GENERALITAT VALENCIANA
Conselleria d'Educació, Investigació, Cultura i Esport

Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

IES MARÍA ENRÍQUEZ

C/ Literat Azorín, 1 46702 – GANDIA
Tfno. 962829430 – Fax. 962829431
Codi: 46004221 – CIF: Q-9655627-I
46004221@edu.gva.es
http://iesmariaenriquez.com

COMPUTER SYSTEMS
UD6: NETWORKS INTRODUCTION_II

CFGS DAW
DPT INF

1. La arquitectura de red

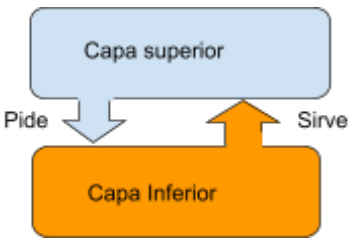
Uno de los problemas más importantes a la hora de diseñar una red no es que los equipos se conecten entre sí, si no que estos equipos puedan comunicarse, entenderse y compartir recursos. Debido a la complejidad que acarrea considerar la red como un todo, se consideró organizar las redes como una serie de capas, donde cada capa se ocuparía de alguna función. De esta forma se reduciría la complejidad del diseño de la red y de las aplicaciones que en ella se utilicen.

Por tanto, podemos definir **arquitectura de red** como el conjunto de capas o niveles, junto con los **protocolos** definidos en cada una de estas capas, que hacen posible que un ordenador se comunique con otro ordenador independientemente de la red en la que se encuentre.

Esta definición implica, que la especificación de una arquitectura de red debe incluir información suficiente para que cuando se desarrolle un programa o se diseñe algún dispositivo, cada capa responda de forma adecuada al protocolo apropiado.

El diseño de una arquitectura basada en niveles está fundamentada en los siguientes principios:

- Cada nivel lleva a cabo unas funciones limitadas y bien definidas.
- El número de niveles y su función puede ser distinto en cada arquitectura de red, lo suficientemente grande como para diferenciar las funciones de cada nivel, pero no tanto como para aumentar la complejidad del proceso.
- Cada nivel debe comunicarse únicamente con su nivel inmediatamente anterior y posterior.
- La comunicación entre niveles se implementa mediante **servicios**: Cada nivel ofrece servicios al nivel superior, y por tanto, utiliza los servicios del nivel inferior.
- Las **interfaces** entre niveles deben estar claramente definidas para respetar el principio de modularidad.
- Las interfaces deben ser sencillas para que se minimice el flujo de información entre los niveles.



Definiciones

- Un **protocolo** es un conjunto de reglas que gobiernan el formato y el significado de los mensajes que intercambian la misma capa en dos máquinas distintas. Un protocolo es un acuerdo entre las partes que se comunican sobre cómo va a proceder la comunicación.

C/ Literat Azorín, 1 46702 – GANDIA
Tfno. 962829430 – Fax. 962829431
Codi: 46004221 – CIF: Q-9655627-I
46004221@edu.gva.es
http://iesmariaenriquez.com

COMPUTER SYSTEMS
UD6: NETWORKS INTRODUCTION_II

CFGS DAW
DPT INF

En una comunicación, la capa x de una máquina se comunica con la capa x de la otra máquina. Las reglas y convenciones que se siguen en la conversación se conocen como protocolo de la capa x. Ej: respuesta del servidor http:

- Una **línea de estado** con la versión del protocolo HTTP utilizado en el servidor, un código de estado y una breve descripción del mismo:

```
HTTP/1.0  200  OK
```

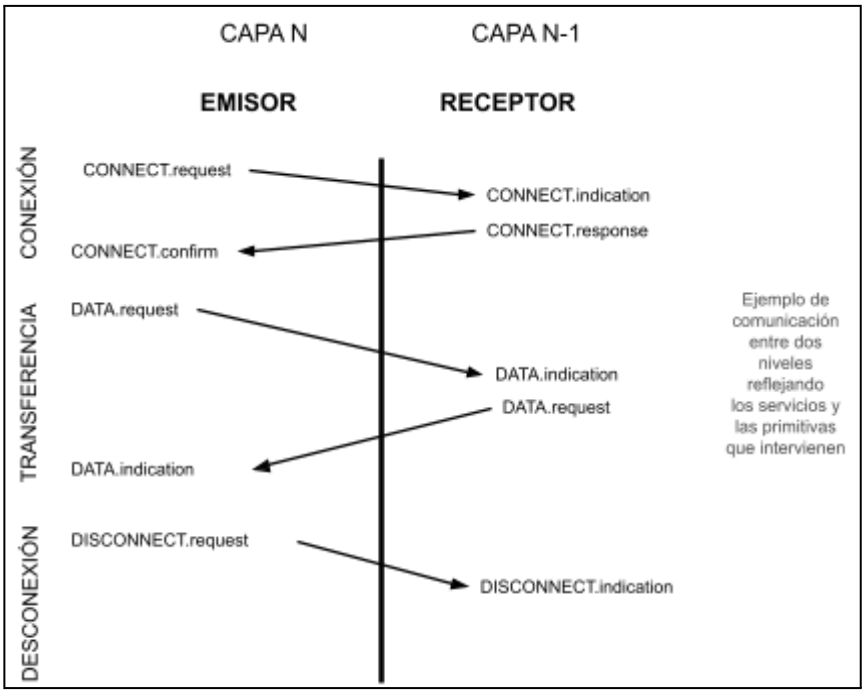
- **Un servicio** es un conjunto de operaciones (llamadas también "primitivas") que ofrece una capa a la que está por encima de ella. Cada primitiva define la función a realizar y posee unos parámetros que permiten datos y /o información de control. En el modelo OSI se emplean cuatro primitivas para definir las interacciones entre dos capas adyacentes:

  - **Request**: esta primitiva es llamada desde la capa superior para solicitar un servicio de la capa inmediatamente inferior.

  - **Indication**: esta primitiva es generada por la capa suministradora del servicio para indicar a la capa superior que ha solicitado un servicio.

  - **Response**: esta primitiva es la respuesta de confirmación de la capa superior de la petición de servicio.

  - **Confirmation**: esta primitiva es emitida por la capa inferior para confirmar el servicio solicitado.



● Entre cada par de capas adyacentes hay una interfaz. **Una interfaz** describe cómo se accede y solicita un servicio. La interfaz define qué operaciones y servicios ofrece la capa inferior a la superior.
● Una **arquitectura de red** es un conjunto de capas y protocolos.
● Cada capa utiliza un protocolo y al conjunto de ellos se le llama **pila de protocolos**.

> **Ejemplo:** La estructura de una empresa de transportes es una buena analogía de como funciona una arquitectura basada en niveles. Una empresa de transportes suele disponer de: un departamento de administración, un departamento de ventas, administradores de distribución, trabajadores de almacenes, y conductores de camiones. Se puede considerar que cada uno de estos grupos es un nivel diferente. Cada uno de ellos depende de los servicios de los departamentos (niveles) adyacentes y, en general, no les afectan las necesidades de los departamentos que no están directamente relacionados con ellos. Los conductores necesitan los servicios de los trabajadores de los almacenes para localizar y entregar los materiales. Sin embargo, los conductores de los camiones no necesitan conocer el funcionamiento del departamento de ventas. Cada departamento podría cambiar su modo de funcionamiento, y algún departamento podría cambiar a sus empleados de puesto, pero las reglas generales de comunicación entre niveles no cambiarían.

Siempre que se pretende una comunicación del tipo que sea, se deben cumplir una serie de requisitos básicos, como son el tipo de lenguaje a utilizar, el tipo de información a transmitir, el momento, el modo, etc. Cuando dos equipos intentan establecer una comunicación deben hablar el mismo lenguaje y ponerse de acuerdo en una serie de normas. Estas normas son lo que denominamos **protocolo**.

1.1. Modelo OSI de interconexión de sistemas

Las primeras redes de ordenadores utilizaban protocolos específicos de cada fabricante que las hacían incompatibles entre ellas.

Como esta situación suponía un problema tanto para los usuarios como para los propios fabricantes, en 1977, la *Organización Internacional de Normalización*, conocida como ***ISO*** (del inglés *International Standard Office*) comenzó el desarrollo de un estándar abierto que facilitara la *interoperabilidad*[1] entre las redes de diferentes fabricantes.

