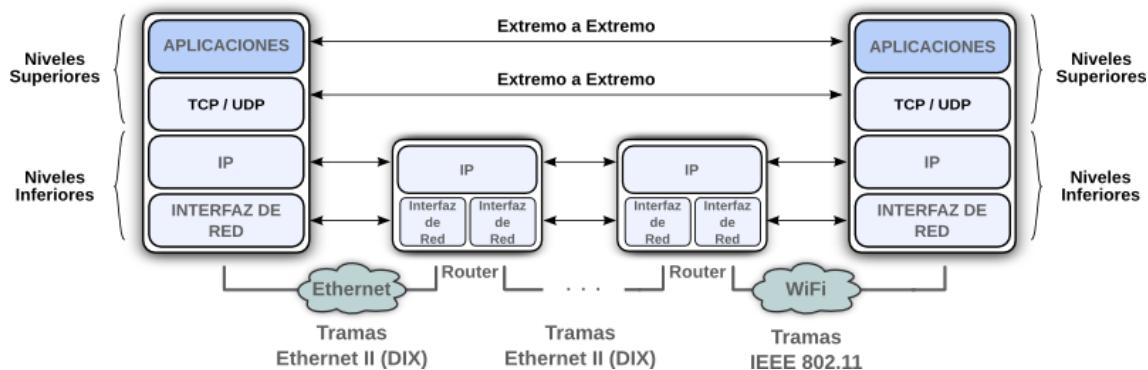


Parte de la cabecera contiene un Identificador (16 bits) y un Número de secuencia (16 bits) para asociar solicitudes con respuestas y secuenciar más de una solicitud y respuesta con el citado comando ping. A su vez, el campo de datos del mensaje contiene unos octetos arbitrarios que introduce la propia implementación y cuya longitud puede especificar el usuario. Por tanto, se pueden mandar tamaños diferentes de datos en cada solicitud de eco.

Nivel de transporte de datos

Introducción y generalidades

El nivel de enlace es responsable de la entrega de tramas entre dos sistemas vecinos en un mismo enlace (comunicación sistema a sistema en un enlace). A su vez, el nivel de red o nivel de Internet es responsable de la entrega de datagramas IP por Internet entre dos sistemas ya sean vecinos o no (comunicación sistema a sistema en Internet). Sin embargo, la comunicación real tiene lugar entre dos procesos de una aplicación (nivel de aplicación). Para ello, es necesario de un nivel de transporte que permita la comunicación entre procesos de aplicación.



Protocolo TCP

Fundamentos

El **protocolo TCP**, ofrece extremo a extremo, fundamentalmente, un servicio orientado a conexión y, por tanto, fiable y con control de flujo, independientemente de las redes de comunicaciones y *routers* que hayan intervenido entre dos sistemas finales. Las unidades de datos del protocolo TCP se denominan **segmentos TCP**.

El protocolo de control de la transmisión TCP proporciona, al correspondiente proceso de aplicación, un servicio de flujo de octetos (*byte-stream*), orientado a la conexión y fiable, multiplexado y dúplex. Estos cuatro servicios fundamentales de TCP se desglosan a continuación:

- **Flujo de octetos (*byte-stream*)**: Este servicio permite al proceso de aplicación (emisor) transmitir un flujo continuo de octetos a su entidad TCP emisora para que ésta los vaya recogiendo, numerando y agrupando en unidades de datos denominadas segmentos. La entidad TCP emisora calcula un MSS (*Maximum Segment Size*) de tal forma que los datagramas IP se correspondan con la MTU de la red de acceso. La entidad TCP receptora pasa al proceso de aplicación (receptor), exactamente, la misma secuencia de octetos y en el mismo orden (sin pérdidas ni duplicaciones).

- **Orientado a conexión**(conexiones lógicas): Este servicio permite, mediante la conexión entre la correspondiente pareja de sockets (cliente y servidor), que la entidad TCP emisora se ponga de acuerdo con la entidad TCP receptora para llevar a cabo todas las funciones de control de errores (fiabilidad) en la fase posterior de transferencia de datos.
 - **Control de errores:** Este servicio TCP abarca tanto el control de los errores lógicos (octetos de datos perdidos, desordenados y duplicados) como físicos (bits cambiados). Los errores lógicos se controlan mediante el uso de:
 - Temporizadores: El campo de datos de cada segmento tiene asociado un único temporizador y a su vencimiento (*timeout*), es decir, si el temporizador expira antes de que se confirmen todos los octetos de datos del segmento, se retransmite el segmento.
 - Números de secuencia: Se asignan a cada octeto transmitido (y nunca a los segmentos que transportan dichos octetos) un número de secuencia.
 - Confirmaciones (*Acknowledgments* o ACKs): Permiten a la entidad TCP emisora desactivar los temporizadores asociados a los octetos de datos transmitidos y, finalmente, cuando todos los octetos de datos de la ventana de transmisión se hayan confirmado, girar ésta para seguir enviando datos. El campo de datos de cada segmento de información tiene asociado una confirmación al igual que un temporizador.

Los errores físicos se controlan mediante dos procesos:

- Detección: Mediante un mecanismo de suma de comprobación que se aplica a todo el segmento.
- Corrección: A través de retransmisiones al vencimiento de los correspondientes temporizadores. La entidad TCP receptora elimina el segmento afectado y no envía una solicitud de retransmisión.
- Control de flujo: Este servicio impide que una entidad TCP emisora “A” transmita más rápidamente de lo que otra entidad TCP receptora “B” es capaz de almacenar y procesar. Tanto “A” como “B” utilizan un protocolo de ventana deslizante para el control del flujo de segmentos de información.
- **Multiplexación:** Este servicio permite al protocolo TCP ofrecer un servicio multiplexado, o en paralelo, a los diferentes procesos de aplicación a través de los números de puerto que identifican a dichos procesos.
- **Full-dúplex:** Este servicio permite que la transmisión de información entre las dos entidades TCP sea bidireccional y simultánea.

Control de flujo

La ventana de recepción (W_R) controla la numeración de los octetos de datos almacenados en el *buffer* de recepción. Esta ventana define una lista de números de secuencia consecutivos que se corresponden con los octetos que en un momento dado el receptor puede aceptar. Cualquier octeto cuyo número de secuencia esté fuera del rango esperado de números de secuencia, es automáticamente rechazado.

A parte de su propia ventana de recepción, toda entidad TCP dispone, también, de una ventana de transmisión (W_T) para controlar la numeración de los octetos de datos almacenados en el *buffer* de transmisión. Esta ventana define una lista de números de secuencia consecutivos que se corresponden con los octetos que en un momento dado el emisor ha enviado sin haber recibido confirmación. El *buffer* de transmisión es una memoria temporal en donde se guarda copia de los octetos contenidos en el campo de datos de cada segmento de información transmitido. Mientras la aplicación envía de forma continua los datos que quiere transmitir, la entidad TCP emisora recoge, numera y almacena estos datos en su *buffer* de transmisión. Seguidamente, la entidad TCP emisora agrupa los octetos de datos almacenados en trozos y les añade una cabecera creando los segmentos que pasa a la entidad IP del nivel inmediatamente inferior. Si no se recibe una confirmación, al vencimiento del temporizador se retransmite la copia de los datos del correspondiente segmento.

Aunque puede variar de una implementación TCP/IP a otra, generalmente, para que W_R y W_T estén debidamente sincronizados y se lleve a cabo un correcto control de flujo, se tiene que cumplir lo siguiente:

- Inicialmente y en fase de establecimiento de la conexión, W_R tiene que coincidir con el tamaño máximo del propio *buffer* de recepción. Posteriormente, en fase de transferencia de datos, W_R va variando puntualmente hasta el tamaño máximo en función de los octetos libres de su *buffer* de recepción. Se resalta que el tamaño del *buffer* de recepción no tiene porqué ser igual del *buffer* de transmisión. Para ello TCP dispone de un mecanismo adaptable de asignación de tamaño para su *buffer* de transmisión en función de la ventana de recepción indicada por el otro extremo en la fase de establecimiento de la conexión.
- Se pasan datos al proceso de aplicación cuando se llena el *buffer* de recepción. Salvo que el bit PSH (*Push*) de la cabecera TCP esté activado.
- El límite inferior de W_R se corresponde con el primer octeto de datos que se espera recibir después del último octeto pasado al proceso de aplicación.
- Se gira W_R (límites inferior y superior) cuando se pasan datos al proceso de aplicación. Se recuerda que esto último ocurre cuando se llena el *buffer* de recepción.
- Inicialmente y en fase de establecimiento de la conexión, W_T tiene que ser menor o igual que la W_R inicial del otro extremo y, por tanto, tiene que ser menor o igual que el tamaño máximo del *buffer* de recepción del otro lado de la comunicación. Posteriormente y en fase de transferencia de datos, W_T va variando puntualmente en función de la W_R indicada por el otro extremo.
- Se gira W_T (límites inferior y superior), y se eliminan las correspondientes copias en el *buffer* de transmisión, cuando se recibe confirmación de todos los octetos comprendidos entre el límite inferior y superior de W_T .
- El límite inferior de W_T debe coincidir con el límite inferior de W_R .
- El límite superior de W_T debe coincidir con el límite superior de W_R .
- Si el límite superior de W_T es mayor que el límite superior de W_R , entonces los octetos fuera de W_R serán descartados. Por tanto, no se puede girar el límite inferior de W_T antes de tiempo porque también se giraría su límite superior.

Formato de un segmento TCP

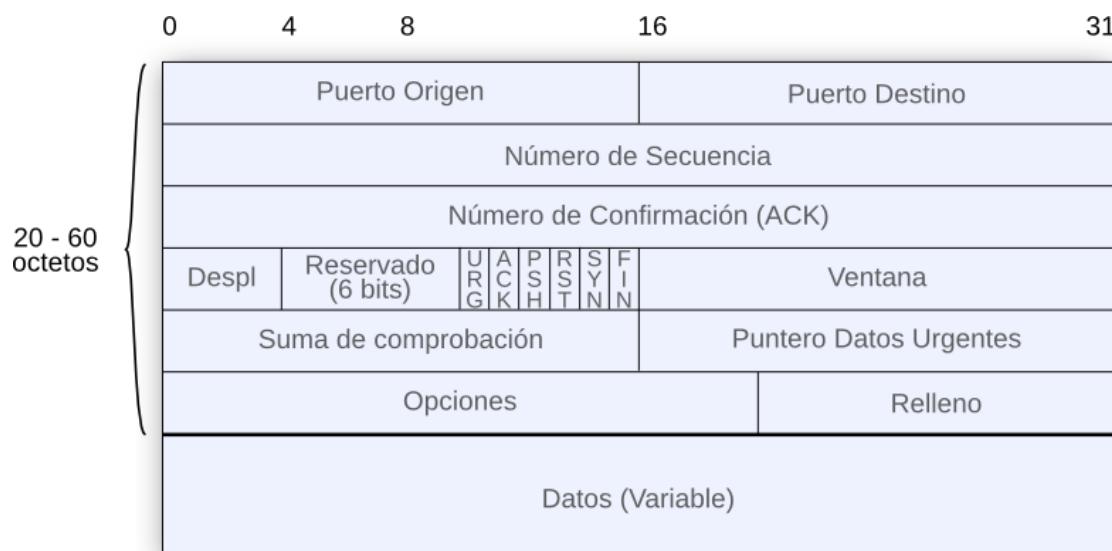
Un **segmento TCP** engloba dos tipos de información:

20 octetos sin opciones



- **Cabecera de control:** Contiene la información de control que manejan las entidades TCP para llevar a cabo sus respectivas funciones. Tiene una longitud mínima, por omisión, de 20 octetos (sin opciones de servicios adicionales) y máxima de 60 octetos (incluyendo opciones).
- **Datos:** Incluye la cabecera de información de control del correspondiente proceso de aplicación y los potenciales datos del mensaje de dicha aplicación (si existen). Este campo de datos se corresponde con lo que se entiende como Tamaño Máximo del Segmento (MSS: *Maximum Segment Size*). La entidad TCP emisora calcula un MSS de longitud variable de tal forma que los datagramas IP se correspondan con la MTU de la red de acceso. Para usar los trozos del tamaño adecuado es necesario que el protocolo TCP disponga en su diseño de un temporizador interno que le permita recoger una cantidad razonable de octetos de datos antes de crear el segmento de información. El proceso de aplicación envía, de forma continua, a su entidad TCP emisora los datos que quiere transmitir para que dicha entidad TCP recoja, numere y almacene estos datos en su *buffer* de envío y, finalmente, los agrupe en segmentos. Posteriormente, la citada entidad TCP emisora, una vez se ha llenado su *buffer* de transmisión, toma un trozo de los datos almacenados y le añade una cabecera, creando un segmento. Finalmente, dicha entidad TCP emisora pasa el segmento a la entidad IP para que lo entregue como un único datagrama IP.

