## Sistemas Informáticos, 1º DAM

# Sistemas en red: Modelo OSI y Modelo TCP/IP

### Concepto de protocolo

En informática, podríamos definir el concepto de **protocolo** como el conjunto de reglas y procedimientos que utilizan los ordenadores para comunicarse entre ellos a través de una red.

Deben establecer una semántica y una sintaxis que permitan identificar a cada participante de forma inequívoca, establecer la comunicación, mantenerla sincronizada y recuperarse de posibles errores.

Como es lógico, todos los dispositivos que participen en la comunicación deben compartir los mismos protocolos.

### 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 entre las redes de diferentes fabricantes.

> ☞ El IEEE define la **interoperabilidad** como la habilidad de dos o más sistemas o componentes para intercambiar información y utilizarla posteriormente.

Para su desarrollo se basó en varios modelos, como DECnet (de la empresa Digital Equipment Corporation), SNA o Systems Network Architecture (de la empresa IBM), o TCP/IP, desarrollado por DARPA (Defense Advanced Research Projects Agency) para la red ARPANET.

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.

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*

El modelo de referencia OSI distribuye su funcionalidad en diferentes capas, lo que nos ofrece algunos beneficios:

- La comunicación se divide en componentes más pequeños y sencillos, simplificando su diseño e implementación.
- Los desarrolladores escriben una entrada y una salida concreta para cada capa, sin que importe cómo se ha desarrollado internamente.

- Los cambios en una capa no afectan al resto. Tampoco afecta el modo en el que se haya escrito una capa al funcionamiento de las demás. Por ejemplo, gracias a esto, las aplicaciones antiguas se siguen ejecutando sobre redes inalámbricas sin necesidad de cambiarlas.
- Cuando las funciones se dividen en partes más pequeñas, resulta más sencillo estandarizarlas.
- Tanto el hardware y software nuevo y antiguo pueden comunicarse entre ellos sin problemas.

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

7 capas que vamos a tratar en orden inverso.


**7. Aplicación**: Ofrece la interfaz entre el software de aplicación y la red, implementando los servicios que serán invocados por éste. Aquí se definen los protocolos que usan los programas para intercambiar datos, como FTP, HTTP, POP, SMTP, Telnet, etc.

En cualquier momento pueden crearse nuevos protocolos de este tipo para satisfacer las necesidades de algún nuevo tipo de aplicación.

**6. Presentación**: Se encarga de ofrecer los datos a la capa de Aplicación. Por lo tanto, se encarga de traducir y codificar los datos obtenidos de la aplicación a un formato genérico antes de transmitirlo por la red. A la inversa, los datos que recibe son convertidos de un formato genérico al formato que espera la capa de Aplicación.

De esta capa depende, por ejemplo, que puedan comunicarse sistemas que utilizan diferentes juegos de caracteres (como EBCDIC o ASCII). También se encarga de comprimir/descomprimir y de cifrar/descifrar la comunicación, cuando es necesario.

**5. Sesión**: En un equipo puede haber varias aplicaciones transmitiendo datos a través de la red (incluso varias instancias de la misma aplicación). Para cada una de ellas, se creará una sesión diferente, y será la capa de Sesión quien se encargue de hacerle llegar los datos que le correspondan.

Será en esta capa donde se creen, administren y destruyan las sesiones. Además, controla y coordina la comunicación entre sistemas

**4. Transporte**: Se centra en la transferencia de los datos de extremo a extremo de la red, encargándose de establecer una conexión lógica entre los dos equipos implicados e independizándolos de la red física que se esté usando.

Proporciona la entrega de datos (que puede ser fiable, o no) y ofrece control de flujo y recuperación de errores.

**3. Red**: Aquí se identifica el origen y el destino mediante direcciones lógicas únicas en la red. Si ambos participantes en la comunicación se encuentran en redes distintas, éstas deben encontrarse unidas mediante enrutadores (routers), que elegirán la mejor ruta posible para entregar el paquete en su destino (tarea que recibe el nombre de determinación de ruta).

Por lo tanto, los routers utilizan la capa de red para realizar su trabajo (direccionamiento lógico, determinación de la ruta y reenvío). También lo hacen algunos cortafuegos (firewalls) para descargar paquetes en función de su origen.