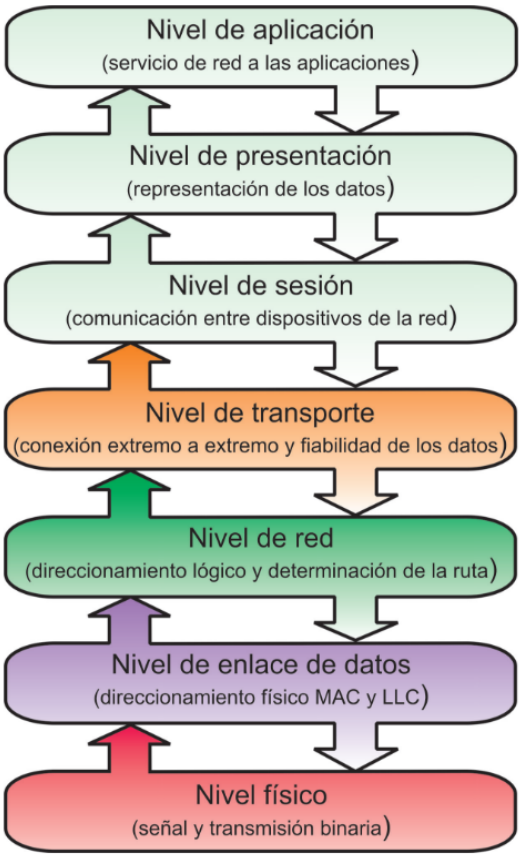
El estándar se publicó en 1984 y fue bautizado como ***modelo OSI*** (*Open Systems Interconnection*, *Interconexión de Sistemas Abiertos*). Aún así, a partir de 1985, el **modelo TCP/IP** comenzó a ganar protagonismo, siendo el que se impuso finalmente.

[1] *El **IEEE** (Institute of Electrical and Electronics Engineers) define la interoperabilidad como la habilidad de dos o más sistemas o componentes para intercambiar información y utilizarla posteriormente.*

En realidad, el **modelo OSI** no se ha implementado en ningún sistema, pero es fundamental entenderlo porque se utiliza a menudo como referencia para compararlo con otros modelos, como el ya mencionado TCP/IP.

Estructura del modelo OSI

Así pues, el **modelo de referencia *OSI*** se divide en siete capas:



- **Nivel de aplicación**. Ofrece a las aplicaciones la posibilidad de acceder a los servicios de las demás capas y define los protocolos que utilizan las aplicaciones para intercambiar datos. Es importante aclarar que el usuario, normalmente, no interactúa con el nivel de aplicación. Suele interactuar con programas que a su vez interactúan con el nivel de aplicación. Ej: **http, ftp, dns, smtp**…

- **Nivel de presentación**. Se encarga de la representación de la información, de manera que aunque distintos equipos puedan tener diferentes representaciones internas de caracteres (ASCII, Unicode, EBCDIC), números (little-endian tipo Intel, big-endian tipo Motorola), sonido o imágenes, los datos lleguen de manera reconocible. Podría decirse que esta capa actúa como traductor.

  Cumple tres funciones principales:

  - Formateo de datos
  - Cifrado de datos
  - Compresión de datos

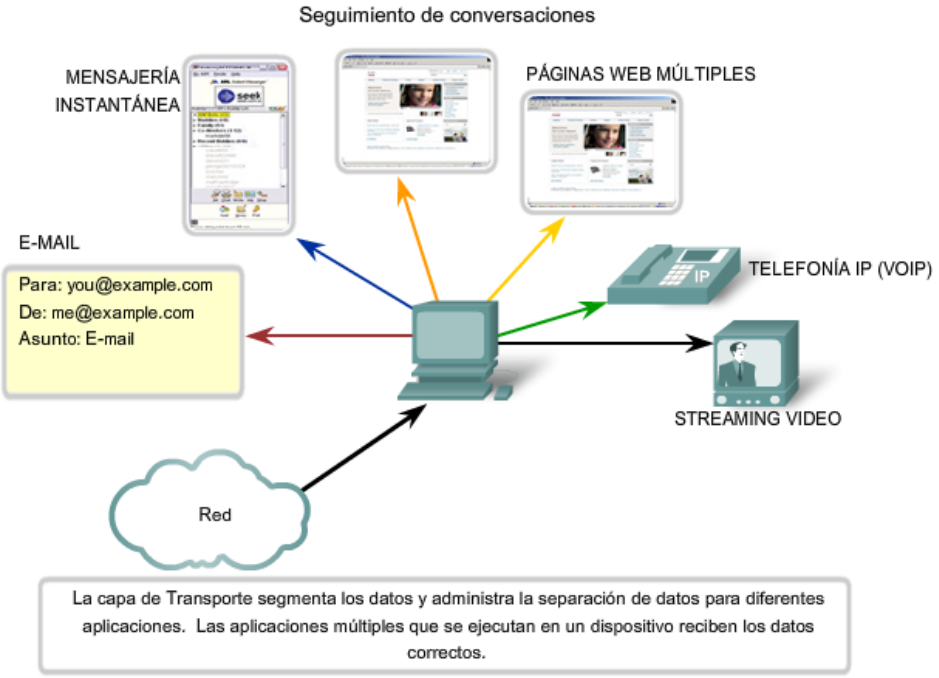
- **Nivel de sesión**. Se encarga de sincronizar el envío de información, mantener y controlar el enlace creado entre dos equipos, estableciendo la conversación, los turnos de palabra, el intercambio de datos, etc. Esta capa está orientada hacia la administración de diálogos y tratamiento de errores no relacionados con la transmisión.

  Se encarga de iniciar una sesión para cada comunicación que se quiera establecer. Así, cada vez que algún host quiere convertirse en emisor se crea y mantiene una sesión de forma que se pasa la información a la capa de transporte para que pueda comenzar a ser tratada y posteriormente enviada a través del medio físico elegido.

  Su objetivo es mantener la sesión mientras dure la transmisión.

Nota: Mientras que las capas de aplicación, presentación y sesión están relacionadas con asuntos de aplicaciones, las cuatro capas inferiores se encargan del transporte de datos

● **Nivel de transporte**. La capa de Transporte permite la segmentación de datos y brinda el control necesario para reensamblar las partes dentro de los distintos streams de comunicación. Se encarga de que los paquetes de datos tengan una secuencia adecuada y de controlar los errores independientemente del tipo de red física. En el caso de que el servicio sea orientado a conexión, trata de ofrecer un **servicio confiable** en una red no confiable (TCP), donde por ejemplo se pueden perder paquetes.



Este nivel se corresponde con una capa del sistema operativo que atiende a los procesos y les sirve para que puedan ellos realizar una comunicación libre de errores.

Ejemplos de protocolos ISO de este nivel son: TP0, TP1, TP2, TP3 y TP4. Y para internet son: **TCP** y **UDP**.

● **Nivel de red**. Envío de paquetes de datos de cualquier origen a cualquier destino de una subred de forma fiable (sin pérdidas ni duplicados). Este nivel resuelve toda la problemática básica cuando el emisor y el receptor no están en la misma red local. Este nivel parte del supuesto de que los computadores dentro de una red se pueden comunicar sin problemas, y que hay algunos ordenadores que interconectan dos redes distintas.

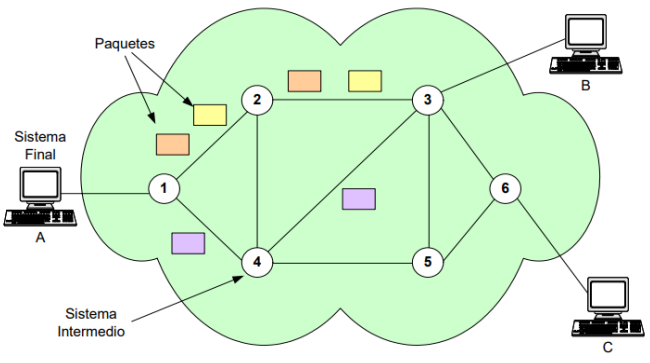
Problemas a resolver:

- **Identificación del emisor y el receptor** (y de sus redes). El emisor y receptor, pese a estar separados y ser equipos de muy diferente naturaleza (por ejemplo un móvil y un supercomputador), deben compartir una forma de identificación común para que uno reconozca al otro.

- **Encaminamiento**. El mensaje irá atravesando diferentes redes, en cada una de ellas, se debe decidir el destino inmediato para que finalmente llegue el mensaje al destinatario esté donde esté.

- **Control de la congestión**. Debe haber algún mecanismo para que emisor y receptor acuerden la cantidad de mensajes o datos a transmitir que permita al receptor tratar correctamente toda la información recibida.

- **Fragmentación**. Dado que el mensaje debe atravesar varias redes, es posible que en alguna de ellas, no pueda realizarse la transmisión del mensaje entero por cuestiones tecnológicas (es decir, porque la red no admite un paquete tan grande). El nivel de red debe resolver este problema, fragmentando el mensaje en trozos más pequeños (**paquetes**).



Este nivel se plasma en una capa del sistema operativo que gestiona la conexión básica a internet. En algunos dispositivos de red se incluye también este protocolo: **puntos de acceso**, **routers**, etc.

Ejemplos de protocolos del nivel de red son: X.25, ATM para redes de conmutación e IP para redes interconectadas.