Los distintos campos que conforman la cabecera de información de control de TCP son los siguientes:



- **Puerto de Origen** (16 bits): Identifica al proceso de aplicación emisor que provoca el envío de dicho segmento TCP.
- **Puerto de Destino** (16 bits): Identifica al proceso de aplicación receptor que recibe un segmento TCP.
- **Número de Secuencia** (32 bits): Indica el primer octeto del campo de datos (si tiene un octeto o más) del segmento de información que se va enviar. Como el número de secuencia es de 32 bits, se trabaja en módulo 2^{32} .
- **Número de Confirmación o ACK** (32 bits): Indica el primer octeto del campo de datos del siguiente segmento de información que se espera recibir.
- **Desplazamiento de Datos (Data Offset) o Longitud de la Cabecera** (4 bits): Indica el número de bloques de cuatro octetos que ocupa la cabecera. Por omisión (sin opciones de servicios adicionales) tiene una longitud de 20 octetos (5 bloques o “0101” de 4 octetos). El tamaño máximo será de 60 octetos (15 bloques o “1111” de 4 octetos). Al igual que para el protocolo IP, éste es un campo necesario para reconocer el inicio del campo de datos de usuario, ya que el campo de opciones de servicios adicionales TCP es de longitud variable.
- **Reservado** (6 bits): Seis bits a cero reservados para un uso futuro.
- **URG (Urgent Pointer)** (1 bit): Bit urgente que si está activo indica que el campo Puntero de Datos Urgentes es un campo relevante que la entidad TCP receptora debe analizar debidamente.
- **ACK (ACKnowledgment)** (1 bit): Bit de confirmación que si está activo indica que el campo Número de Confirmación o ACK es un campo relevante que la entidad TCP receptora debe analizar debidamente.
- **PSH (Push)** (1 bit): Bit de empuje que define un mecanismo para proporcionar un servicio forzado de transferencia de octetos de datos, obligando a que la entidad TCP emisora transmita sin esperar a que se llene su *buffer* de transmisión. A su vez, esto evita que la entidad TCP receptora almacene temporalmente el segmento hasta llenar, a su vez, su *buffer* de recepción.
- **RST (Reset)** (1 bit): Bit de reinicio que si está activo indica a la entidad TCP receptora que abandone la comunicación debido a un error o a una situación anormal. Se usa como respuesta para rechazar una solicitud de establecimiento de una conexión TCP y para un cierre abrupto de ésta en cualquier momento posterior a la fase de establecimiento de la conexión.
- **SYN(Synchronize)** (1 bit): Bit de sincronización que si está activo indica que se está estableciendo una conexión TCP. Asimismo, sincroniza los números de secuencia empleados por las entidades TCP extremo a extremo. Combinándose con el bit ACK proporciona dos segmentos específicos de control TCP (sin datos de usuario).
 - SYN = 1, ACK = 0: Solicitud de establecimiento de una conexión TCP.
 - SYN = 1, ACK = 1: Respuesta afirmativa (¡OK!) a la solicitud previa de establecimiento de una conexión TCP.
- **FIN (Finalize)** (1 bit): Bit de finalización o liberación de la transmisión. La conexión se libera completamente cuando se transmite en cada sentido un segmento sin datos con dicho bit activado. Consecuentemente, un lado de la conexión se puede liberar y el otro seguir activo transmitiendo datos.

- **Ventana** (16 bits): Es la ventana deslizante de recepción (W_R) de la entidad TCP receptora. Su función es controlar el flujo de segmentos de información que le transmite la otra entidad TCP emisora. Se basa en un sistema de créditos de transmisión de octetos de datos, pendientes de confirmación, que se concede al otro extremo para que pueda transmitir sin colapsar al receptor. Por tanto, dicho sistema de créditos indica el número de octetos, pendientes de confirmación, que se conceden al otro extremo a partir del primer octeto de datos indicado por el campo Número de Confirmación. Se recuerda que la confirmación individual de un segmento de información se asocia a todos los octetos de datos (carga útil) de dicho segmento. La ventana W_R puede ser de 1 octeto, 2 octetos hasta $2^{16}-1$ octetos como máximo (65.535 octetos o 16 “unos” en binario). Quiere esto decir, que el *buffer* de recepción es como máximo de 65.535 octetos. También, es válido un tamaño de 0 para indicar que se han recibido los octetos hasta el NÚMERO DE CONFIRMACIÓN (ACK) - 1 pero que el receptor necesita un tiempo extra de reposo y que de momento no desea recibir más datos. El permiso para enviar se transmite, posteriormente, con un segmento con el mismo NÚMERO DE CONFIRMACIÓN (ACK) y un campo VENTANA distinto de cero. En el establecimiento de la conexión cada entidad TCP define el tamaño máximo de la ventana que desea. Posteriormente, durante la fase de transferencia de datos, este tamaño puede ir variando (se introducen nuevos valores en el campo de Ventana) hasta el máximo definido previamente en la anterior fase de establecimiento de la conexión.
- **Suma de Comprobación** (16 bits): Suma aritmética binaria o en módulo 2 (sin acarreo o suma OR exclusiva) de todos los bloques de 16 bits del segmento completo (cabecera y datos). El procedimiento de cálculo de la suma de comprobación es similar al utilizado para calcular la suma de comprobación de IP salvo por los dos siguientes puntos:
 - Cuando la longitud del segmento no es un múltiplo de 16 bits, el segmento se rellena de ceros hasta hacerlo múltiplo de 16 bits. Sin embargo, el segmento real que se ha de enviar no se modifica por lo anterior.
 - Se incluye una pseudocabecera al inicio del segmento cuando se calcula la suma de comprobación. Esta pseudocabecera, que tampoco se transmite, se crea en el origen y destino durante el proceso de cálculo. Dicho procedimiento le permite al módulo TCP, por un lado, identificar inmediatamente la conexión a la cual pertenece el segmento y, por otro lado, asegurarse de que el segmento ha alcanzado realmente la máquina de destino y el pertinente puerto.
- **Puntero de Datos Urgentes** (16 bits): Cuando el bit URG está activo, el valor de este campo de 16 bits denominado Puntero Urgente sumado al valor del campo Número de Secuencia, apunta al último octeto de los datos urgentes de control. Estos datos urgentes de control requieren un procesamiento especial por parte de la aplicación. Se asume que estos datos urgentes pueden ser avisos, interrupciones o datos de control (por ejemplo, caracteres de control o secuencias de escape) de la propia aplicación que pueden ir entremezclados en el *buffer* de recepción con los datos normales de ésta y se desean que aparezcan antes que dichos datos normales.

- **Opciones** (variable): Concebido para futuras mejoras o extensiones del protocolo TCP y que contiene opciones de servicios extras de dicho protocolo.
- **Datos** (variable): El protocolo TCP calcula un MSS de tal forma que los datagramas IP resultantes se correspondan con la MTU de la red de acceso.

Todas las opciones se especifican en la fase de establecimiento de la conexión (sólo en aquellos segmentos con el bit SYN = 1).

8 bits	8 bits	Variable
Tipo	Longitud	Datos Opción

Dichas opciones, que se van incorporando poco a poco en las diferentes implementaciones TCP, son las siguientes:

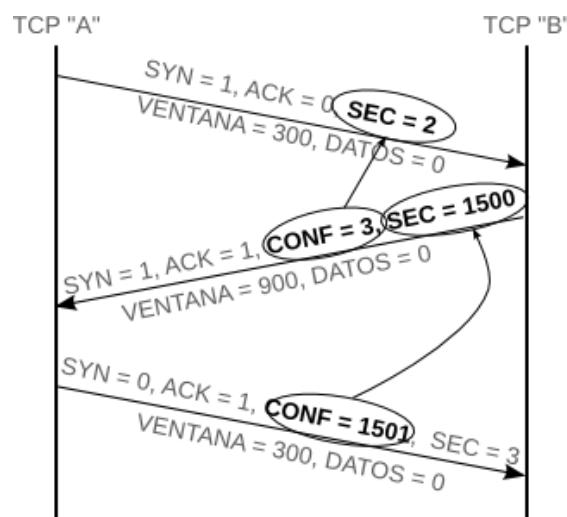
- Opción 0 - Fin de la lista de Opciones: Indica el final de la lista de opciones incluidas en el segmento. Es un único octeto (0000 0000) que sólo se incluye cuando el final de las opciones no coincide con el final de la cabecera TCP.
- Opción 1 - Sin operación - Espacio de relleno: Es un único octeto (0000 0001) que puede incluirse entre las opciones para que la longitud de la opción correspondiente sea un múltiplo de 4 octetos.
- Opción 2 - Tamaño Máximo de Segmento (MSS: *Maximum Segment Size*).: Indica el número de octetos del campo de datos del segmento que la entidad TCP emisora del segmento que transporta esta opción, desea recibir para un procesamiento más óptimo. Esta es una de las opciones más relevantes y, por tanto, más utilizadas con el objetivo de permitir la escalabilidad de los protocolos TCP/IP. A través de esta opción, el protocolo TCP permite manejar diferentes tamaños de *buffers* y segmentos de información.
- Opción 3 -
- Factor de Escala de Ventana: Permite el uso de un tamaño mayor de Ventana (de recepción). Por omisión, el tamaño máximo de Ventana es de $2^{16}-1$ octetos (65.535 octetos). Se puede llegar hasta $2^{32}-1$ (21.073.725.440 octetos o 1 GB aproximadamente).
- Opción 4 - Aviso de Confirmación Selectiva Permitida (SACK PERMITTED: *Selective Acknowledgment Permitted*): Indica a la otra entidad TCP, en la fase de establecimiento de la conexión TCP (en un segmento SYN = 1), que puede usar la opción SACK en la fase de transferencia de datos. Si la entidad TCP receptora no ha recibido esta opción en el establecimiento de la conexión, no debe usar la opción Tipo 5 - SACK en la posterior fase de transferencia de datos.
- Opción 5 - Confirmación Selectiva (SACK: *Selective Acknowledgment*): Informa al emisor de bloques no contiguos de datos que han sido recibidos correctamente y, por tanto, qué octetos de datos se han perdido o no han llegado todavía a la entidad TCP receptora.
- Opción 8 - Marca o sello de Tiempo (*Timestamp*): Permite al emisor calcular el valor RTT (*Round Trip Time*: Tiempo de ida y vuelta de un segmento) para configurar sus

temporizadores. Se resalta que durante el establecimiento de la conexión, el emisor puede enviar esta opción en el primer segmento con SYN = 1, ACK = 0 y puede hacer uso de dicha opción en otros segmentos siempre y cuando haya recibido respuesta con SYN = 1, ACK = 1.

Establecimiento de la conexión

El establecimiento de una conexión entre dos entidades TCP denominadas “A” y “B” se realiza mediante un procedimiento inicial de saludo o diálogo basado en el intercambio de tres segmentos de control (sin datos o Datos=0). Durante esta fase, se informa de los números de secuencia iniciales empleados y del tamaño máximo de la ventana de recepción.

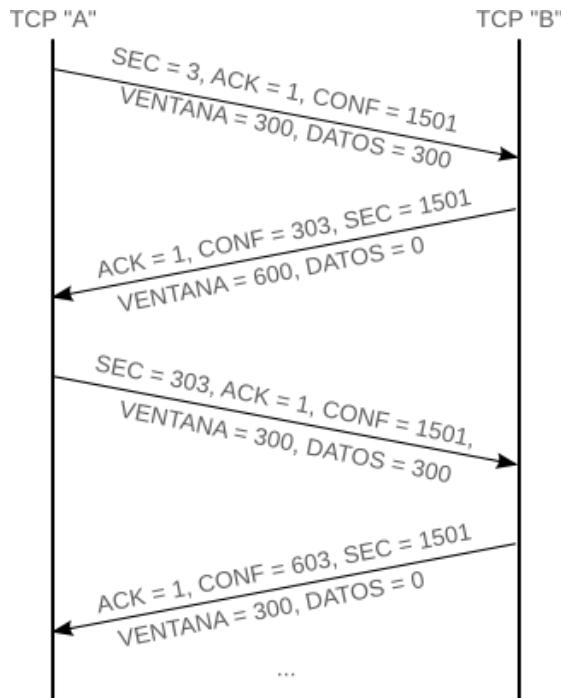
En la siguiente figura se muestra un ejemplo de establecimiento de conexión. Sólo se indican los campos más significativos que aparecen en las correspondientes cabeceras de información de control.