En la capa de red se utilizan dos tipos de protocolos:

- Enrutados: Utilizados para el direccionamiento lógico y el reenvío.
- De enrutamiento: Utilizados para la determinación de la ruta

Un dato importante es que los routers sólo se encargan de hacer llegar los paquetes a la red de destino. Una vez allí, el receptor concreto se identificará en la capa de enlace de datos.

**2. Enlace de datos**: Todos los ordenadores de una red disponen de una dirección física y otra lógica. La dirección física sólo es significativa dentro de la red local.

Es el nivel de enlace de datos el que centra su atención en la transferencia de datos dentro de la red local usando direcciones físicas.

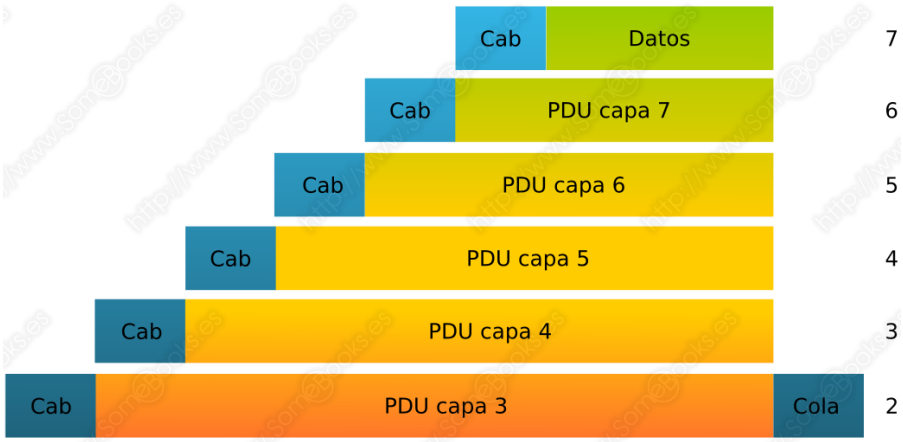
Cuando en un medio pueden coincidir varios equipos tratando de enviar paquetes al mismo tiempo, para evitar colisiones, se debe asegurar que sólo uno de ellos lo utiliza cada vez. En esta capa se establecen los protocolos usados para el envío y recepción de datos a través de los medios. Estos deberán determinar si el medio está disponible y detectarán las colisiones y los posibles errores en los datos recibidos.

**1. Física**: En esta capa es donde se gestionan los dispositivos físicos encargados de la comunicación. En ella se envían o reciben los bits que forman el mensaje sin interpretar su significado y será la encargada de iniciar, mantener y finalizar la comunicación entre sistemas (por ejemplo, entre un equipo y un switch).

De esta capa dependen las transformaciones que se realizan en los bits, que dentro del ordenador están representados por diferentes niveles de electricidad, para adaptarlos al medio de transmisión utilizado (óptico, infrarrojo, láser, radio, microondas, etc.).

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 se observa en la imagen, 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.

Como se deduce de lo dicho más arriba, podemos organizar las capas en dos grupos:

Las capas 4, 5, 6 y 7 definen cómo interaccionan las aplicaciones entre ellas, con el usuario y con la interfaz del equipo. Por este motivo, se agrupan bajo el nombre Host layers.

Las capas 1, 2, 3 definen cómo se transmiten los datos entre el equipo y la red. Suelen conocerse, en conjunto, con el nombre Media layers.

## El modelo TCP/IP

El modelo TCP/IP es un estándar abierto con un planteamiento muy parecido al modelo de referencia OSI. A pesar de todo, el modelo TCP/IP es el que se utiliza de forma generalizada en la actualidad.

En este caso, también encontramos una estructura de capas. Sin embargo, a diferencia del modelo OSI, ahora sólo encontramos cuatro:

- **Aplicación**: Implementa los diferentes protocolos de servicio incluidos en las capas de Aplicación (7), Presentación (6) y Sesión (5) incorporados en el modelo OSI. Así, esta capa incluye la comunicación con el software de aplicación, la traducción y la codificación de los datos, y el diálogo entre sistemas.

  Algunos de los protocolos de aplicación que se implementan en esta capa son: Telnet, HTTP, FTP, SMTP, TFTP, DNS y DHCP, entre otros, que se verán en este documento.

- **Transporte**: Como en el modelo OSI, esta capa administra la conexión lógica entre los equipos que se comunican y el envío de datos entre ellos.

  En esta capa disponemos de dos protocolos distintos: TCP y UDP. También en este caso, hablaremos de ellos más adelante.