● **Nivel de enlace de datos**. Proporciona direccionamiento físico y procedimientos de acceso a medios. Tiene como objetivo asegurar que el enlace físico de transmisión sea lo más fiable posible (una línea de

comunicación libre de errores entre dos entidades directamente conectadas). Partiendo de que ya es posible enviar bits entre dos equipos conectados, este nivel se encarga de que todo un mensaje llegue desde el emisor al receptor si ambos están en la misma red local, resolviendo los siguientes problemas:

- **Identificación de las entidades**. Identifica al emisor y receptor. Pueden haber muchos equipos compartiendo el mismo medio y cada mensaje debe especificar quién es el emisor y a quién va dirigido.
- **Entramado**. El entramado aclara cuándo empiezan y terminan los datos (**tramas**). A veces no se puede distinguir fácilmente si se está enviando bits o no. (Ej: permite distinguir jamónjamónjamón.. de monjamonjamonja,....)
- **Control de errores**. Detecta (pero no corrige) errores en la transmisión.
- **Control de flujo**. Frena o acelera los envíos de datos
- **Control de acceso al medio de transmisión** (subcapa MAC). Gestiona la disponibilidad del medio para asegurar que la transmisión no colisiona con la de otro nodo emisor. Cuando todos los computadores pueden transmitir por el mismo medio (p. ej. wifi) hay que establecer un orden y coordinación para efectuar las transmisiones de uno en uno. Se emplean las direcciones "MAC" en la cabecera de las tramas para identificar origen y destino.

En muchos casos, la Capa de enlace de datos está incorporada en una entidad física como la **tarjeta de interfaz de red** (NIC) de Ethernet, que se inserta dentro del bus del sistema de una computadora, switch o router y hace la conexión entre los procesos de software que se ejecutan en los dispositivos finales y los medios físicos. Sin embargo, la NIC no es solamente una entidad física. El software asociado con la NIC permite que la NIC realice sus funciones de intermediaria preparando los datos para la transmisión y codificando los datos como señales que deben enviarse sobre los medios asociados.

Si no se desea realizar una comunicación con equipos situados fuera de la red local, estos dos niveles son suficientes para que una aplicación o programa se conecte a otro computador. Por ejemplo, con Windows 95, sin configurar TCP/IP se podía crear una red local y compartir archivos e impresoras.

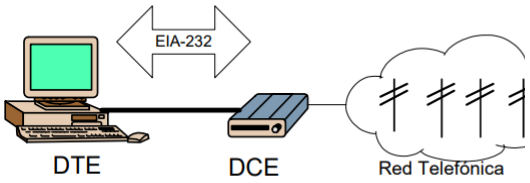
Ejemplos de protocolos del nivel de enlace son: LLC (Logical Link Control), CSMA/CD (Carrier sense multiple access / Collision detection), ALOHA, …

- **Nivel físico**. Define las reglas para el intercambio físico de bits entre dispositivos o sistemas.

Definición y estandarización de cuestiones relativas a

C/ Literat Azorín, 1 46702 – GANDIA
Tfno. 962829430 – Fax. 962829431
Codi: 46004221 – CIF: Q-9655627-I
46004221@edu.gva.es
http://iesmariaenriquez.com

- Parámetros eléctricos (voltaje representa un 1, velocidad transmisión de 1 bit,. .), magnéticos, electromagnéticos u ópticos.
- Medios de transmisión y modulación, forma de las señales a transmitir.
- Protocolos de bajo nivel empleados en transmisiones. Ej: RS232, RJ45, ...

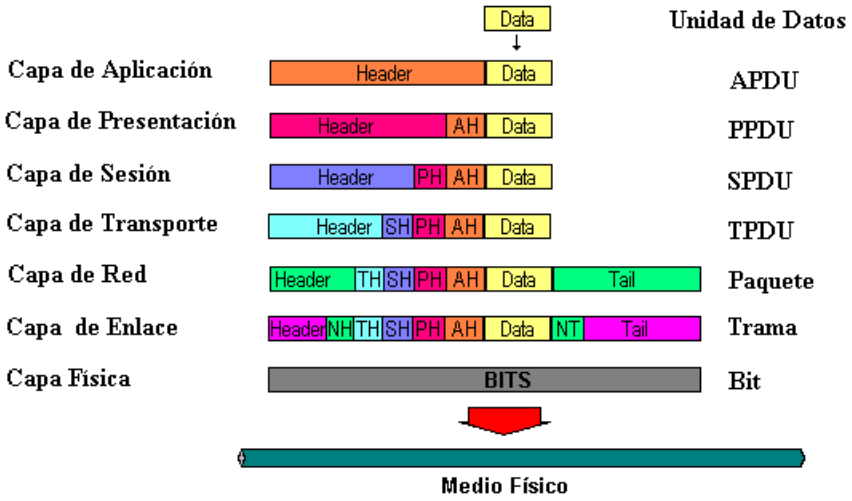


Parameter	RS232	RS485
Line configuration	Single-ended	Differential
Mode of operation	Simplex or full duplex	Simplex or half duplex
Maximum cable length	50 feet	4000 feet
Maximum data rate*	20 kbits/s	10 kbits/s
Typical logic levels	±5 to ±15 V	±1.5 to ±6 V
Minimum receiver input impedance	3 to 7 Ω	12 Ω
Receiver sensitivity	±3 V	±200 mV
*Maximum rate at maximum cable length		

- Conexionado y cableado.

Durante la comunicación entre dos entidades remotas, cada capa toma los datos de la capa inferior y añade una cabecera propia donde se insertan los datos relevantes al servicio que está proporcionando. Por ejemplo, el nivel 3 se encarga de identificar al emisor cuando el destinatario está en una red diferente. En la actualidad, eso se consigue con la dirección IP. Por tanto el nivel 3 de red añade la dirección IP en la cabecera de dicho nivel (lo que en el dibujo es "NH" (Network Header). Si estamos transmitiendo un texto en chino, entonces la cabecera del nivel 6 o de presentación, indicará que los caracteres contenidos en los datos debe interpretarse como caracteres chinos.

Durante la comunicación entre un emisor y un receptor, cada capa interactúa con su equivalente en el extremo contrario. Así, cada capa añade una cabecera al mensaje en el origen. Dicha cabecera contiene la información de control relativa a su capa correspondiente y será interpretada en el destino por la capa correspondiente. Este mecanismo recibe el nombre de ***encapsulación***.



El conjunto que forma la cabecera y la información de una capa y que pasa a la capa inferior, recibe el nombre de ***Unidad de Datos de Protocolo*** o **PDU** (del inglés *Protocol Data Unit*).

Como puedes observar en la imagen siguiente, en la capa 2 también se añade una cola al mensaje para controlar el final de la transmisión. La capa 1 no añade información y se limita a transmitirla bit a bit.

Cómo transitan los datos a través del modelo OSI

Para que la información legible para los seres humanos se pueda transferir a través de una red de un dispositivo a otro, los datos deben atravesar las siete capas del modelo OSI en orden descendente en el dispositivo emisor y luego en orden ascendente en el extremo del receptor.

Por ejemplo, el señor Cooper quiere enviar a la señora Palmer un correo electrónico. El señor Cooper redacta dicho mensaje en una aplicación de correo y después le da a enviar. Su aplicación de correo pasa entonces su mensaje a la capa de aplicación, y ésta elige un protocolo (SMTP) y pasa los datos a la capa de presentación. La capa de presentación comprime entonces los datos y los pasa a la capa de sesión, que será la que inicie la sesión de comunicación.

Los datos llegarán entonces a la capa de transporte del emisor y serán allí segmentados. Después, estos **segmentos** serán rotos en trozos más pequeños, **paquetes**, en la capa de red y en trozos aún más pequeños, **tramas**, en la capa de enlace de datos. Entonces la capa de enlace de datos enviará las tramas a la capa física para que puedan ser convertidas por esta en una secuencia de bits formada por unos y ceros que viaje a través de un medio físico, por ejemplo, un cable.

Cuando el ordenador de la señora Palmer reciba la secuencia de bits a través de un medio físico (por ejemplo, su wifi), los datos viajarán a través de la misma serie de capas, solo que ahora en su dispositivo y en orden inverso. Primero, la capa física convertirá la secuencia de bits en tramas que pasarán a la capa de enlace de datos. Segundo, esta capa ensamblará las tramas para formar paquetes que pueda utilizar la capa de red. Tercero la capa de red creará segmentos a partir de tales paquetes y los enviará a la capa de transporte. Por último, la capa de transporte convertirá tales segmentos en trozos de información.

Los ahora ya datos pasarán a la capa de sesión del receptor, y ésta, a su vez, los hará llegar a la capa de presentación; después pondrá fin a la sesión de comunicación. La capa de presentación eliminará entonces la compresión y pasará los datos brutos a la capa de aplicación. Por último, la capa de aplicación suministrará datos legibles por humanos al software de correo de la señora Palmer a fin de que esta persona pueda leer en la pantalla de su portátil el correo del señor Cooper.

1.2. El modelo TCP/IP

El *modelo TCP/IP* (formado por una gran variedad de protocolos, siendo los más importantes TCP/IP de los que adopta su nombre) es un estándar abierto con un planteamiento muy parecido al *modelo de referencia OSI*. El ***modelo TCP/IP*** es el que se utiliza de forma generalizada en la actualidad.