- Primer segmento (SYN = 1, ACK = 0, SEC = 2,...): Se usa para que la entidad TCP “A” se ponga de acuerdo con la entidad TCP “B” con el objetivo de llevar a cabo todas las funciones de control de errores y flujo en la fase posterior de transferencia de datos. Además, indica el número de secuencia (SEC = 2) que desea usar la entidad TCP “A” y el tamaño actual de su ventana (300 octetos).
- Segundo segmento (SYN = 1, ACK = 1, CONF = 3, SEC = 1500,...): Se usa para confirmar (CONF = 3) el número de secuencia “2” del otro extremo de la comunicación (“A”), y para indicar el número de secuencia (SEC = 1500) que va a emplear la entidad TCP “B”. Por tanto, el primer octeto de datos, enviado por “A”, se numerará con el valor 3. Asimismo, la entidad TCP “B” indica el tamaño de su ventana (900 octetos).
- Tercer segmento (SYN =0, ACK = 1, CONF = 1501,...): Se usa para confirmar (CONF = 1501) el número de secuencia “1500” que va a utilizar el otro extremo de la comunicación. Por tanto, el primer octeto de datos, enviado por “B”, se numerará con el valor 1501. Asimismo, la entidad TCP “A” recuerda el tamaño de su ventana (300 octetos).

Transferencia de datos

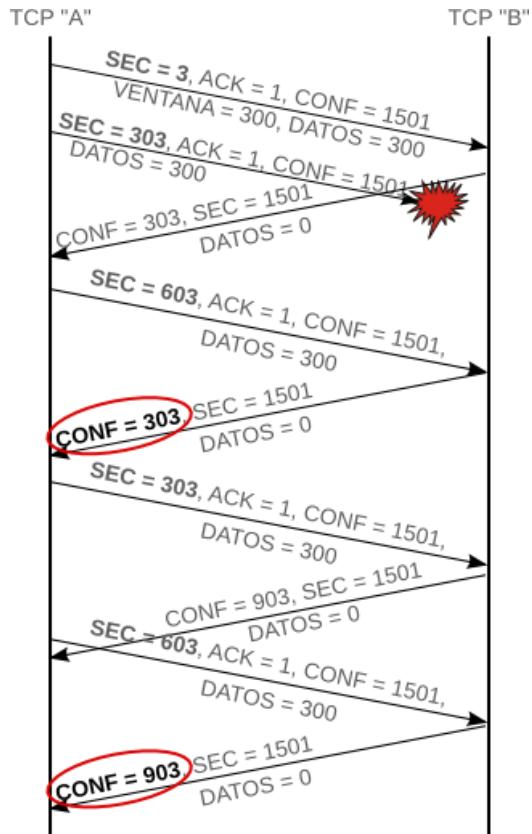
En la siguiente figura se muestra una transmisión unidireccional de segmentos de información siempre de la entidad TCP “A” a la entidad TCP “B” en modo semidúplex una vez se ha llevado a cabo el establecimiento de la conexión. Se supone que en dicha transmisión no se producen errores y se dispone de una ventana máxima de recepción en “A” de 300 octetos y en “B” de 900 octetos.



- Primer segmento de “A” a “B” (SEC = 3, ACK = 1,...): La entidad TCP “A” transmite un primer segmento de información contenido en el campo de datos (DATOS = 300). El primer octeto de dicho campo de datos está identificado con el número 3 (SEC = 3). Asimismo, confirma (CONF = 1501) el número de secuencia “1501” que va a utilizar “B” y recuerda el tamaño de su ventana (300 octetos).
- Segundo segmento de “B” a “A” (ACK = 1, CONF = 303,...): La entidad TCP “B” envía un segmento sin datos (DATOS = 0) y confirma todos los octetos recibidos (CONF = 303). Asimismo, “B” recuerda el número de secuencia (SEC = 1501) que va a utilizar y el tamaño actual de su ventana (600 octetos).
- Tercer segmento de “A” a “B” (SEC = 303, ACK = 1,...): La entidad TCP “A” transmite un tercer segmento de información contenido en el campo de datos (DATOS = 300). El primer octeto de dicho campo de datos está identificado con el número 303 (SEC = 303). Asimismo confirmar (CONF = 1501) el número de secuencia “1501” que va a utilizar “B” y recuerda el tamaño actual de su ventana (300 octetos).
- Cuarto segmento de “B” a “A” (ACK = 1, CONF = 603,...): La entidad TCP “B” envía un segmento sin datos (DATOS = 0) y confirma todos los octetos recibidos (CONF = 603). Asimismo, “B” recuerda el número de secuencia (SEC = 1501) que va a utilizar y el tamaño actual de su ventana (300 octetos).

En la siguiente figura se muestra, en fase de transferencia de datos, una transmisión unidireccional de datos (siempre de la entidad TCP “A” a la entidad TCP “B”). Asimismo, se

asume que en dicha transmisión se producen errores, disponiéndose de una ventana máxima de recepción en "A" de 300 octetos y en "B" de 900 octetos.



En este escenario, la entidad TCP "A" transmite 900 octetos (agrupados en tres segmentos) pendientes de confirmación. Así, el primer octeto está identificado con el número 3 (SEC = 3). Se asume que el segundo segmento se pierde por el camino. La entidad TCP "B" ante la llegada de los 300 primeros octetos procede a confirmarlos (CONF=303). La llegada a la entidad TCP "A" de la primera confirmación (CONF=303) produce una desactivación del temporizador. Cuando llega a la entidad TCP "B" el tercer segmento de información, vuelve a transmitir la anterior confirmación (CONF=303). Sólo al vencer los temporizadores (por no recibir las confirmaciones correspondientes) de los dos últimos segmentos de información enviados por "A", éstos se transmiten de nuevo.

Posteriormente, cuando "B" recibe el segundo segmento retransmitido por "A", procede a confirmar los octetos de datos de dicho segmento (CONF=903). El mismo procedimiento ocurre (CONF=903) ante la llegada otra vez a "B" del tercer segmento, el cual todavía estaba sin confirmar.

Gestión de temporizadores

Con respecto a los valores que tienen que tener los temporizadores de espera de respuesta, no es lo mismo que el emisor y el receptor estén en una misma red de área local o dispersos geográficamente por Internet. La elección de un valor adecuado tiene una consecuencia directa en el funcionamiento eficiente de TCP:

- Si el temporizador es demasiado alto, el emisor esperará innecesariamente en ciertos casos por confirmaciones que nunca llegarán

- Si el temporizador es demasiado bajo se producirán retransmisiones innecesarias de segmentos que habían sido correctamente recibidos

La dificultad de manejar los temporizadores radica en que hay que determinar “aproximadamente” el periodo de tiempo existente desde que se envía un segmento hasta que se recibe su ACK, es decir, el tiempo de ida y vuelta (RTT: *Round Trip Time*). Dicho RTT se puede calcular haciendo uso de marcas de tiempo mediante la opción número 8 de TCP.

En este contexto, el protocolo TCP se enfrenta a las siguientes dificultades:

- Diferencias en capacidad y retardo de unas conexiones a otras.
- Oscilaciones debidas a la presencia de *routers* y situaciones de congestión que están fuera de su control.
- Aún en situaciones de control, los *routers* pueden tener largas colas de datagramas IP que atender, los enlaces pueden ser de diferentes velocidades y la ruta puede variar durante la conexión.

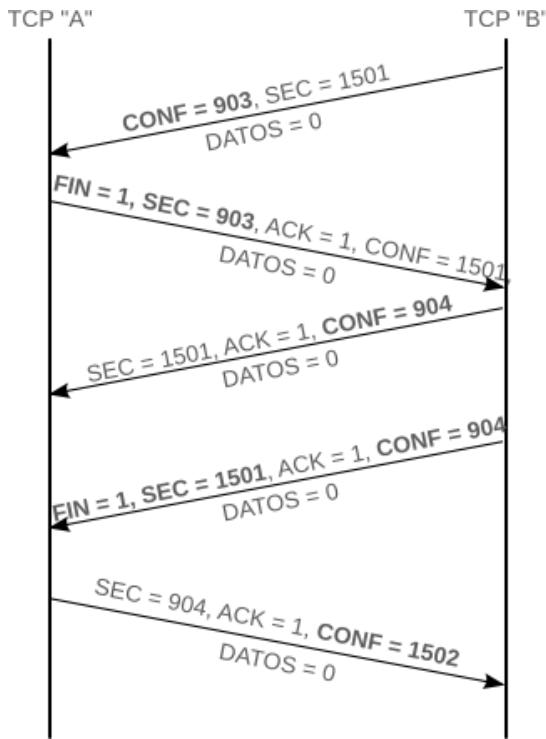
Los valores del temporizador se establecen mediante algoritmos autoadaptativos que dinámicamente ajustan los valores al estado de la red, según es percibido éste por la entidad de transporte emisora. Toda implementación TCP/IP debe disponer de un algoritmo adaptable de retransmisión muy dinámico (algoritmo de Karn) para calcular el RTT e ir adaptando los correspondientes temporizadores en función de los diferentes destinos. Básicamente, el proceso consiste en ir midiendo los RTTs desde que se envía un segmento de información hasta que se recibe su confirmación. Finalmente, el temporizador se obtiene mediante un promedio de las muestras de cada segmento transmitido más un determinado margen de seguridad.

Liberación de una conexión

Una vez terminada la fase de transferencia de datos, se procede a liberar de forma ordenada la conexión previamente establecida. Para liberar cada lado de la conexión, es necesario haber recibido previamente una confirmación de todos los octetos enviados. Esta liberación ordenada implica un cierre independiente en cada dirección de la conexión mediante un proceso de diálogo similar al del establecimiento. Cualquiera de las partes puede solicitar la liberación de la conexión. Conceptualmente, este proceso suele ser el siguiente:

- TCP “A”: “He terminado” (FIN). “No tengo más datos que transmitir”.
- TCP “B”: “OK”.
- TCP “B”: “Yo también he terminado” (FIN). “No tengo más datos que transmitir”.
- TCP “A”: “OK”.

En la siguiente figura se muestra una finalización de una conexión. La conexión finaliza completamente cuando se transmite en cada sentido un segmento sin datos con el bit FIN activado.



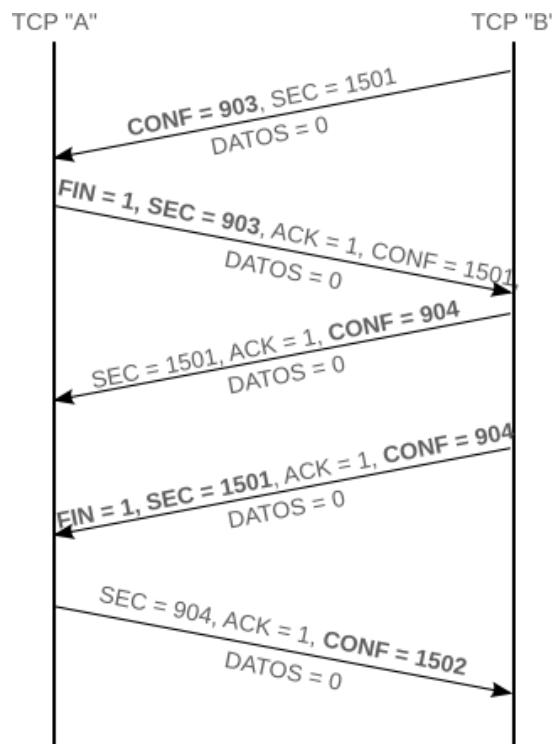
- Primer segmento de “A” a “B” (FIN = 1, SEC = 903, ...): La entidad TCP “A” transmite un segmento sin datos con el bit FIN activado.
- Segundo segmento de “B” a “A” (ACK = 1, SEC = 1501, CONF = 904, ...): La entidad TCP “B” confirma (CONF = 904), con un segmento sin datos, la solicitud de liberación propuesta por el otro extremo “A” de la comunicación.
- Tercer segmento de “B” a “A” (FIN = 1, SEC = 1501, ...): La entidad TCP “B” transmite un segmento sin datos con el bit FIN activado.
- Cuarto segmento de “A” a “B” (ACK = 1, CONF = n+1, ...): La entidad TCP “A” confirma (CONF = 1502) la solicitud de liberación propuesta por el otro extremo “B” de la comunicación.