- **Internet**: Como recordarás, en el modelo OSI, la capa de red identificaba el origen y el destino de la transmisión y, cuando se encontraban en redes diferentes, los enrutadores (routers) que los unían elegían la mejor ruta posible para entregar cada paquete en su destino. En el caso del modelo TCP/IP, esta tarea corresponde a la capa Internet. Por lo tanto, los routers utilizan esta capa para realizar su trabajo (direccionamiento lógico, determinación de la ruta y reenvío).

  En esta capa encontramos cinco protocolos, IP, ARP, ICMP, IGMP y RARP, que se verán en este documento.

- **Acceso a la red**: Esta capa se corresponde con las capas Enlace de datos y Física del modelo OSI. Contiene las especificaciones relativas a la transferencia de los paquetes recibidos de la capa Internet a través de la red física. El destinatario puede ser otro equipo de la misma red o un router que se encargue de enviar los paquetes a una red distinta.

  Por lo tanto, a diferencia de la capa Internet, que tiene alcance sobre toda la red, el de la capa de Acceso a la red sólo llega hasta el primer router.

  En esta capa, se resuelven aspectos como:

  - Enrutamiento de datos a través de la conexión
  - Sincronización de la transmisión
  - Formato de los datos

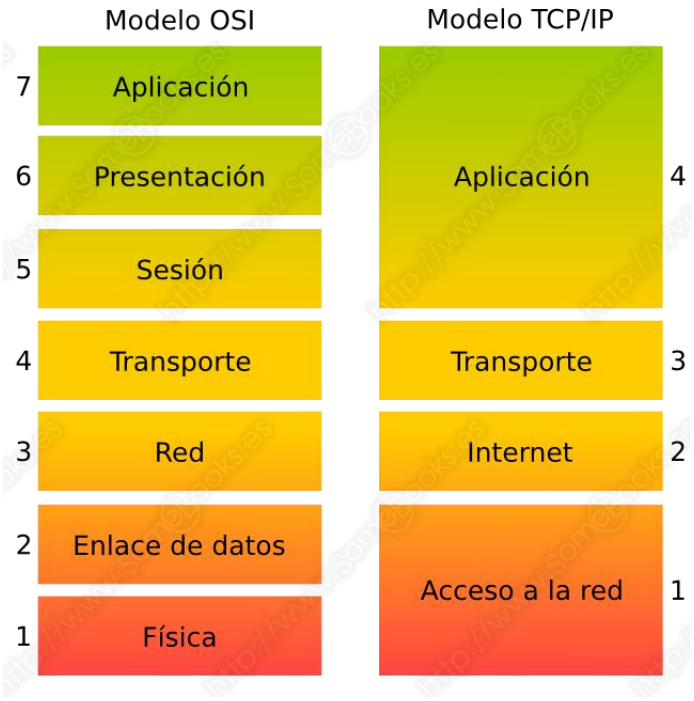
- Conversión de señales (analógica/digital)
- Detección de errores
- etc.

Como se puede deducir, en esta capa encontramos un gran número de protocolos, como los protocolos Ethernet, cuando nos encontramos en una red local, o los protocolos PPP (Point-to-Point Protocol) y Frame Relay, cuando nos encontramos en una red WAN. De cualquier modo, son el sistema operativo y los controladores de hardware los responsables de su manejo.

## Correspondencia entre el modelo OSI y el modelo TCP/IP

A pesar de que el modelo OSI dispone de siete capas diferentes y el modelo TCP/IP sólo tiene cuatro, podemos establecer un paralelismo entre ambos modelos teniendo en cuenta su funcionalidad.

Esta correspondencia puede verse en la siguiente imagen:




## Protocolos de la capa de aplicación

La capa de Aplicación tiene asociados diferentes protocolos. Entre los más frecuentes, tenemos los siguientes:

- Telnet: Es un protocolo que permite emular una terminal de texto de un equipo remoto. El equipo accedido (servidor) debe ejecutar un servicio, también llamado Telnet, que atiende las solicitudes de los equipos remotos (clientes).

  En el sistema operativo cliente, también necesitaremos un programa que establezca la conexión y nos muestre el contenido de la terminal de texto del servidor.

  El sistema operativo del servidor procesará las órdenes recibidas en la terminal remota como si procedieran de los dispositivos de entrada locales.

- HTTP (*Hypertext Transfer Protocol*, en español, Protocolo de Transferencia de Hipertexto): Facilita la transferencia de páginas web, con todo su contenido relacionado, entre un equipo servidor y un equipo cliente.