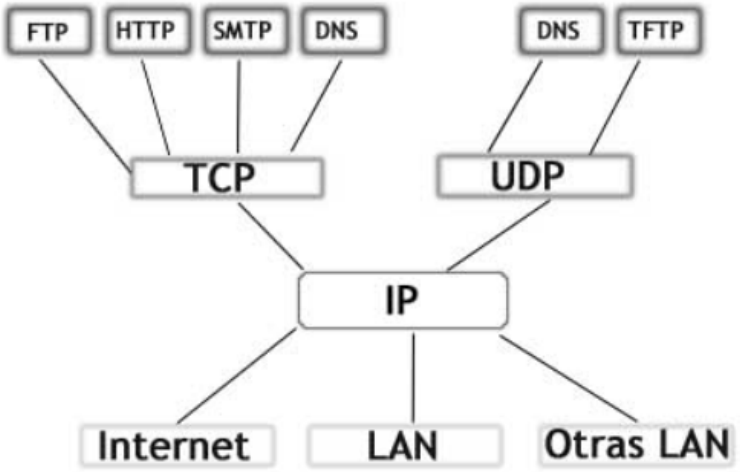
TCP/IP cuenta con 4 capas: **Aplicación, Transporte, Internet y Acceso a la red**, agrupando las capas Aplicación, Presentación y Sesión del modelo OSI en una única capa y la de Enlace y Físico en la de Acceso a la red del modelo TCP/IP.

Sin embargo, este modelo no es muy aceptable desde el punto de vista del diseño de redes, sobre todo porque no se distinga claramente entre la capa física y la de enlace de datos, por lo que la realidad es que se utiliza como modelo de referencia el modelo OSI y como protocolos se utilizan los de TCP/IP.

1.2.1. La capa de aplicación

Esta capa desarrolla las funciones de las **capas de sesión, presentación y aplicación del modelo OSI**. Maneja aspectos de representación, codificación y control del diálogo. La capa de aplicación contiene todos los protocolos de alto nivel que se utilizan para ofrecer servicios a los usuarios como **telnet**, **FTP, SMTP, DNS, HTTP**, etc…

Ejemplos de Protocolos de la capa de Aplicación TCP/IP:



1.2.2. La capa de transporte

Esta capa recibe el mismo nombre y desarrolla la misma función que la **cuarta capa del modelo OSI**, encargada de controlar la comunicación extremo a extremo (host a host) en la red.

Proporciona un servicio de transferencia de datos garantizado entre sistemas finales, ocultando detalles de la red. Aquí se definen dos protocolos: el **TCP** (Transmission Control Protocol) ofrece un servicio orientado a conexión fiable, con lo que los paquetes (aquí llamados segmentos) llegan ordenados y sin errores. **TCP** se ocupa también del control de flujo extremo a extremo, para evitar que por ejemplo un host rápido sature a un receptor más lento. Ejemplos de protocolos de aplicación que utilizan TCP son el **SMTP** (Simple Mail Transfer Protocol) y el **FTP** (File Transfer Protocol).

Los **puertos** (Source Port y Destination Port) son fundamentales para el buen funcionamiento de TCP. TCP usa estos números de puertos para identificar un socket, es decir, una aplicación que emite datos o que recibe datos. Los puertos TCP van desde el 0 hasta el 65535, pero tenemos tres tipos de puertos diferentes:

Puertos conocidos: del 0 al 1023. Estos puertos están reservados por la IANA para determinadas aplicaciones, como servidor HTTP(80), FTP(21), SSH(22), HTTPS(443),..

Puertos registrados: de 1024 al 49151. Estos puertos están reservados para aplicaciones concretas, como sistemas gestores de bases de datos, BitTorrent, y muchas otras aplicaciones.

Puertos privados: de 49152 a 65535. Estos puertos no están reservados por ninguna aplicación, y puedes usarlos libremente sin que afecte a ningún otro protocolo.

El otro protocolo de transporte es **UDP** (User Datagram Protocol) que da un servicio no orientado a conexión, no fiable. **UDP** no realiza control de errores ni de flujo. Una aplicación típica donde se utiliza UDP es la transmisión de voz y vídeo en tiempo real; aquí el retardo que introduciría el control de errores produciría más daño que beneficio: es preferible perder algún paquete que retransmitir fuera de tiempo…

1.2.3. La capa de internet

Esta capa equivale a la **capa de red en el modelo OSI**, es decir, se ocupa de encaminar los paquetes de la forma más conveniente para que lleguen a su destino, y de evitar que se produzcan situaciones de congestión en los nodos intermedios. Esta capa solo proporciona un servicio de conmutación de paquetes[2] no orientado a conexión, ya que exige un control de errores que haría ineficiente una comunicación orientada a conexión. Los paquetes pueden llegar desordenados a su destino, en cuyo caso es responsabilidad de las capas superiores en el nodo receptor la reordenación para que sean presentados al usuario de forma adecuada.

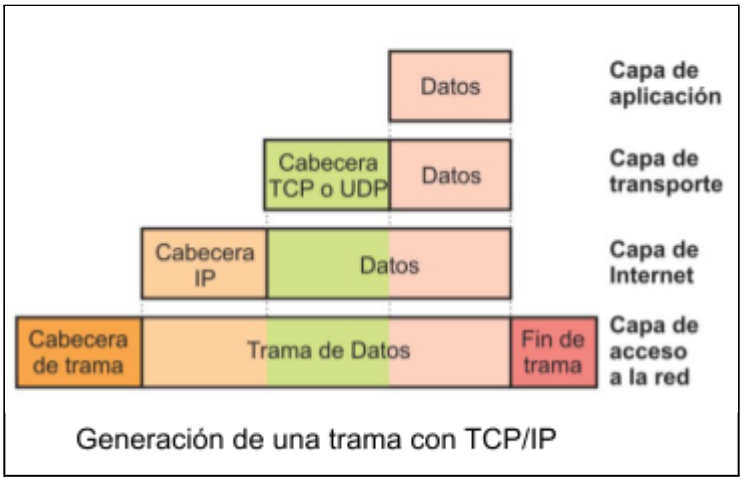
[2] método de agrupar los datos transmitidos a través de una red digital en paquetes

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

Esta capa incluye a su vez, el protocolo de resolución de direcciones (**ARP**[3]) y el protocolo de mensajes de control de Internet (**ICMP**[4]).

1.2.4. La capa de acceso a la red

Esta capa se corresponde con las **capas física y de enlace de datos del modelo OSI**. En este caso, el modelo TCP/IP no detalla demasiado estos niveles, ya que deja bastante libertad a los diseñadores de protocolos para escoger el medio físico, con tal que manejen paquetes IP. Eso es debido a que, al tratarse de una red de redes, cada subred se basa en protocolos propios para el medio físico (por ejemplo, **ATM**[5]). Gracias a este diseño, podemos transmitir paquetes IP por cables eléctricos, infrarrojos, etc.



Generación de una trama con TCP/IP

En resumen, para poder conectar cualquier dispositivo a la red es necesario utilizar el protocolo de red adecuado. Este protocolo permitirá enviar datos a cualquier dispositivo conectado a la red, directamente o a través de un dispositivo de encaminamiento en el caso de que el destino se encuentre en otra red.

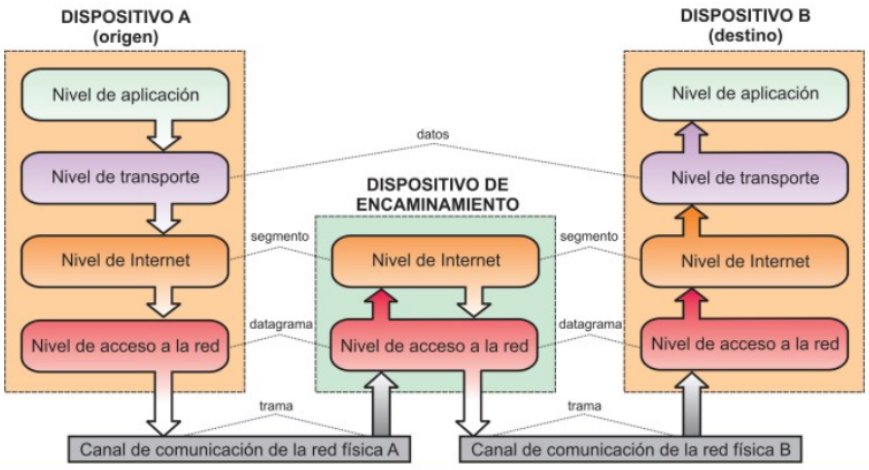
[3] protocolo de resolución de direcciones, para encontrar la dirección física (MAC) correspondiente a una determinada IP.
[4] Internet Control Message Protocol
[5] Asynchronous Transfer Model

Para este último caso, el protocolo de capa de internet proporciona todos los mecanismos necesarios para transportar los datos desde el equipo origen hasta el equipo destino, atravesando todos los dispositivos de encaminamiento necesarios. Por tanto, esta capa debe estar implementada en todos los dispositivos de la red, tanto equipos finales como dispositivos de encaminamiento. Sin embargo las capas de transporte y aplicación únicamente estarán implementadas en los equipos finales para garantizar que los datos recibidos son correctos y ofrecérselos de forma ordenada a las aplicaciones correspondientes.