Las dos partes pueden iniciar la liberación simultáneamente. En este caso, dicha liberación de la conexión se completa cuando una de las partes ha enviado una confirmación (ACK = 1, ...). Como no hay confirmaciones de confirmaciones, al transmitirse la última confirmación, se lanza un temporizador con un tiempo de estimación de la llegada de dicha confirmación al destino en función de los temporizadores de segmentos de información utilizados en la fase de transferencia de datos. Con este tiempo, la entidad TCP que trasmite la última confirmación (ACK = 1) arranca un temporizador con un valor inicial dos veces el tiempo máximo de vida del segmento (*MSL: Maximum Segment Lifetime*). Se entiende por MSL, el tiempo máximo que puede permanecer un segmento en la red antes de que sea descartado. El primer MSL tiene en cuenta el tiempo máximo que puede permanecer en la red un segmento en una dirección. El segundo MSL tiene en cuenta, a su vez, el tiempo máximo que puede permanecer en la red una respuesta en la otra dirección. Al vencer el citado plazo de espera se libera oficialmente la conexión. Con este temporizador se da tiempo a que llegue la confirmación (ACK = 1, ...) a su destino y a recibir potenciales segmentos retrasados u obsoletos para su eliminación inmediata. Asimismo, se asume que durante este tiempo no se deben usar unos mismos números de secuencia como medida de protección contra segmentos retrasados u obsoletos.

Cualquiera de las partes puede invocar una liberación abrupta, es decir, finalizar en cualquier momento tanto en la fase de transferencia de datos como en la fase de liberación. Este tipo de terminación se produce cuando una de las partes ha detectado un problema irrecuperable. Dicha liberación se lleva a cabo mediante el envío de uno o más segmentos de control con el bit RST (RESET) activado. Esto causa que TCP descarte cualquier dato almacenado en el *buffer* de transmisión listo para su salida.

Finalmente, se puede producir una liberación de la conexión TCP en donde una de las partes (“B”) transmite datos después de que la otra (“A”) haya enviado un segmento con el bit FIN activado. Por tanto, el lado de la conexión de “A” a “B” se cierra. Sin embargo, la conexión de “B” a “A” continúa abierta ya que “B” desea transmitir octetos de datos a la entidad “A”.

En la siguiente figura, la entidad TCP “A” envía un primer segmento sin datos con el bit FIN activado, SEC=903 y CONF=1501. Seguidamente, la entidad TCP “B” transmite un segmento con CONF=904. Posteriormente, “B” transmite un segmento de 300 octetos de datos, los cuales son confirmados por “A”. Una vez la entidad TCP “B” termina de transmitir toda su información, procede a enviar un segmento sin datos con el bit FIN activado, SEC=1801 y CONF=904. Finalmente, la entidad TCP “A” transmite un último segmento con CONF=1802, completándose la liberación de la conexión TCP.



Protocolo UDP

Fundamentos

El protocolo UDP (User Datagram Protocol) (RFC-768, STD 0006) ofrece, extremo a extremo, un servicio no orientado a conexión y, por tanto, no fiable y sin control de congestiones en el nivel de transporte. El protocolo UDP se utiliza en los siguientes escenarios:

- Aplicaciones interactivas (audioconferencias, videoconferencias, VoIP, etc.) y no interactivas (streaming de audio y vídeo, etc.) en tiempo real
- El intercambio de mensajes es muy escaso y los mensajes son cortos, por ejemplo, consultas al DNS
- Los mensajes se producen regularmente y no importa si se pierde alguno: SNMP, NTP,...
- Los mensajes se envían en una RAL del tipo *Ethernet* (sin errores físicos): DHCP, SNMP,...
- Para tráfico broadcast/multicast

El protocolo de datagramas de usuario UDP (*User Datagram Protocol*) es un protocolo muy simple del nivel de transporte que ofrece, a las correspondientes entidades del nivel de aplicación, un servicio no orientado a conexión y no fiable. Es un protocolo que proporciona dos servicios que no ofrece el protocolo IP:

- **Detección opcional, y sin recuperación, de errores físicos:** Se comprueba opcionalmente la integridad del datagrama UDP completo, es decir, cabecera y datos. Se recuerda que el protocolo IP sólo verifica la integridad de su cabecera de control. Para efectuar este servicio, el protocolo UDP utiliza un mecanismo de suma de comprobación similar al empleado por el protocolo IP.
- **Multiplexación (origen) y demultiplexación (destino) en función de los números de puerto.** Estas funciones principales, del protocolo UDP, proporcionan un servicio añadido al protocolo IP, el cual es capaz de entregar datagramas IP a una máquina determinada en Internet; pero incapaz de hacer lo propio con respecto a los procesos de usuario del nivel de aplicación. Para llevar a cabo dichas funciones, UDP utiliza un mecanismo basado en números de puerto.

Asimismo, y al contrario que con el protocolo TCP, la entidad del nivel de aplicación debe pasar los datos bien delimitados, es decir no ofrece un servicio de flujo de octetos(*byte-stream*).

Teniendo en cuenta lo poco que ofrece el protocolo UDP, ¿qué motivos ha habido para haber diseñado este protocolo en el nivel de transporte? La respuesta es que gracias a este protocolo se evita toda la sobrecarga añadida de enviar y recibir las múltiples unidades de datos necesarias para establecer y liberar una conexión. Muchas aplicacionesse basan en un simple mecanismo de solicitud y respuesta con el intercambio de muy poca información. Asimismo, este es el protocolo ideal para aplicaciones de difusión y multidifusión al eliminar el trabajo extra de retransmitir. Consecuentemente, dichas aplicaciones no necesitan montarse sobre el protocolo TCP y sí sobre otro tipo de protocolo de transporte que ofrezca una mayor rapidez en la entrega de los mensajes. Dado que los protocolos IP y UDP proporcionan servicios no orientados a conexión no fiables; en caso de que la aplicación requiera fiabilidad, y transporte rápido UDP, será necesario implementar la fiabilidad en el nivel de aplicación.

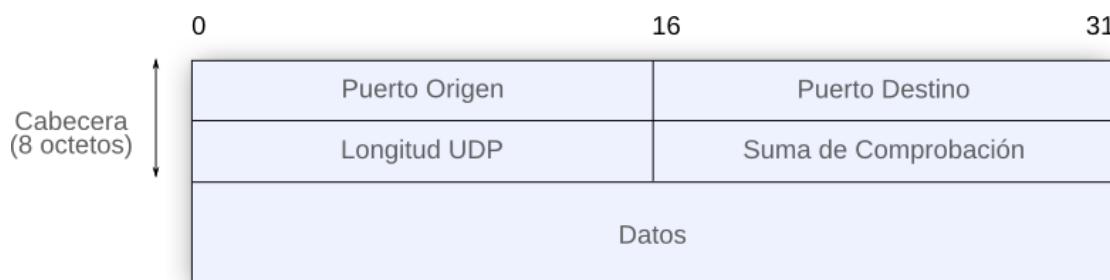
Formato de un datagrama UDP

Un **datagrama UDP** engloba dos tipos de información:



- **Cabecera de control:** Contiene la información que manejan las entidades UDP para llevar a cabo sus respectivas funciones. Por las mínimas funciones de transporte que realiza, la cabecera UDP es muy simple comparada con la cabecera TCP.
- **Datos:** Incluye la cabecera de información de control de la correspondiente aplicación y los potenciales datos del mensaje de dicha aplicación (si existen). La entidad del nivel de aplicación debe pasar los datos en bloques bien delimitados ya que no existe un servicio de flujo de octetos (*byte-stream*) como en TCP.

Concretamente, los campos de esta cabecera de control son los siguientes:



- **Puerto Origen** (16 bits): Identifica al proceso de aplicación emisor que envía un datagrama UDP.
- **Puerto Destino** (16 bits): Identifica al proceso de aplicación receptor que recibe un datagrama UDP.
- **Longitud UDP** (16 bits): Indica la longitud en octetos del datagrama UDP completo incluyendo la cabecera y los datos. Como mínimo un datagrama UDP tiene una longitud de 8 octetos. A pesar de que la máxima longitud de un datagrama UDP puede ser de 65535 octetos no es común ver datagramas UDP mayores de 512 octetos. Por otro lado, no sería necesario el campo Longitud UDP ya que en un paquete IP hay un campo longitud total y otro que define la longitud de la cabecera. Por tanto, si se resta el valor del segundo campo al primero, se puede deducir la longitud del datagrama UDP que está encapsulado en un datagrama IP. Sin embargo, se usa el campo Longitud UDP por una cuestión de eficiencia para la entidad UDP destino.
- **Suma de Comprobación**(16 bits): Suma aritmética binaria o en módulo 2 (sin acarreo o suma OR exclusiva) de todos los bloques de 16 bits del datagrama UDP completo (cabecera y datos). Si la longitud del datagrama no es múltiplo de 16 bits, el datagrama se rellena de ceros hasta hacerlo múltiplo de 16 bits. Si se detecta un datagrama UDP con errores físicos se elimina y no se entrega al proceso de aplicación. Su uso es opcional, es decir, si una entidad UDP de origen no desea calcular la suma de comprobación, este campo debe contener sólo ceros, de forma que la entidad UDP de destino sepa que la suma de comprobación no se ha calculado. Puede suceder que la

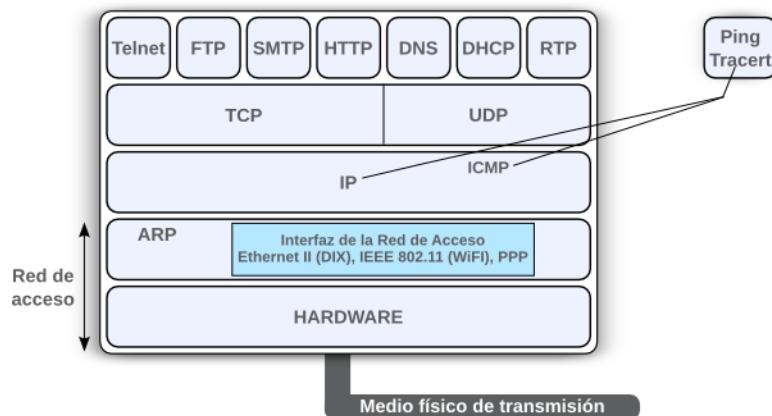
entidad UDP de origen calcule la suma de comprobación y encuentre que el resultado es cero. En este caso se establece un campo de suma de comprobación con todo a unos. Al igual que en TCP, el procedimiento de cálculo es similar al utilizado por IP salvo por los dos siguientes puntos:

- Cuando la longitud del datagrama UDP no es un múltiplo de 16 bits, el datagrama se rellena de ceros hasta hacerlo múltiplo de 16 bits. Sin embargo, el datagrama real que se ha de enviar no se modifica por lo anterior.
- Se incluye una pseudocabecera al inicio del datagrama UDP cuando se calcula la suma de comprobación. Esta pseudocabecera que tampoco se transmite, se crea en el origen y destino durante el proceso de cálculo. Dicho procedimiento le permite al módulo UDP, por un lado, identificar inmediatamente la comunicación a la cual pertenece el datagrama UDP y, por otro lado, asegurarse de que dicho datagrama UDP ha alcanzado realmente la máquina de destino y el pertinente puerto.