  El programa que utiliza este protocolo en el lado servidor suele denominarse servidor web o servidor http. De igual modo, el programa que realiza las solicitudes y recibe los contenidos en el lado cliente suele llamarse cliente web o cliente http. Es común que la tarea de cliente web la realice un navegador de internet.

- FTP (*File Transfer Protocol*, en español, Protocolo de Transferencia de Archivos): Es un protocolo para la transferencia de archivos entre dos equipos. De nuevo, necesitamos un software específico que actúe como servidor y como cliente en ambos lados de la conexión.

  Con el programa cliente, el usuario tendrá acceso al árbol de directorios (o a una parte de éste) y descargar o subir archivos en él.

- SMTP (*Simple Mail Transfer Protocol*, en español, Protocolo Para Transferencia Simple de Correo): Permite enviar correos electrónicos, desde un programa que actúa como **cliente smtp** al **servidor smtp** de la entidad que provee el correo al usuario remitente.

  Dicho servidor enviará el mensaje al servidor smtp que provee el correo al usuario destinatario, usando también el protocolo SMTP.

  Finalmente, éste último envía el mensaje al destinatario usando los protocolos **POP3** o **IMAP**.

- TFTP (Trivial File Transfer Protocol, en español, Protocolo de Transferencia de Archivos Trivial): Es una versión simplificada del protodolo FTP que sólo permite enviar y recibir archivos. Es muy rápido y fácil de implementar, pero también inseguro, porque no permite autenticación.
- DNS (*Domain Name System,* en español, Sistema de Nombres de Dominio): Se trata de un protocolo que permite atender solicitudes para un **servidor dns**. Las funciones de este tipo de servidores consisten en traducir un **nombre de dominio** en su **dirección IP** correspondiente. Así se evita que el usuario deba aprender las direcciones IP de los servidores que visita.

  El uso del servidores DNS facilita la migración de dominios, o nombres de dominios, a otros equipos. Cuando hay cambios, únicamente hay que actualizar en los servidores DNS las nuevas direcciones IP de los nombres de dominio.

  De este modo, cada vez que escribimos la dirección de una página web en un navegador, éste actúa como cliente de un servidor dns que la traduce en su dirección IP correspondiente. Después, el navegador establecerá una sesión HTTP con el servidor de dicha dirección IP.

- DHCP (*Dynamic Host Configuration Protocol*, en español, Protocolo de Configuración Dinámica de Host): Mediante este protocolo, un equipo de la red puede obtener la configuración de sus tarjetas de red, lo que incluye: la dirección IP de la tarjeta, la dirección de la puerta de enlace, la dirección de un servidor dns, una máscara de subred, etc., de forma automática. El equipo que lo proporciona recibe el nombre de **servidor dhcp**.

## Protocolos de la capa de transporte

La capa de transporte tiene asociados dos protocolos: TCP y UDP

- **TCP** (*Transmission Control Protocol*, en español, Protocolo de Control de Transmisión): Es un protocolo confiable, orientado a la conexión, que utiliza Un método de acuse de recibo independiente de los niveles inferiores.

  ☞ Un datagrama es un fragmento de un mensaje con la información necesaria para encaminarlo a su destino. Esto significa que los diferentes fragmentos del mismo mensaje pueden seguir diferentes caminos y llegar desordenados o, incluso, no llegar

  De este modo, los routers que intervienen en la comunicación, y que sólo actúan a nivel de red, únicamente ven un datagrama y no tienen responsabilidad sobre su contenido. Será TCP el encargado de verificarlos y ordenarlos.

- **UDP** (*User Datagram Protocol*, en español, Protocolo de datagrama de usuario): Se trata de un protocolo muy simple, y no confiable, que no incluye detección de errores. Es decir, no está orientado a la conexión.

  La cabecera UDP sólo contiene la longitud del paquete y un chequeo de sumas (checksum). Esto lo hace más ligero y rápido, y lo convierte en la solución ideal para el transporte de información pesada, como vídeo o audio, siempre que la red sea fiable.

En estos casos, las comprobaciones de confiabilidad se atienden en la capa de aplicación.

## Concepto de puerto

Tanto TCP como UDP utilizan el concepto de número de puerto para realizar su trabajo.