Por ejemplo, si un dispositivo A desea transmitir datos a un dispositivo B, el **nivel de aplicación** solicitará un servicio de transferencia de datos a una entidad de la capa de transporte. En dicha solicitud se suministrarán los datos a enviar, así como el nombre de la aplicación que los generó y a qué aplicación en destino van dirigidos. La **entidad de transporte** con toda esa información generará el bloque a transmitir, denominado **segmento**. (Si el segmento fuese muy grande para transmitirlo de una sola vez, la entidad de transporte lo dividiría en trozos, añadiéndole a cada uno cierta información de control y enviándolos por separado).

Una vez generado el segmento, la entidad de transporte solicitará un servicio a una entidad de la capa de Internet. Al igual que anteriormente, en dicha solicitud se pasa el segmento a transmitir y cierta información necesaria, como por ejemplo el dispositivo destino. La **entidad de Internet** con estos datos y otros que añadirá, como por ejemplo la dirección IP del dispositivo destino, creará un nuevo bloque de datos llamado **datagrama o paquete**. La entidad de la capa de Internet pasará el datagrama a una **entidad de la capa de acceso a red**, que la transformará de forma adecuada para que pueda ser enviada a través del canal de comunicación, generando una o varias **tramas**.

La trama será recibida por el dispositivo destino directamente o a través de dispositivos de encaminamiento. En ambos casos, la **entidad par de la capa de acceso a la red** leerá la trama y realizará las tareas definidas en el protocolo usando la información de control añadida al datagrama. Si la trama no se desecha, dicha entidad pasará el datagrama a la **entidad par del nivel de Internet**, donde se procesará con la información de control añadida en el dispositivo origen.



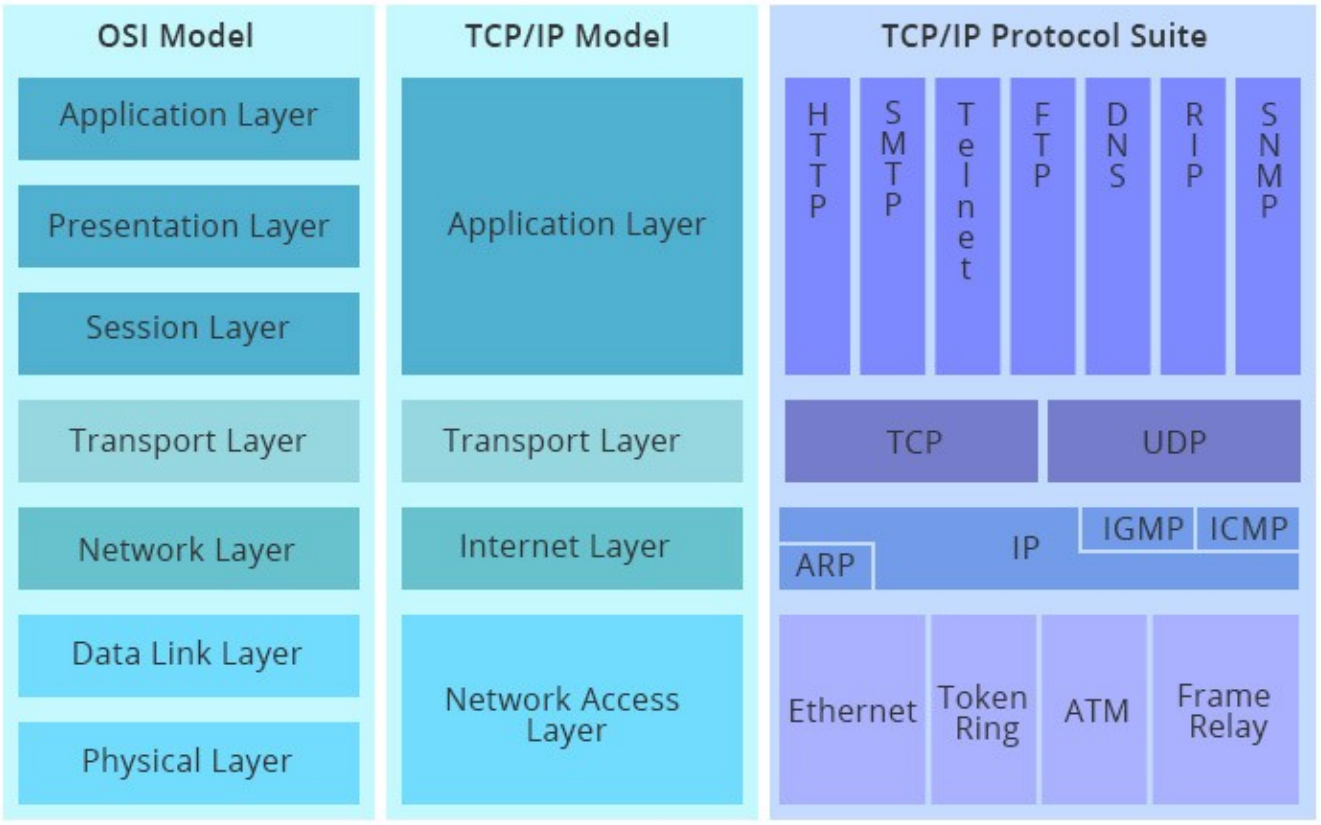
I. Si se trata de un dispositivo de encaminamiento

empleará esta información para trazar el camino hacia el destino. Con esta información generará un nuevo datagrama que será enviado empleando los servicios del nivel de acceso a la red. La trama emitida viajará por el nuevo canal de comunicación, pasando por tantos dispositivos de encaminamiento como sean necesarios hasta llegar al dispositivo destino.

II. Si por el contrario se trata de un dispositivo destino (dispositivo B), la **entidad par de la capa de Internet** pasará el segmento a una **entidad de la capa de transporte**. Esta entidad separará la información de control y los datos. Con la información de control se realizarán las tareas definidas en el protocolo y, si todo es correcto, pasarán los datos a la aplicación destino. Si los datos enviados fueron divididos en trozos, la entidad par de la capa de transporte esperará a tener todos los fragmentos para ordenarlos y pasarle el bloque completo a la aplicación destino.

1.3. Correspondencia entre el modelo OSI y TCP/IP

Si tenemos en cuenta los significados de los dos modelos de referencia, el **modelo OSI** sería solo un modelo conceptual; este se utiliza principalmente para describir, discutir y comprender funciones de



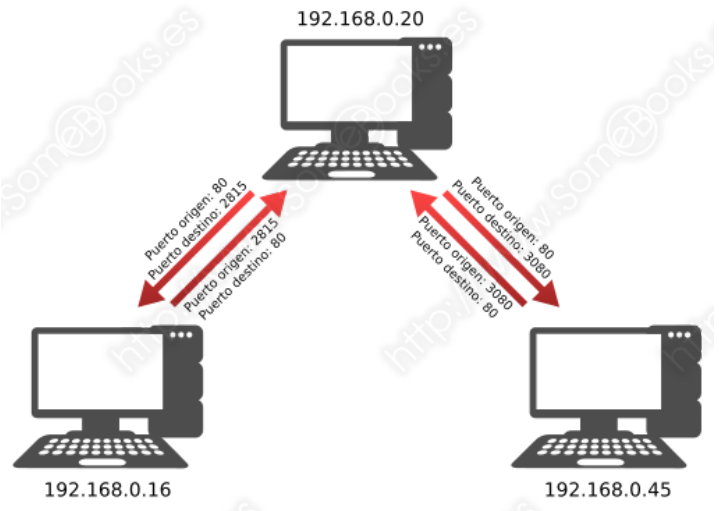
red individuales. Sin embargo, **TCP/IP** está diseñado para resolver un conjunto específico de problemas y no para funcionar como una descripción de generación para todas las comunicaciones de red, tal y como lo hace el modelo OSI. El **modelo OSI** es genérico e independiente del protocolo, aunque la mayoría de los protocolos y sistemas se adaptan a él; mientras que el **modelo TCP/IP** se basa en protocolos estándar desarrollados por Internet.

Otro factor a tener en cuenta en el modelo OSI es que, para las aplicaciones más simples, no todas las capas son utilizadas. Si bien las capas 1, 2, 3 son obligatorias para cualquier comunicación de datos, también existen aplicaciones que pueden usar ciertas capas de interfaz específicas en lugar de las capas superiores habituales del modelo.

2. Conexión de sistemas en red

Para identificar a cada ordenador de la red, el *protocolo IP* del *modelo TCP/IP* utiliza *direcciones IP*.

Existe un organismo responsable de asignar *direcciones IP* a los ordenadores que se conectan directamente a *Internet*. Su nombre es *ICANN* (*Internet Corporation for Assigned Names and Numbers*, en español, *Corporación de Internet para la Asignación de Nombres y Números*).