Nivel de aplicación

Introducción y generalidades

En el nivel de aplicación de la arquitectura TCP/IP se podrían ejecutar los siguientes protocolos:



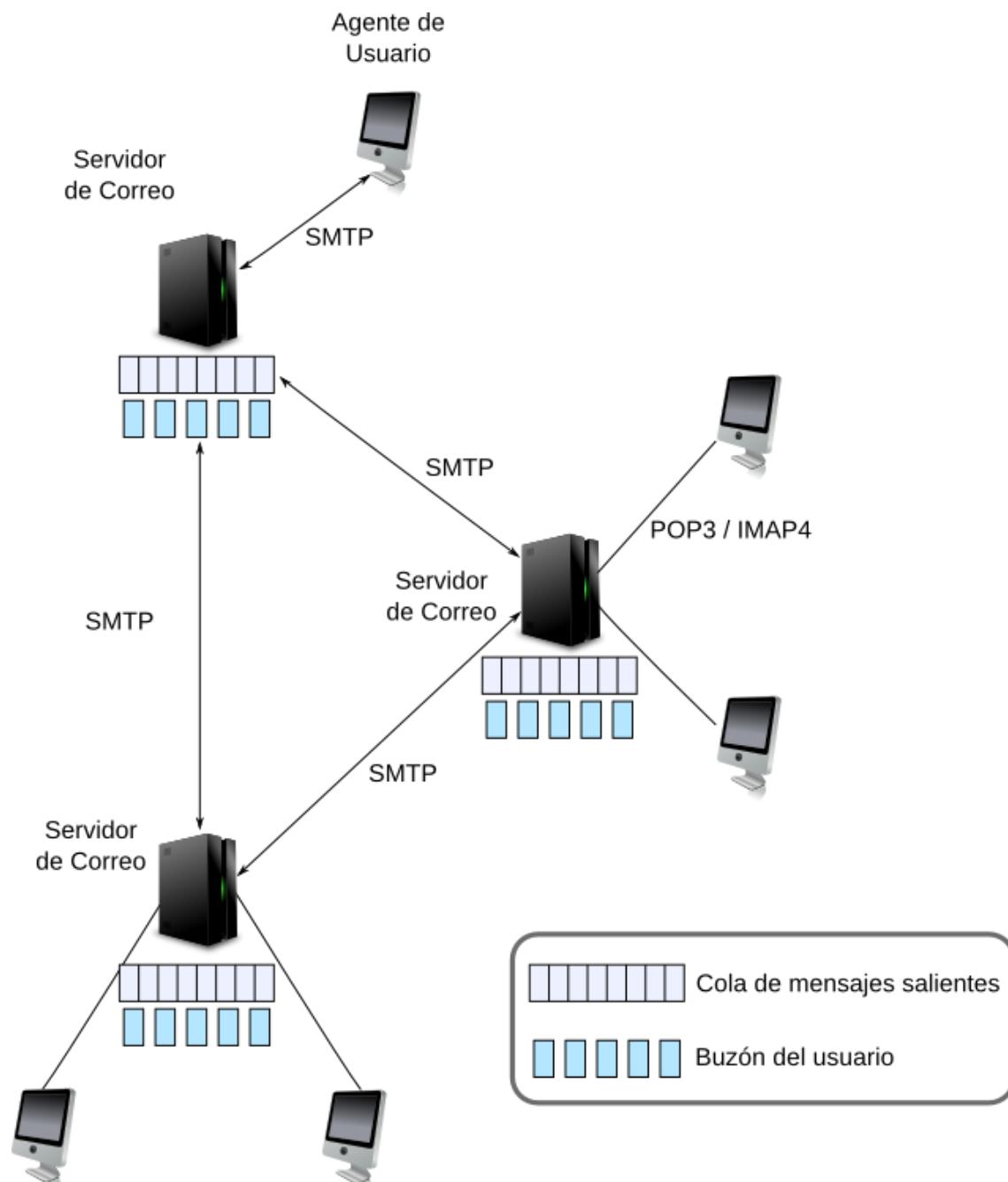
- TELNET: Ofrece un servicio de terminal remoto sobre TCP.
- FTP: Proporciona un servicio fiable de transferencia de ficheros sobre TCP.
- SMTP: Ofrece un servicio de envío de correo sobre TCP.
- HTTP: Proporciona un servicio sobre TCP de copiado de páginas en formato HTML para su visualización o interpretación local en la máquina del usuario.
- DNS: Ofrece un servicio de traducción de una dirección simbólica en su correspondiente dirección IP vía UDP.
- DHCP: Proporciona toda la información de configuración TCP/IP incluyendo una dirección IP para que cualquier sistema pueda crear, inmediatamente, su tabla de encaminamiento IP y acceder a una red TCP/IP o a Internet.

En este contexto, conviene distinguir entre servicios del nivel de aplicación que siguen o no el modelo cliente-servidor:

- Servicios que se rigen según un protocolo de comunicaciones del nivel de aplicación:
 - Telnet, FTP, SMTP, HTTP (Web), NFS, DNS, DHCP, etc.
- Servicios basados en herramientas o utilidades que no se rigen según un protocolo de comunicaciones y sí en base a comandos:
 - PING, IPCONFIG (IFCONFIG en Unix), ARP, TRACERT (TRACEROUTE en Linux), NSLOOKUP, etc.

Servicio de correo electrónico

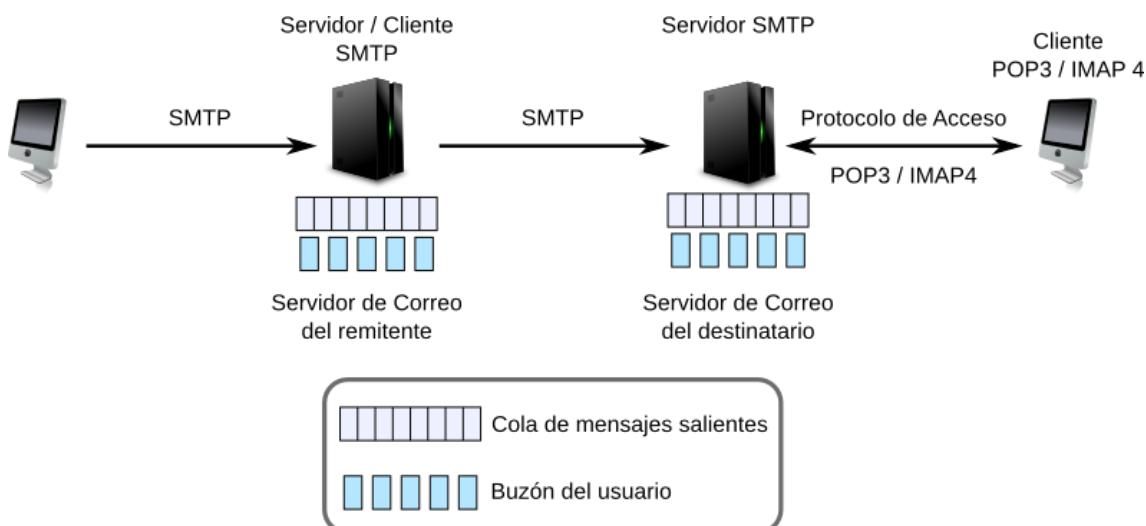
El correo electrónico es un servicio de aplicación TCP/IP basado en tres componentes fundamentales:



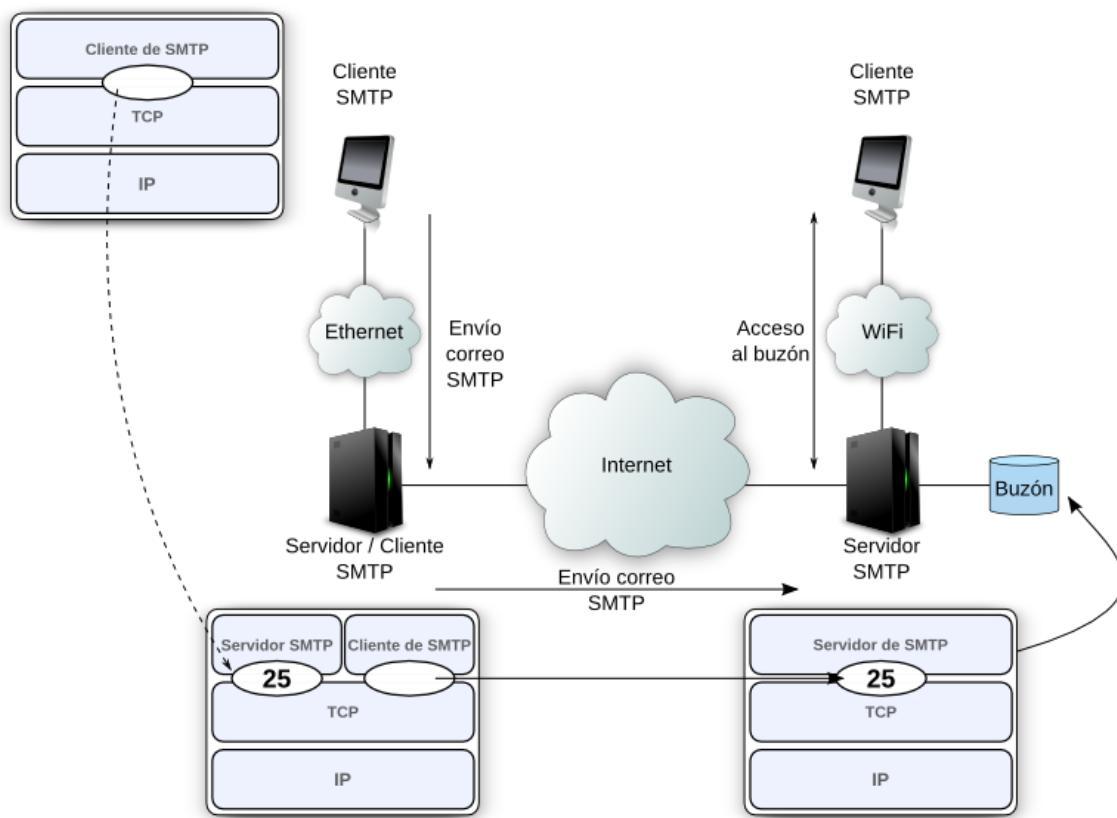
- **Agente de usuario** (por ejemplo, Microsoft Office Outlook). El agente de usuario del correo se corresponde con el propio sistema de correo en la máquina del usuario. Dispone, a su vez, de los siguientes elementos:
 - Un proceso cliente de correo SMTP
 - Un editor de texto
 - Un codificador/decodificador MIME
 Un protocolo de acceso al correo POP3/IMAP4 para recuperar el correo desde el buzón del destinatario en su servidor de correo a un directorio de su disco duro.
- **Servidor de correo:** Se corresponde con el sistema de correo en el servidor de correo de la organización del usuario o en la red IP de su ISP. Dispone de los siguientes elementos:
 - Un proceso servidor de correo SMTP
 - Buzones de los usuarios
 - *Buffer* de mensajes salientes
- **Protocolo de Envío de Correo SMTP (*Simple Mail Transfer Protocol*):** Permite enviar correo desde el cliente de correo SMTP de un agente de usuario a su servidor de correo SMTP por omisión. También permite enviar correo desde el servidor origen (del remitente) al servidor destino (buzón del destinatario).

Servicio de envío de correo: Protocolo SMTP

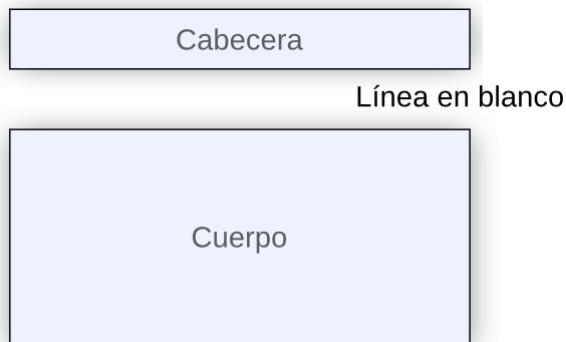
El **protocolo SMTP (*Simple Mail Transfer Protocol*)** proporciona un servicio de envío de correo electrónico por Internet. Este protocolo permite a un usuario enviar correo a otros usuarios aunque éstos no estén activos durante la comunicación, es decir, no estén conectados simultáneamente. Esto es así, ya que se adopta la idea del **buzón electrónico**, es decir, un directorio o carpeta particular donde los mensajes se depositan hasta que el receptor del mismo, se conecta y los recupera. Por regla general, cada usuario dispone de su propio buzón en la máquina servidora de correo de su organización (o de su ISP) donde se está ejecutando un proceso servidor SMTP. Asimismo, el usuario en su computadora dispone del correspondiente cliente SMTP que se ejecuta en el entorno de su aplicación de correo (agente de usuario).



Cuando un usuario (usuario1) compone un mensaje y decide transmitirlo al buzón de otro usuario (usuario2), el proceso cliente SMTP del primero, identificado con un número de puerto TCP (un número libre), transfiere el mensaje a su proceso servidor SMTP, identificado siempre en cualquier máquina servidora con el número de puerto 25 asociado al protocolo TCP. El proceso servidor SMTP se ejecuta en una máquina servidora de la organización del usuario emisor (o de su ISP) en la cual está su buzón. Dicho proceso servidor SMTP procede a analizar la dirección de correo del destinatario (usuario2@identificador_máquina). Por el identificador_máquina, en donde se ejecuta el proceso servidor SMTP, se da cuenta que el mensaje se debe transferir al buzón de otra máquina remota. El proceso servidor de SMTP del usuario1 se convierte en un proceso cliente de SMTP del proceso servidor de SMTP del usuario2. Seguidamente, el proceso servidor de SMTP del usuario 2 transfiere el mensaje al correspondiente buzón (usuario2).