Un equipo que se encuentra unido a una red, puede estar ejecutando a la vez varias aplicaciones que envíen y/o reciban datos a través de esta. La forma de separar el tráfico de cada aplicación consiste en que éstas usen puertos diferentes. Tanto TCP como UDP añaden un número de puerto en el encabezado del tráfico que generan.

El puerto concreto que se utilice dependerá del protocolo de aplicación utilizado en la capa de aplicación. De hecho, los más comunes disponen de un puerto predeterminado en el intervalo 1 hasta 1024. De este modo, los programas que usen dichos protocolos escucharán sus puertos correspondientes TCP o UDP, analizando la cabecera, sabrán a quién hacer llegar los paquetes que reciban.

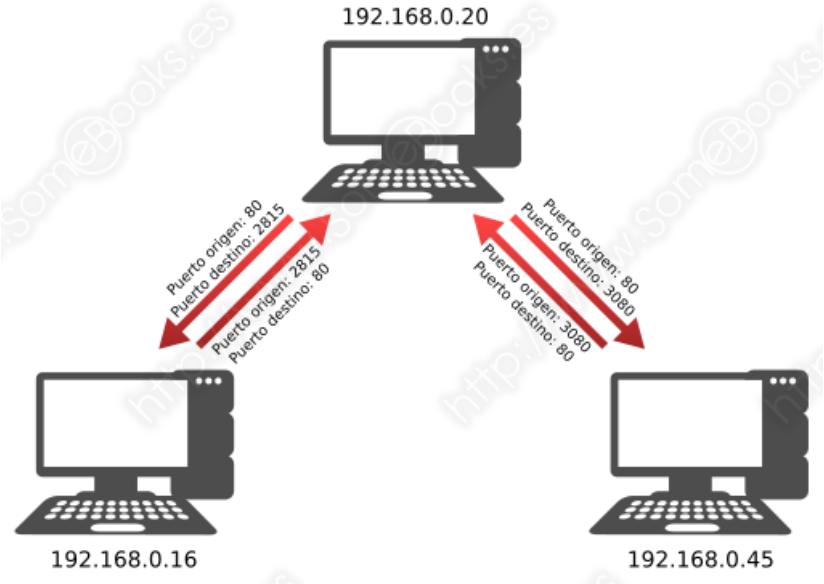
En la tabla se muestra una lista con los números de puerto asignados a los protocolos de aplicación más usados:

| Protocolo de aplicación | Protocolo de transporte | Número de puerto |
|---|---|---|
| Telnet | TCP | 23 |
| HTTP | TCP | 80 |
| HTTPS | TCP | 443 |
| FTP (control) | TCP | 21 |
| FTP (datos) | TCP | 20 |
| SMTP | TCP | 25 |
| TFTP | UDP | 69 |
| DNS | TCP, UDP | 53 |
| DHCP | UDP | 67 (server)<br>68 (cliente) |
| SSH | TCP | 22 |

En el equipo que inicia la comunicación también se asigna un número de puerto aleatorio a cada sesión (siempre mayor que 1024) para recibir el tráfico de retorno.

El conjunto formado por la dirección IP del equipo, el protocolo utilizado (TCP o UDP) y el número de puerto recibe el nombre de **socket**.

El concepto de socket permite que una aplicación pueda comunicarse con diferentes equipos de la red de forma simultánea. El motivo es que el socket usado con cada uno será diferente.




## Protocolos de la capa Internet

En esta capa encontramos los siguientes cinco protocolos: IP, ARP, ICMP, IGMP y RARP.

**IP** (*Internet Protocol*, en español, Protocolo de Internet): Se encarga del transporte de datagramas, pero no garantiza su entrega.

**ARP** (*Address Resolution Protocol*, en español, Protocolo de Resolución de Dirección): Lo utilizan los enrutadores (routers) para obtener información sobre las diferentes redes que se encuentran a su alcance.

**ICMP** (*Internet Control Message Protocol*, en español, Protocolo de mensajes de control de Internet): Facilita la administración de los mensajes de error.

**IGMP** (*Internet Group Management Protocol*, en español, Protocolo de Gestión de Grupo de Internet): Permite que diferentes equipos formen parte de un grupo de multidifución. Los routers analizan periódicamente si se mantiene la pertenencia.

**RARP** (*Reverse Address Resolution Protocol*, en español, Protocolo de Resolución de Direcciones Inverso): Facilita la resolución de la dirección IP de un dispositivo a partir de su dirección hardware (por ejemplo, su dirección Ethernet).