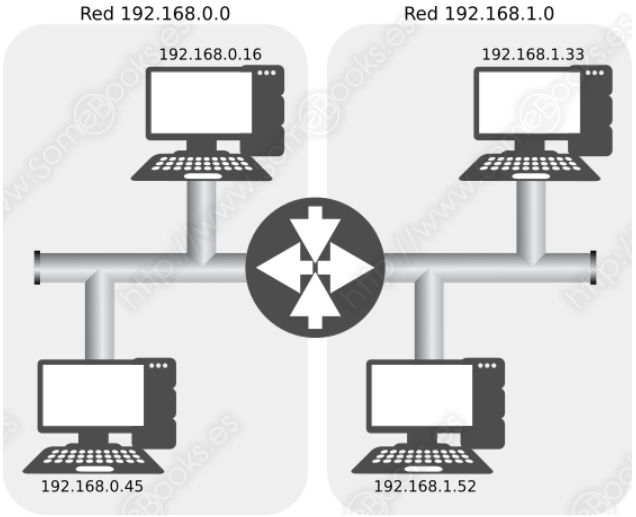
Cada *dirección IP* está formada por 4 bytes, que representan cuatro números enteros, sin signo, con valores comprendidos entre 0 y 255. Estos valores se escriben separados por un punto y sin dejar espacios entre ellos. Por ejemplo, las *direcciones IP* 192.168.0.16, 192.168.0.20 y 192.168.0.45.

Cada equipo debe tener una *dirección IP* única dentro de la misma red.

En cuanto a la estructura de la dirección IP, sus 32 bits se dividen en dos partes:
- Un primer subconjunto a la izquierda, que identifica la red, llamado ***netID***.
- El resto de los bits a la derecha identifican a cada ordenador dentro de la red y se llama ***hostID***.

Lógicamente, cuanto mayor sea el número de bits reservados a la identificación de la red, menor será el número de equipos que pueden formarla. Sin embargo, el uso de los identificadores de red (*netID*) nos permite establecer una jerarquía de subredes.

Clases de redes



En función del número de bits que se utilicen para representar la parte correspondiente a la red y la correspondiente a los equipos dentro de la configuración de una red, las redes se dividen según la siguiente clasificación:

- ***Clase A***: Sólo se utiliza el primer octeto (byte) para identificar la red. Los 24 restantes se utilizan para identificar equipos en la red. Además, el primer bit estará siempre a cero, lo que significa que, en realidad, sólo se utilizan 7 bits para identificar redes. Es decir, el máximo número teórico de redes en esta clase serían 128 (de 0 a 127), pero como el valor 127 está reservado (lo veremos más abajo) y la red 0 no existe, sólo nos quedan 126 posibilidades (de 1 a 126).

Esta clase se utiliza en implementaciones con un número pequeño de redes y un gran número de equipos en cada red.

- ***Clase B***: Emplea 16 bits para el *netID* y los otros 16 para los equipos. Además, los primeros dos bits tienen siempre el valor 10, por lo que únicamente nos quedan 14 bits para identificar redes. Es decir, 16384.

Se utiliza para redes que se encuentran a medio camino entre las de clase A y las de clase C.

- ***Clase C***: Pensada para redes pequeñas, utiliza los primeros 24 bits para identificar la red y sólo 8 para los equipos que la forman. Además, los primeros tres bits tienen siempre el valor 110, por lo que nos quedan 21 bits para identificar redes. Esto nos arrojaría, siguiendo los cálculos anteriores, 2.097.152 redes diferentes.
- ***Clase D***: Utilizan sólo los primeros 4 bits para identificar la red, que siempre valen 1110, y se emplean para multidifusión de contenidos.
- ***Clase E***: Emplea únicamente los 4 primeros bits para la identificación de la red, que siempre valen 1111, y está destinada a uso en investigación.

Clases de redes					
Clase	Formato	Intervalo	Redes	Equipos	Aplicación
A	0xxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx	1.0.0.0 126.0.0.0	126	16777214	Redes grandes
B	10xxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx	128.0.0.0 191.255.0.0	16384	65534	Redes medianas
C	110xxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx	192.0.0.0 223.255.255.0	2097152	254	Redes pequeñas
D	1110xxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx	224.0.0.0 239.255.255.255	-	-	Multicast
E	1111xxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx	240.0.0.0 254.255.255.255	-	-	Investigación

Direcciones IP especiales

Cuando hablamos de *direcciones IP*, existen algunos valores que tienen un significado especial:

- ***Dirección de red***: Es una *dirección IP* donde todos los bits destinados a identificar equipos aparecen con el valor 0. Por eso, en la imagen de la página anterior, identificábamos las dos redes como 192.168.0.0 y 192.168.1.0.El ejemplo corresponde con una red de *clase C*.

- ***Dirección del equipo***: Al contrario de la anterior, es cuando todos los bits que identifican la red se ponen a cero.

- ***Dirección de difusión (broadcast)***: Se obtiene poniendo a 1 todos los bits que identifican al equipo. Se trata de una *dirección IP* especial que se utiliza para enviar un mensaje a todos los equipos que formen parte de la red. Por ejemplo, en la red 192.168.0.0, la dirección de difusión sería 192.168.0.255.

- ***Dirección de bucle de retorno* (*loopback*)**: Hace referencia al propio equipo en el que nos encontramos. Su valor es 127.0.0.1.

Por otro lado, la mayor parte de las veces, los equipos de una red no tienen acceso directo a *Internet* y acceden a ésta a través de otro ordenador, que actúa como proxy, o de una pasarela.

Para estos casos, el único dispositivo que necesita disponer de una *dirección IP* asignada por la *ICANN* es el que actúa como pasarela. Los demás, harán uso de un conjunto de direcciones de ámbito local, reservadas por la *ICANN* para este tipo de situaciones. De este modo, no se producirán conflictos con las *direcciones IP* utilizadas en *Internet*. Estas direcciones se encuentran reflejadas en la siguiente tabla:

Direcciones IP reservadas		
Clase	Valor Inicial	Valor fimal
A	10.0.0.1	10.255.255.254
B	172.16.0.1	172.31.255.254
C	192.168.0.1	192.168.0.254

3. Máscara de red

Una *máscara de red* está formada por 32 bits, agrupados de ocho en ocho, con una construcción similar a una *dirección IP*. Sin embargo, a diferencia de ésta, en una máscara de red, todos los bits que estarían destinados a la identificación de red (*netID*) tendrían el valor 1 y los destinados a la identificación del equipo (*host-ID*) tendrían el valor 0.

Por lo tanto, en función de la *clase de red*, tendríamos las siguientes *máscaras de red*:

Máscaras de red		
Clase	Binario	Decimal
A	11111111.00000000.00000000.00000000	255.0.0.0
B	11111111.11111111.00000000.00000000	255.255.0.0
C	11111111.11111111.11111111.00000000	255.255.255.0

C/ Literat Azorín, 1 46702 – GANDIA
Tfno. 962829430 – Fax. 962829431
Codi: 46004221 – CIF: Q-9655627-I
46004221@edu.gva.es
http://iesmariaenriquez.com

Función de una máscara de red

Las ***máscaras de red*** permiten separar, de manera sencilla, la parte de una *dirección IP* que identifica la red a la que pertenece. Para lograrlo, basta con aplicar una operación lógica *Y* (*AND*) entre la *dirección IP* y su máscara correspondiente. El resultado será la ***dirección de red***.

En la siguiente imagen tienes un ejemplo en el que se aplica una máscara de red para una *dirección IP* de *clase A*:

	Decimal	Binario	
Dirección IP	120.140.3.48	01111000.10001100.00000011.00110000	AND
Máscara de red	255.0.0.0	11111111.00000000.00000000.00000000	
Dirección de red	120.0.0.0	01111000.00000000.00000000.00000000	

Creación de subredes

Cuando una red crece en exceso, podemos hacerla más manejable utilizando *subredes*. De esta forma, se puede controlar el tráfico entre las diferentes subredes y se reduce el ámbito de las operaciones de *broadcast*.

Broadcast es un tipo de transmisión por la que un determinado equipo puede enviar información al resto de los equipos de su red de forma simultánea. Es decir, en una sola operación, en lugar de hacerlo uno a uno.

Para crear ***subredes*** dentro de una red, sólo tenemos que recurrir a la ***máscara de red***, añadiendo a la dirección de red tantos bits como necesitemos.

Por ejemplo, supongamos una red de *clase B* con la dirección 172.16.0.0 y una máscara de subred 255.255.0.0. Para dividir esta red en cuatro posibles subredes, bastaría con tomar los dos primeros bits (por la izquierda) del tercer byte.