El formato de texto de los mensajes de correo se basan en el estándar RFC-822 que define algunos campos de encabezado (To:, Cc:, Bcc:, From:, etc.), una línea en blanco y el cuerpo del mensaje. Sólo puede enviar mensajes en formato ASCII de 7 bits y no se puede utilizar para enviar ficheros que no contienen texto como son los ficheros binarios, de sonido o vídeo.



La extensión de correo electrónico multipropósito **MIME** (*Multipurpose Internet Mail Extensions*) permite enviar datos en formato no ASCII a través del correo electrónico. MIME transforma los datos en formato no ASCII en el lado emisor a datos ASCII y lo entrega al cliente de correo electrónico para que sean enviados a través de Internet. El mensaje en el lado receptor se transforma de nuevo a los datos originales.

La idea básica de MIME es continuar usando el formato RFC-822, pero agregando una estructura al cuerpo del mensaje y definir reglas de codificación para los mensajes no ASCII. Todo lo que hay que hacer es añadir software MIME en los programas emisores y receptores de correo.

MIME define cinco nuevos encabezados de mensaje:

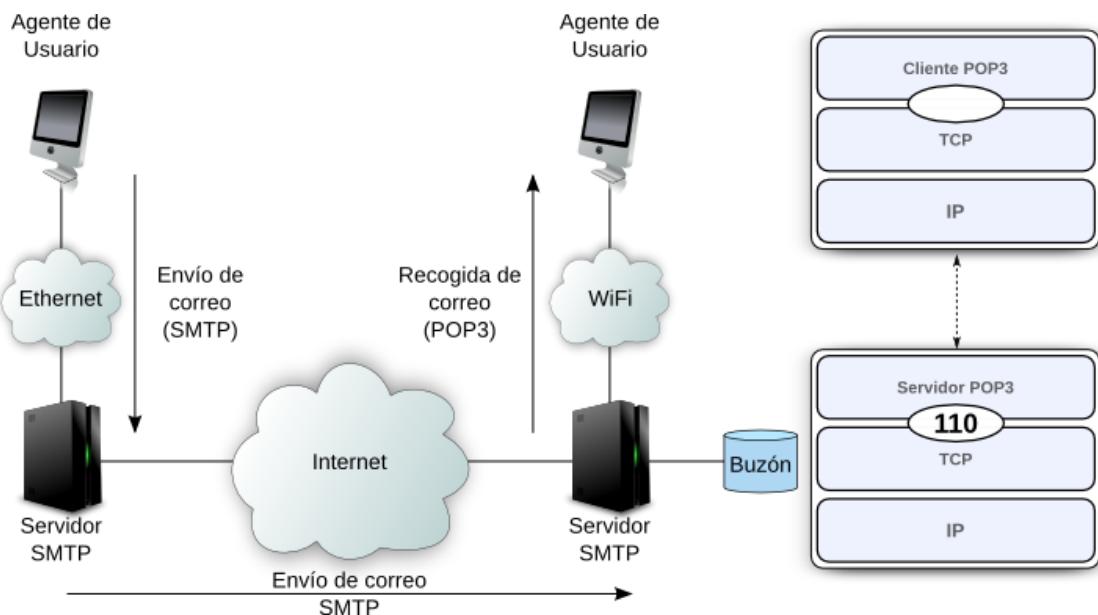
Encabezado	Significado
MIME - Version	Identifica la versión MIME
Content - Description	Cadena de texto que describe el contenido
Content - Id	Identificador único
Content - Transfer - Encoding	Cómo se codifica el mensaje para su transmisión
Content - Type	Naturaleza del mensaje

- **MIME-Version:** Identifica la versión de MIME.
- **Content-Description:** Cadena de texto que describe el contenido.
- **Content-Id:** Identificador único.
- **Content-Transfer-Encoding:** Cómo MIME codifica el mensaje para su transmisión. El esquema más sencillo es simplemente texto ASCII y el sistema de codificación por excelencia el base 64.
- **Content-Type:** Naturaleza del mensaje. Desde texto normal (text/plain) hasta ficheros postscript, imágenes fijas (jpeg), audio (mp3), vídeo (mpeg4), etc.

Servicio de recogida del correo: Protocolo POP3

El **protocolo POP3** (*Post Office Versión 3*) proporciona un servicio de recogida de mensajes del buzón de correo. El buzón de un usuario se encuentra en la máquina servidora de correo y no en su computadora. Por tanto, se necesita de un protocolo como POP3 que recoja el correo del buzón y lo traiga a un directorio de su disco duro.

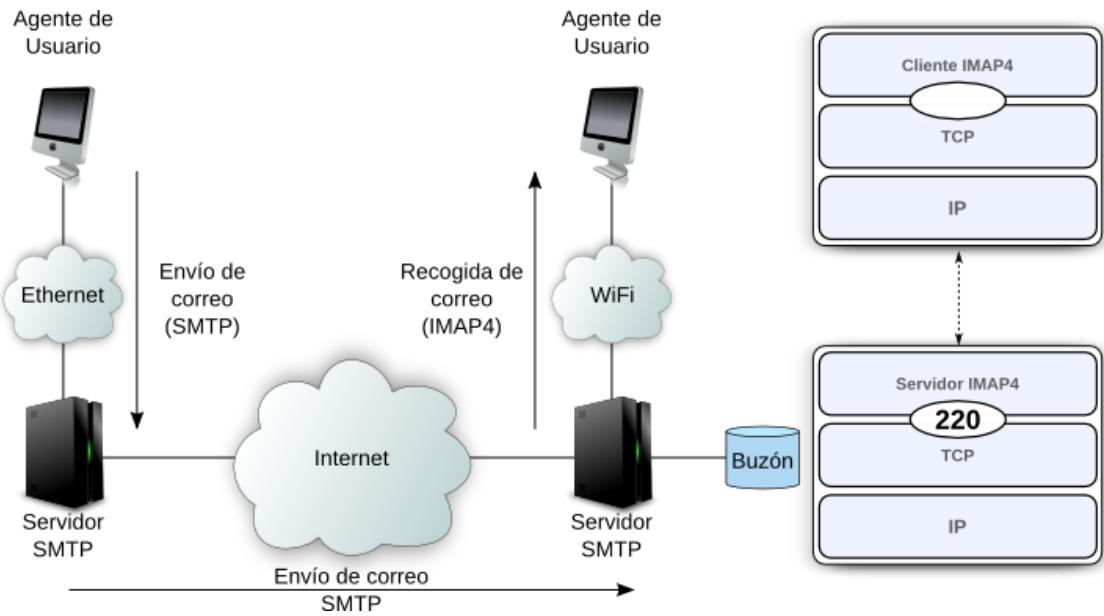
El procedimiento consiste en que un proceso cliente POP3 identificado con un número de puerto TCP (un número libre) se comunica (para recoger y traer el correo) con el proceso servidor POP3 identificado siempre, en cualquier máquina servidora, con el número de puerto TCP 110. El proceso servidor POP3 se ejecuta en la máquina servidora de correo donde se ejecuta el proceso servidor SMTP. La aplicación de correo, que se ejecuta en la máquina del usuario, dispone de un proceso cliente SMTP y un proceso cliente POP3.



Servicio de gestión del correo: Protocolo IMAPv4

El **protocolo IMAP4** (*Internet Message Access Protocol Rev 4*) proporciona un servicio de gestión de mensajes en el mismo buzón de correo. Este protocolo permite al usuario gestionar todo su correo en su propio buzón sin necesidad de recoger todos los mensajes y traerlos al disco duro de su máquina como se hacía por omisión con POP3. El protocolo IMAP4 permite eliminar, clasificar y distribuir su correo en distintas carpetas dentro de la propia máquina servidora de correo. Asimismo, permite copiar o mover mensajes previamente seleccionados desde el buzón hasta su computadora. Es importante resaltar que el correo no se lee a través de la red sino que se trae vía IMAP4 de forma temporal a un directorio local en la máquina del usuario.

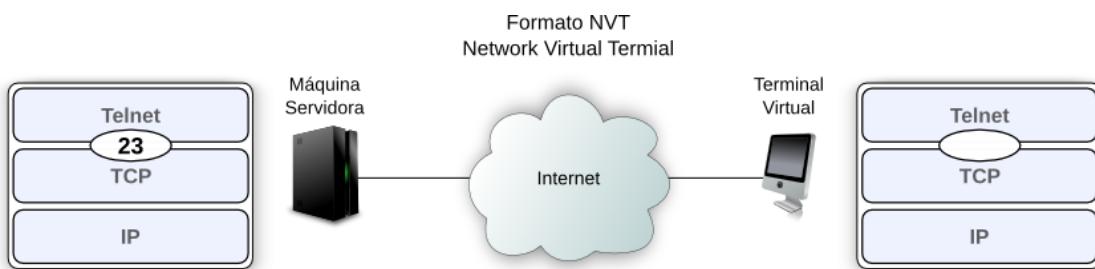
El procedimiento de trabajo de este protocolo consiste en que un proceso cliente IMAP4 identificado con un número de puerto TCP (un número libre) se comunica (para la gestión del correo) con el proceso servidor IMAP4 identificado siempre, en cualquier máquina servidora, con el número de puerto TCP 220. El proceso servidor IMAP4 se ejecuta en la máquina servidora de correo en donde se ejecuta el proceso servidor SMTP. La aplicación de correo, que se ejecuta en la máquina del usuario, dispone de un proceso cliente SMTP y un proceso cliente IMAP4.



Servicio de acceso remoto: Protocolo Telnet

El **protocolo Telnet** ofrece un servicio de terminal remoto, permitiendo que la máquina del usuario se convierta en un terminal de una máquina servidora por Internet o por una red privada TCP/IP. A través de este protocolo, el usuario tiene la apariencia de estar sentado delante de la pantalla de la máquina servidora aunque ésta esté dispersa geográficamente por Internet a miles de kilómetros.

El procedimiento consiste en que un proceso cliente Telnet identificado con un número de puerto TCP (un número libre) se comunica con el proceso servidor de Telnet identificado siempre, en cualquier máquina servidora, con el número de puerto TCP 23. Se asume que el usuario dispone de cuenta (login/password) en la máquina servidora remota, es decir, está debidamente registrado por el administrador de dicha máquina.

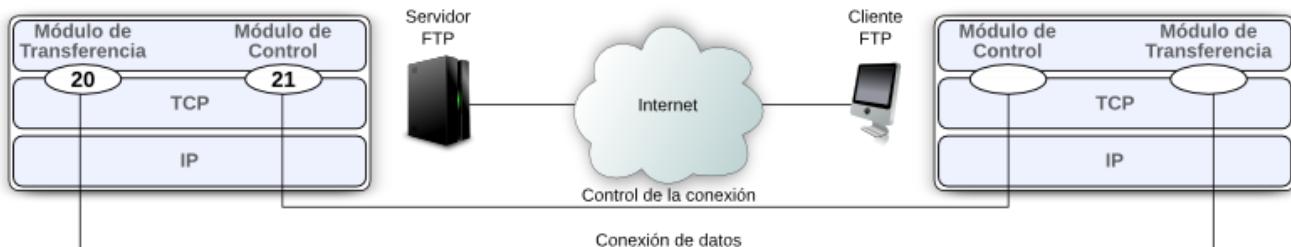


En concreto, el servicio de terminal remoto ofrecido por el protocolo Telnet se basa en unas secuencias de escape o caracteres de control ASCII pertenecientes a un**Terminal Virtual de Red** que permite una vez establecida la conexión TCP, negociar el modo de operación (opciones de comunicación) entre el proceso cliente y servidor Telnet. La negociación del modo de operación permite indicar el tipo de terminal (p.ej., VT100, ANSI, etc.), el tipo de comunicación (dúplex o semidúplex), modo línea o modo carácter, etc. Un terminal VT100 o un terminal ANSI es un software (intérprete) incluido en el telnet cliente y servidor que

permite interpretar las secuencias de escape para la definición del número de filas y columnas en la pantalla del cliente, el movimiento del cursor, tabulados, retornos de carro, etc.

Servicio de transferencia de ficheros: Protocolo FTP

El **protocolo FTP** (*File Transfer Protocol*) ofrece un servicio de transferencia de uno o más ficheros (ASCII o binarios) entre dos máquinas.