Es decir, pasamos de utilizar esta máscara 11111111.11111111.00000000.00000000 (255.255.0.0), a utilizar esta 11111111.11111111.11000000.00000000 (255.255.192.0)

Así, la dirección 172.16.6.120 podría pertenecer a un ordenador de la primera *subred*, la dirección 172.16.70.120 pertenecería a un ordenador de la segunda *subred*, 172.16.134.120 haría referencia a un ordenador de la tercera *subred* y, finalmente, 172.16.198.120 se referiría a un ordenador de la cuarta *subred*. Observa el desglose de los cálculos en la siguiente tabla:

	Decimal	Binario	
Dirección IP	172.16.6.120	10101100.00010000.00000110.01111000	AND
Máscara de red	255.255.192.0	11111111.11111111.11000000.00000000	
Dirección de red	172.16.0.0	10101100.00010000.00000000.00000000	

C/ Literat Azorín, 1 46702 – GANDIA
Tfno. 962829430 – Fax. 962829431
Codi: 46004221 – CIF: Q-9655627-I
46004221@edu.gva.es
http://iesmariaenriquez.com

Como cabe esperar, cada una de estas cuatro *subredes* admitirá 16382 ordenadores diferentes (2^{14}-2). Los dos equipos que perdemos en cada *subred* equivalen a la dirección 0 (que coincide con la dirección de red) y a la dirección más elevada (la que tiene todos los bits de la parte *host-ID* a uno), que es la *dirección de difusión* (*broadcast*).

A modo de resumen, la siguiente tabla muestra las direcciones válidas en cada una de las *subredes* del ejemplo anterior:

Construir la tabla anterior puede ser relativamente sencillo. Observa que la dirección de *broadcast* de una *subred* se puede obtener restando uno a la dirección de la siguiente *subred*. Además, el intervalo de direcciones puede obtenerse sumándole uno a la dirección de red y restándole uno a la dirección de *broadcast*.

Dirección de red	Rango de direcciones para equipos	Dirección de broadcast
172.16.0.0	172.16.0.1 - 172.16.63.254	172.16.63.255
172.16.64.0	172.16.64.1 - 172.16.127.254	172.16.127.255
172.16.128.0	172.16.128.1 - 172.16.191.254	172.16.191.255
172.16.192.0	172.16.192.1 - 172.16.255.254	172.16.255.255

En la siguiente tabla, puedes obtener una guía rápida para encontrar la equivalencia entre los valores binarios y decimales en los diferentes octetos de una máscara de red:

Decimal	Binario
0	00000000
128	10000000
192	11000000
224	11100000
240	11110000
248	11111000
252	11111100
254	11111110
255	11111111

C/ Literat Azorín, 1 46702 – GANDIA
Tfno. 962829430 – Fax. 962829431
Codi: 46004221 – CIF: Q-9655627-I
46004221@edu.gva.es
http://iesmariaenriquez.com

ACTIVIDADES

Actividad resuelta 1: ¿Podemos utilizar el valor 255.255.64.0 como máscara de subred? Razona el motivo de tu respuesta.

Como hemos explicado al principio de este apartado, todos los bits que están destinados a la identificación de red (netID) deben tener el valor 1.

Sin embargo, si convertimos la máscara anterior a binario, obtenemos el valor **11111111.11111111.01000000.00000000** y, como vemos, el primer bit del tercer octeto no tiene el valor 1.

Por lo tanto, el valor anterior no es correcto.

Actividad resuelta 2: Si en una infraestructura de red de clase A utilizamos una máscara de *subred* como esta: **255.224.0.0**, ¿Cuántas *subredes* diferentes podremos implementar en dicha infraestructura? ¿Y cuántos ordenadores podremos utilizar en cada una de ellas?

Si convertimos la máscara de *subred* a binario, obtenemos el valor **11111111.11100000.00000000.00000000**. De este modo, comprobamos que se utilizan **tres** bits del segundo octeto para implementar *subredes*. Esto significa que podremos crear un máximo de 8 subredes (2^3).

Como nos quedan 21 bits para direccionar ordenadores, podremos disponer de un total de 2097150 (2^{21}-2) equipos en cada subred.

Actividad resuelta 3: Realiza una tabla donde expreses el número de *subredes* que podemos crear en función del número de bits empleados para crear la máscara de subred.

A continuación tenemos el número de subredes según el número de bits utilizados para crearlas:

Número de bits	Número de subredes
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	Imposible, sería la siguiente clase

Notación simplificada

La *IETF* (*Internet Engineering Task Force*, en español, *Grupo de Trabajo de Ingeniería de Internet*) es una organización internacional dedicada a establecer estándares de Internet.

En 1993, la *IETF* introdujo un modo de notación simplificada para las *máscaras de red*, a la que llamó *CIDR* (Classless Inter-Domain Routing, en español, *Enrutamiento Entre Dominios Sin Clases*).

Esta notación propone escribir la dirección IP seguida de una barra inclinada (*slash*) y un valor que representa el número de unos consecutivos que tiene la *máscara de red* por la izquierda.

Así, por ejemplo, si la configuración de un equipo dispone de una *dirección IP* 192.168.0.32 y una *máscara de red* 255.255.255.0, también podríamos representarla, de forma resumida, como 192.168.0.32/24.

En la siguiente tabla incluimos las equivalencias entre los valores de máscaras de red representadas con sus valores decimales y su equivalencia con notación CIDR:

Actividad resuelta 4: Responde a las siguientes preguntas:
¿Qué tipo de dirección es 192.168.1.17/28?
¿Cuál es la dirección de red?
¿Cuál es la dirección de broadcast?
¿Cuántos ordenadores podremos conectar en esa subred?

- Como la máscara de red utiliza 28 dígitos, tendrá el siguiente aspecto en binario: 11111111.11111111.11111111.11110000. Es decir, 255.255.255.240 en decimal. En definitiva, pertenece a una red de *clase C* con 16 subredes (24).
- Para obtener la dirección de red, combinamos la dirección IP con la máscara de subred con el operador Y (AND):

11000000.10101000.00000001.00010001 (IP)
11111111.11111111.11111111.11110000 (Máscara)
11000000.10101000.00000001.00010000 (Dirección de red)

Por lo tanto, la dirección de red es 192.168.1.16

- La dirección de *broadcast* será la última *dirección IP* válida para la subred. Es decir 192.168.1.31
- Como disponemos de 4 bits para direccionar equipos, podríamos incluir hasta 14 equipos (2^4-2), ya que no podemos utilizar la primera (que coincide con la dirección de red), ni la última (que es la *dirección de difusión*).

C/ Literat Azorín, 1 46702 – GANDIA
Tfno. 962829430 – Fax. 962829431
Codi: 46004221 – CIF: Q-9655627-I
46004221@edu.gva.es
http://iesmariaenriquez.com

COMPUTER SYSTEMS
UD6: NETWORKS INTRODUCTION_II

Clase	Decimal	CIDR
Clase A	255.0.0.0	/8
	255.128.0.0	/9
	255.192.0.0	/10
	255.224.0.0	/11
	255.240.0.0	/12
	255.248.0.0	/13
	255.252.0.0	/14
	255.254.0.0	/15
Clase B	255.255.0.0	/16
	255.255.128.0	/17
	255.255.192.0	/18
	255.255.224.0	/19
	255.255.240.0	/20
	255.255.248.0	/21
	255.255.252.0	/22
	255.255.254.0	/23
Clase C	255.255.255.0	/24
	255.255.255.128	/25
	255.255.255.192	/26
	255.255.255.224	/27
	255.255.255.240	/28
	255.255.255.248	/29
	255.255.255.252	/30

4. El protocolo TCP/IPv6

Hasta este punto de la documentación, cada vez que hemos hablado del protocolo IP nos referíamos a IPv4, que se implementó en 1983 para su uso en la red ARPANET.

En aquella época, los 4294967296 (232) direcciones únicas parecían más que suficientes, pero después del crecimiento que ha sufrido *Internet* desde entonces, han sido completamente insuficientes. Sobre todo por varios motivos:

- Porque muchas de ellas están reservadas para redes locales
- Porque en la actualidad cada usuario individual puede tener múltiples dispositivos conectados a *Internet* (ordenadores, teléfonos, tablets, etc.)
- Porque en el futuro inmediato, con la generalización del *Internet de las cosas*, se prevé que los tipos de dispositivos conectados se diversifique mucho más (vehículos, televisores y otros muchos dispositivos domésticos)

Cómo se forma una dirección IPv6

Una dirección IPv4 está formada por cuatro grupos de 8 bits cada uno.

Para resolver esta situación, se diseñó la versión IPv6, que propone direcciones de 128 bits. Es decir, podremos obtener hasta 340.282.366.920.938.463.463.374.607.431.768.211.456 (2^{128}) direcciones diferentes.