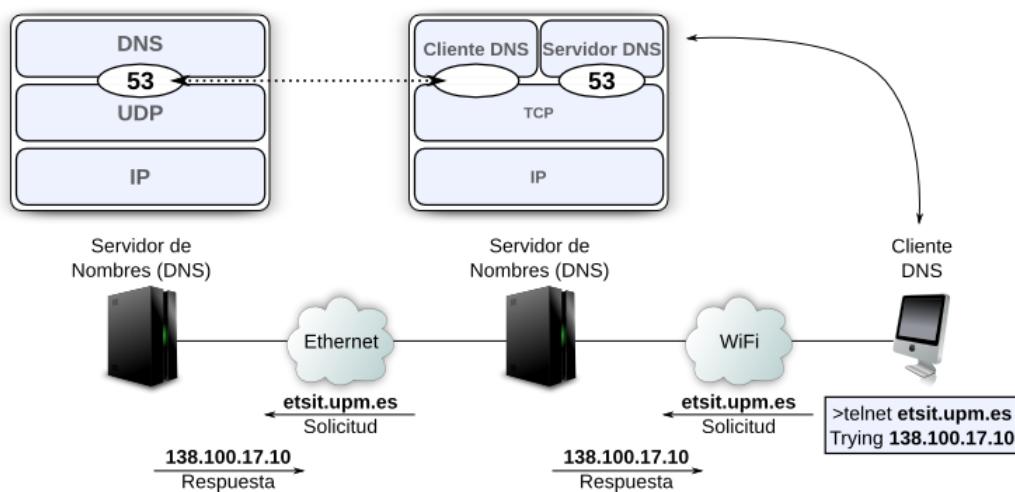
El protocolo FTP consiste en dos módulos o procesos o entidades diferentes:

- **Módulo de Control:** Basado en un proceso cliente identificado con un número de puerto TCP (un número libre) que se comunica con el correspondiente proceso servidor identificado siempre, en cualquier máquina servidora, con el número de puerto TCP 21.
- **Módulo de Transferencia de Datos:** Basado en un proceso cliente identificado con un número de puerto TCP (un número libre) que se comunica con el correspondiente proceso servidor identificado siempre, en cualquier máquina servidora, con el número de puerto TCP 20.

En consecuencia, el protocolo FTP utiliza dos números de puerto, uno (21) para establecer la conexión entre dos entidades FTP y otro (20) para transferir ficheros (get/put) entre dichas máquinas sin salir de la conexión previamente establecida.

Servicio de resolución de direcciones: Protocolo DNS

El **protocolo DNS** (*Domain Name System*) ofrece un servicio de traducción de una dirección simbólica en una dirección IP de máquina.



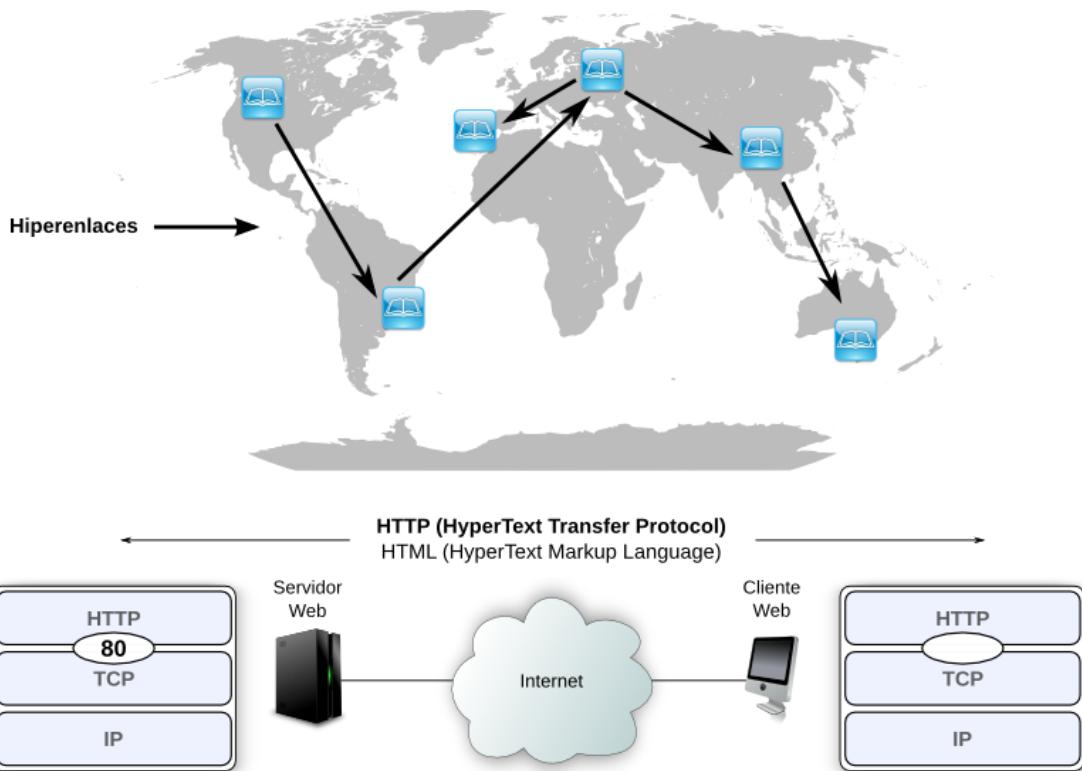
El procedimiento consiste en que un proceso cliente DNS identificado con un número de puerto UDP (un número) se comunica con el proceso servidor DNS identificado, siempre en cualquier máquina servidora, con el número de puerto UDP 53. El proceso cliente DNS se ejecuta en la máquina del usuario y el proceso servidor DNS en la máquina servidora de la organización (o del proveedor de servicios). Si el proceso servidor DNS, con el que interactúa directamente el proceso cliente DNS, no sabe traducir la dirección simbólica en su formato numérico; entonces, se transforma en un proceso cliente DNS de otro proceso servidor DNS de superior jerarquía en la estructura por niveles del servicio DNS existente por Internet. Y así se procederá hasta encontrar (o no, dando por concluida la fase de búsqueda) la asociación simbólica-numérica invocada. Una vez encontrado el formato numérico solicitado, éste se transmite por el mismo camino de invocaciones.

En una organización se puede encontrar dos tipos de servidores DNS:

- **Servidores DNS primarios** (*Primary Name Servers*): Almacenan la información de dominios en una base de datos local. Son responsables de mantener la información de los dominios actualizada, por lo que cualquier cambio en los datos o cualquier alta o baja de dominio debe ser comunicada a estos servidores.
- **Servidores DNS secundarios** (*Secondary Name Servers*): Se encuentran por debajo de los anteriores en la jerarquía, por lo que deben obtener de ellos los datos correspondientes a su zona de acción mediante un proceso de copia denominado “transferencia de zona”. Además, estos servidores actúan como sistemas de seguridad, al mantener la información de forma redundante, con lo que si un servidor DNS tiene problemas, la información se puede recuperar desde otro. Además, evitan la sobrecarga del servidor principal, distribuyendo el trabajo entre distintos servidores situados estratégicamente con lo que se gana velocidad en las resoluciones.

Servicio World Wide Web (WWW)

El **servicio World Wide Web** (WWW) o simplemente Web es un inmenso conjunto de información diseminado por todo el mundo y enlazado entre sí. En concreto, WWW es una colección mundial de ficheros de texto y multimedia en **código HTML** (*HyperText Makeup Language*) y ligados mutuamente a través de un sistema de documentos de hipertexto e hipermedia vía el **protocolo HTTP** (*HyperText Transfer Protocol*). Con el hipertexto e hipermedia, una palabra, frase, imagen, etc., pueden contener un hiperenlace con otro texto, audio, imagen y vídeo.



Desde el punto de vista tecnológico, el servicio Web es un servicio distribuido del tipo cliente-servidor en el que un cliente o navegador Web puede acceder a una página Web de inicio ofrecida como servicio desde un servidor Web. Sin embargo el servicio está distribuido sobre muchas localizaciones denominados sitios.

Cada sitio almacena uno o más documentos, denominados páginas Web. Cada página Web puede contener un enlace a otras páginas del mismo sitio o de otros sitios. Las páginas se pueden recuperar y visualizar utilizando navegadores Web. Un navegador consta fundamentalmente de un protocolo cliente que, generalmente, es HTTP y de un intérprete que puede ser HTML, Java o JavaScript, dependiendo del tipo de documento Web que se desea visualizar o interpretar.

Localizador de recursos URL

Un cliente que quiera acceder a una página Web necesita una dirección. Para facilitar el acceso a una página Web, el protocolo HTTP utiliza un localizador o **URL (Uniform Resource Locator)** que identifica de forma única a la página en Internet. El URL define cuatro aspectos:

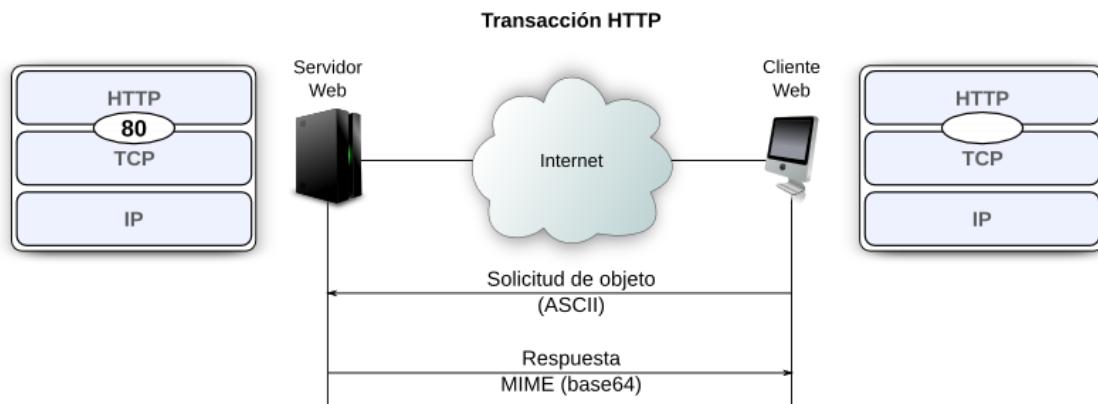
protocolo://máquina:puerto/ruta

- **Protocolo:** lenguaje utilizado entre el cliente y el servidor para recuperar el documento. Se pueden utilizar diferentes protocolos para recuperar un documento. El más común es HTTP por ser el lenguaje nativo del servicio Web.
- **Máquina:** computadora en la que se localiza la información. Las páginas Web se almacenan, generalmente, en computadoras y las computadoras tienen un alias que, normalmente, comienzan con “www”. Esto no es obligatorio, ya que la estación puede ser cualquier nombre dado a la computadora que almacena la página Web.

- **Puerto:** número entero que identifica al proceso servidor. El URL puede contenerlo opcionalmente.
- **Ruta:** nombre del fichero donde se encuentra la información. Puede contener separadores (/) para separar los directorios de los subdirectorios y ficheros.

Servicio de transacciones Web: Protocolo HTTP

El **protocolo HTTP (HyperText Transfer Protocol)** está basado en un modelo cliente-servidor orientado a transacciones. La versión actual de HTTP es la 1.1. HTTP dispone de una variante cifrada mediante SSL llamada HTTPS. Para proporcionar fiabilidad, HTTP hace uso de TCP. Un cliente HTTP (navegador) abre una conexión y envía su solicitud al servidor Web, el cual responderá con el recurso solicitado si está disponible y si permite su acceso.



HTTP funciona como una combinación de los protocolos FTP y SMTP. Es similar a FTP debido a que trasfiere ficheros y utiliza los servicios de FTP. Sin embargo, es mucho más sencillo que FTP debido a que sólo utiliza una conexión TCP. No hay una conexión de control diferente; sólo se transfieren datos entre el cliente y el servidor. A su vez, HTTP es similar a SMTP debido a que los datos transferidos entre el cliente y el servidor se parecen a mensajes SMTP. Además, el formato de los mensajes es controlado por cabeceras MIME. Al contrario que SMTP, los mensajes HTTP no están destinados a ser leídos por las personas; son leídos e interpretados por el servidor HTTP y el cliente HTTP (navegador). Los mensajes SMTP se almacenan y reenvían, pero los mensajes HTTP se entregan inmediatamente. Las órdenes del cliente al servidor se introducen en un mensaje de solicitud. El contenido del fichero solicitado u otra información se incluyen en un mensaje diferente.