Para escribir una dirección IPv6 se utilizan 8 campos de 16 bits separados por dos puntos. Por otro lado, en lugar de la notación decimal que suele utilizarse en IPv4, en IPv6 se emplea notación hexadecimal para representar cada uno de los campos. Además, se utilizan dos puntos (:) para separar un campo del siguiente. Así, una dirección TCP/IPv6 tendría el siguiente aspecto:

2001:0db8:ac10:0013:0000:0000:2b4e:0c11

Notación simplificada

Como, a pesar de todo, las direcciones IPv6 tienden a ser difíciles de manejar, existen formas abreviadas de escribir algunas de ellas:

- Si en la dirección IPv6 tenemos campos cuyo valor es cero, podemos representarlos con un único cero en lugar de cuatro. Por ejemplo, la dirección anterior, podríamos escribirla como 2001:0db8:ac10:0013:0:0:2b4e:0c11
- Incluso podemos ir más allá y eliminar el campo por completo: 2001:0db8:ac10:0013:::2b4e:0c11

Sin embargo, esto no podemos hacerlo dos veces en la misma dirección. Es decir, si la dirección original fuese 2001:0db8:ac10:0000:0000:0013:0000:0000, podríamos eliminar completamente uno de los dos

bloques de ceros, pero no ambos.
Por lo tanto, serían válidas las siguientes abreviaturas: 2001:0db8:ac10::0013:0:0 y 2001:0db8:ac10:0:0:0013::.
Si el número de campos consecutivos que se encuentran a cero son más de dos, la abreviatura sería la misma. Es decir, si la dirección original fuese 2001:0db8:0000:0000:0000:0000:2b4e:0c11, podríamos escribir 2001:0db8::2b4e:0c11

- También pueden omitirse los dígitos a la izquierda de un campo cuando su valor es cero. Además, podemos utilizarlo como complemento a cualquiera de los métodos anteriores. Por ejemplo: 2001:db8::2b4e:c11

Partes de una dirección IPv6

Desde un punto de vista lógico, las direcciones IPv6 se dividen en tres partes:

- *Prefijo del sitio*: Ocupa, como máximo, los tres primeros campos (48 bits) y forman la parte pública de la dirección. Suele expresarse en notación *CIDR*. Así, cuando escribimos 2001:db8:ac10:13:0:0:2b4e:c11/48 estaremos indicando que el prefijo ocupa los tres primeros campos.

Si queremos referirnos únicamente al prefijo, podríamos utilizar la notación con ceros comprimidos: 2001:db8:ac10::/48

- La *dirección MAC* es un identificador único de cada dispositivo de red.

Utiliza 48 bits. Los primeros 24 los asigna el fabricante y el resto el IEEE.
Prefijo de subred: suele ocupar el cuarto campo (16 bits) y le permite al *enrutador* (*router*) identificar la topología interna de la red.
Para expresar la subred suele emplearse también notación CIDR. Por ejemplo: 2001:db8:ac10:13::/64
La suma entre los bits del *prefijo del sitio* y los del *prefijo de subred* será siempre 64.

- *ID de la interfaz* : Serán los cuatro últimos campos (64 bits). Suele llamarse token y su valor puede asignarse manualmente o provenir de la *dirección MAC* (del inglés, *Media Access Control*) de la tarjeta de red.

Configuración de Ipv6

Existen tres formas diferentes de configurar un equipo para el uso de Ipv6:

- *Manual*: El administrador del equipo introduce manualmente los valores de confiugración para el protocolo TCP/IPv6
- *Autoconfiguración* (también conocida como *Configuración Automática de Dirección Sin Estado Ipv6*): El equipo buscará en la red un enrutador IPv6 que le devuelve el prefijo de subred. A continuación, el equipo añade su dirección de capa de enlace (*dirección MAC*) en formato *EUI-64 Modificado*.
- *Mediante un servidor* (*Configuración de Direcciones con Estado IPv6*): Usando un servidor *DHCP* de la red.

El formato *EUI-64 modificado* utiliza los 48 bits de la d*irección MAC* para crear el *ID de la interfaz* en la dirección IPv6.

Dado que el *ID de la interfaz* emplea 64 bits, los 16 bits que faltan se completan con el valor FF:FE. Este valor se inserta en el centro de la *dirección MAC*, de modo que, si la dirección original fuese 3a:87:b0:47:22:11, el *ID de la interfaz* para la *dirección IPv6* sería 3a87:b0ff:fe47:2211.

Si el *ID de la interfaz* va a formar parte de una dirección IPv6, también se invierte su séptimo bit por la izquierda (llamado bit *Universal/Local*). Así, el *ID* resultante sería 3887:b0ff:fe47:2211. De esta forma, se disminuyen las posibilidades de que coincida con una dirección asignada manualmente en otro equipo de la red.

El valor hexadecimal **A(16** equivale a **1010(2** en binario. Por su parte, el valor hexadecimal **8(16** equivale a **1000(2** en binario.

Direcciones IPv6 con direcciones IPv4 incrustadas

Es posible combinar direcciones IPv6 y direcciones IPv4 de forma que éstas últimas se incrusten en las primeras.

Para lograrlo, la dirección IPv6 se divide en dos partes:

- La primera sigue utilizando notación hexadecimal y representa los 6 primeros campos de la dirección.
- La segunda (el segmento IPv4) tiene cuatro campos y utiliza notación decimal con valores de 8 bits.

Así, se asegura la compatibilidad de los equipos que aún funcionen con una configuración IPv4 y los que ya dispongan de configuración IPv6 dentro de la misma red. De este modo, los dispositivos de red que trabajen con IPv6 representarán las direcciones de los dispositivos IPv4 como direcciones IPv6

Veamos un ejemplo: 0:0:0:0:0:ffff:192.1.1.25

También podemos escribirlo en formato abreviado así: ::ffff:192.1.1.25/96.

5. Configurar el acceso a red en un sistema operativo

Tenemos dos opciones para configurar conexión de red en un equipo:

Es frecuente que los servidores *DHCP* permitan definir un rango de *direcciones IP* para que no sean asignadas.

Estas direcciones quedarán libres para su asignación manual cuando sea necesario.

● *Automática*: En estos casos, dispondremos de un dispositivo en la red con capacidades de servidor *DHCP* (*Dynamic Host Configuration Protocol*, en español, *Protocolo de Configuración Dinámica de Host*). Éste permitirá asignar, de forma automática, una dirección IP, la dirección de un servidor DNS, una máscara de subred, etc., a cualquier equipo que lo solicite.
Los datos asignados podrán cambiar, de un arranque al siguiente, en función de las necesidades del servidor *DHCP*.

● *Manual*: En lugar de la asignación automática, tenemos la oportunidad de establecer los valores de conexión de forma manual.
Es muy común utilizar este modo de configuración en ordenadores que comparten recursos en la red o que actúan como servidores. El objetivo es lograr que sean fácilmente localizables.

Normalmente, cuando nos disponemos a configurar, de forma manual, el acceso a la red en un sistema operativo, será necesario tener en cuenta los siguientes aspectos:

● ***Dirección IP* y *Máscara de subred*:** Ya hemos hablado ampliamente de ellas en los apartados anteriores.

● ***Puerta de enlace predeterminada***: Es la *dirección IP* del dispositivo que nos permite acceder a una red diferente (normalmente, Internet). Es frecuente que dicho dispositivo sea un *enrutador* (*router*), pero también puede ser otro ordenador de la red.

● ***Servidor DNS***: Será la *dirección IP* del servidor de la red al que acudiremos para que traduzca un *nombre de dominio* en su *dirección IP* correspondiente

Incluimos a continuación artículos sobre la configuración del acceso a la red en los sistemas operativos más habituales.

- Averiguar la IP en un ordenador con Windows 10.
- Asignar una dirección IP fija en Windows 10.
- Encuentra todos los dispositivos de tu red con Angry IP Scanner sobre Windows 10.
- Consultar la configuración de la red en Windows con ipconfig.
- Netsh: Configurar la red en Windows 10 desde la línea de comandos.
- Averiguar la dirección IP en un ordenador con Ubuntu 18.04 LTS.
- Buscar los dispositivos conectados a la red local con Angry IP Scanner.
- Consultar la configuración de la red en Ubuntu con ifconfig.
- Configurar la red en Ubuntu modificando el archivo de configuración.
- Averiguar la IP en un ordenador con Ubuntu desde la línea de comandos.
- Establecer una dirección IP estática en Ubuntu Server 17.10 y posteriores.
- Averiguar la IP pública desde Ubuntu.
- Averiguar la dirección MAC de los dispositivos de nuestra red en Ubuntu.

http://somebooks.es/capitulo-2-conexion-sistemas-red/7/

Verificación red

Verificación protocolo TCP/IP: se realiza haciendo ping localhost (o ping 127.0.0.1). Esto permite verificar si el protocolo TCP/IP está correctamente instalado y en funcionamiento. Es enviado y respondido internamente por el propio equipo.

Verificación del adaptador de red: haciendo ping al propio equipo (p. ej: 192.168.0.2). El comando es enviado a través de la red y recibido por el propio equipo, el cual envía la respuesta a la red y la recoge de ella. Esto permite verificar si la tarjeta de red está funcionando correctamente.

Verificación de la red local: haciendo ping a un equipo próximo conectado a la red . Esto permite verificar el cableado del equipo hacia la red. Si se ejecuta ping “puerta de enlace” se puede verificar si el cableado general de la red funciona correctamente.

Verificación de la conexión a Internet: ping a otro equipo conectado a Internet. Por ejemplo: ping 173.194.45.23 (IPv4 de google).

Verificación de los servidores DNS: ping google.es o cualquier otra dirección de Internet (URL) conocida. Esto verifica si están correctamente configuradas las IPs de los servidores DNS, puesto que el servidor debe responder con la IP de la dirección de Internet.