Una característica de HTTP es su independencia en la visualización y representación de los datos, independientemente del desarrollo de nuevos avances en dichas visualizaciones y representaciones de los datos. El protocolo HTTP es un servicio distribuido de presentación de la información basado en hiperenlaces que permite copiar una página en formato HTML para su visualización o interpretación local en la máquina del usuario. El procedimiento operativo consiste en que un proceso cliente HTTP identificado con un número de puerto TCP (un número libre) se comunica con el proceso servidor HTTP identificado siempre, en cualquier máquina servidora, con el número de puerto TCP 80. Un usuario que quiera disponer de su propio servidor Web (servidor Web privado, diferente del servidor Web del administrador), y no sólo de su página Web, arrancará dicho servidor en un número de puerto diferente al 80; generalmente, se suele poner el 8080. E

El proceso cliente HTTP se ejecuta en la máquina del usuario y el proceso servidor HTTP en la máquina servidora de la página Web solicitada.

Características del protocolo HTTP 1.1

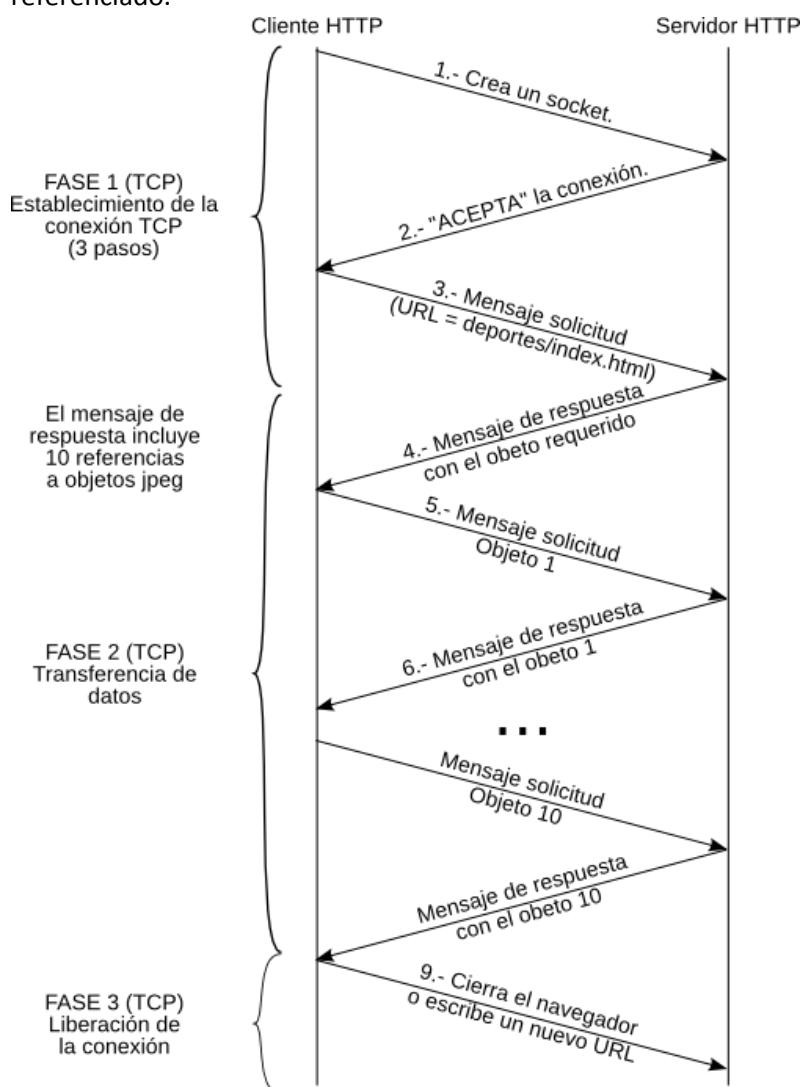
El protocolo HTTP 1.1 es un protocolo cliente-servidor orientado a transacciones. Para proporcionar fiabilidad, HTTP hace uso de TCP. Un cliente HTTP (navegador) abre una conexión y envía su solicitud al servidor Web, el cual responderá con el recurso solicitado si está disponible y se permite su acceso. Seguidamente, la conexión se cierra.

Las principales características de HTTP 1.1 son:

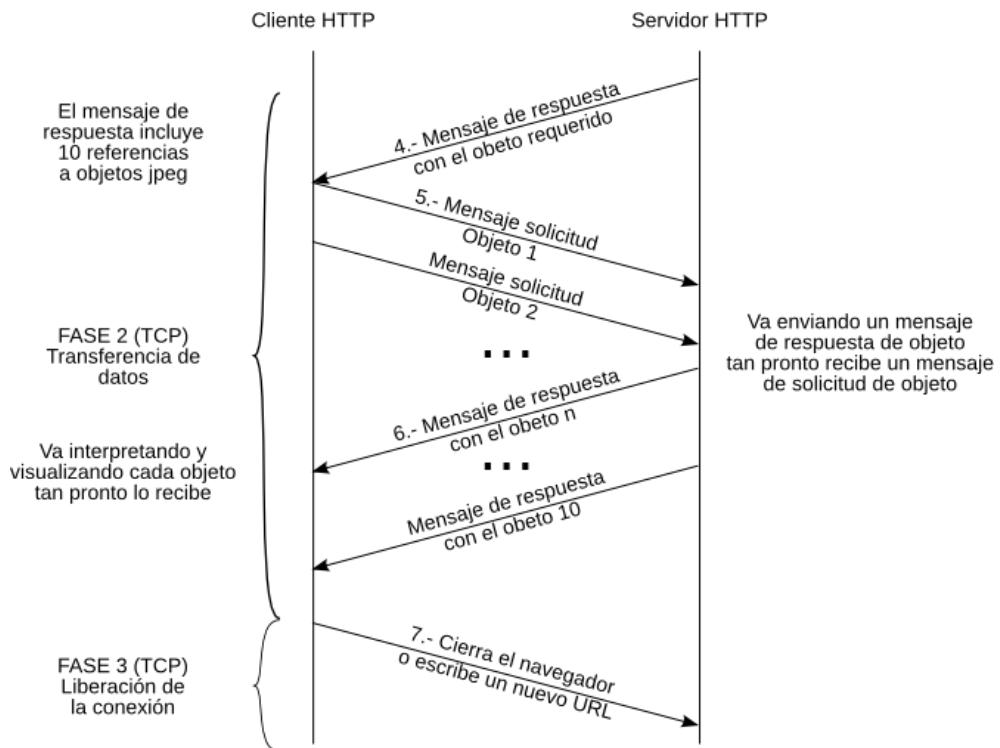
- **Persistente:** Soporta conexiones persistentes entre el cliente y el servidor. Se pueden enviar múltiples objetos por una única conexión TCP. Una típica página Web puede contener múltiples iconos, imágenes, etc., de tal manera que es necesario enviar múltiples solicitudes y, todo ello, en la misma conexión TCP.

En HTTP 1.1 existen dos tipos de persistencia:

- **Persistencia sin pipelining:** El cliente envía una nueva solicitud sólo cuando ha recibido el objeto de la anterior solicitud. Un RTT por cada objeto referenciado.



- **Persistencia con pipelining:** El cliente envía más de una solicitud tan pronto encuentra más de un objeto referenciado. Un solo RTT para todas las referencias. Por omisión en HTTP 1.1.

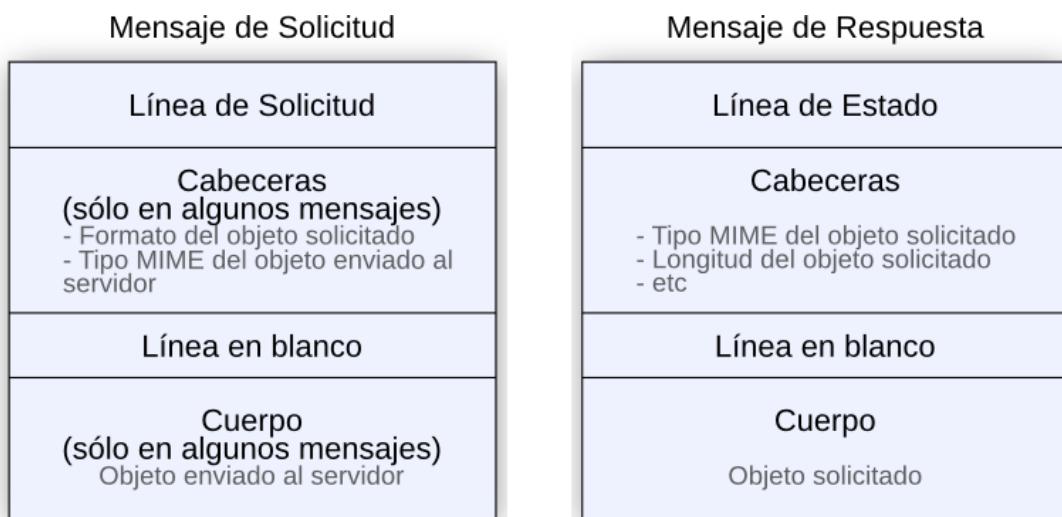


- **Sin estado:** El servidor HTTP no mantiene el estado o la información sobre las solicitudes (historial) de los clientes HTTP al cerrarse la conexión TCP. Para mantener el estado, el programador de la aplicación Web tendrá que gestionar el estado por encima de HTTP:

- **Reescritura de los URL:** Captura posterior de la información del usuario incrustada previamente por el servidor en cada URL.
- **Campos ocultos en la página HTML:** Captura posterior de la información del usuario incrustada por el servidor en campos no visibles HTML.
- **Cookies ("galletas"):** Ficheros de texto o fragmentos de información (trozos de datos) diferentes que contienen las acciones del usuario para cada servidor Web visitado y que se almacenan en el disco duro del cliente, a través de su navegador, a petición del servidor Web. Esta información puede ser recuperada luego por el servidor en posteriores visitas para diferenciar usuarios y actuar de diferente forma dependiendo del usuario. Se utilizan para:
 - Control de usuarios: Cuando un usuario introduce su nombre de usuario y contraseña, se almacena una cookie para que no tenga que estar introduciéndolas por cada página del servidor
 - Seguimientos de usuarios: Estadísticas de usos, aficiones, cestas virtuales de compras, etc.
 - Personalización del sitio Web en función de los hábitos o preferencias del usuario: Particularizar el aspecto en cuanto a presentación y funcionalidad.

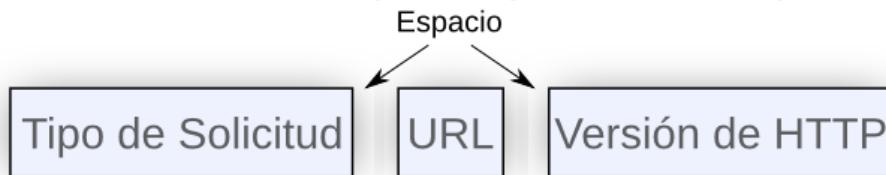
Formato de un mensaje HTTP 1.1

El protocolo HTTP especifica el formato de los mensajes entre los clientes y los servidores Web. El formato de los mensajes HTTP 1.1 de solicitud y respuesta es similar en cuanto a formato. Un mensaje de solicitud consta de una línea de solicitud, una cabecera y en algunas ocasiones un cuerpo. A su vez, un mensaje de respuesta consta de una línea de estado, una cabecera y, en muchas ocasiones, un cuerpo con el objeto solicitado.



A su vez, el Tipo de Solicitud es el primer campo de la línea de solicitud de un mensaje de solicitud.

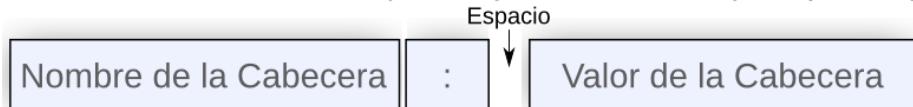
Línea de solicitud (Mensaje de solicitud)



Línea de estado (Mensaje de respuesta)



Formato de la cabecera (Mensajes de solicitud y respuesta)



Existen diferentes tipos de solicitud en función de los métodos HTTP utilizados. Los tres métodos o solicitudes más relevantes son:

- **GET:** Método HTTP para solicitar un objeto, a veces, con parámetros que se encapsulan en el localizador URL e indican exactamente lo que el usuario necesita o

busca. La mayoría de las solicitudes HTTP son mediante GET. Los parámetros pasados se codifican (pares de valores: nombre-valor) y son visibles en el URL.

- **POST:** Método HTTP para solicitar un objeto siempre con parámetros del usuario que no son visibles en el localizador URL. Típica solicitud HTTP cuando se hace clic en el botón de un formulario. Los parámetros se introducen en el cuerpo del mensaje (pares de valores: nombre-valor) y no son visibles en el localizador URL. El tamaño de los datos no está limitado como en GET.
- **PUT:** Método HTTP para enviar un objeto al servidor.

Monitorización y gestión de redes TCP/IP