Tema 3: ARQUITECTURA DE COMUNICACIONES

1. *Conceptos de arquitectura estructurada*
2. *Elementos de la arquitectura OSI*
3. *Interfaces y Servicios*
4. *Primitivas de servicios*
5. *Características generales de los niveles del modelo OSI*
6. *Otras Arquitecturas de Red*
7. *Introducción a las arquitecturas de redes de área local*

3.1.- Conceptos de arquitectura estructurada

Para evitar confusiones entre los diferentes tipos de sistemas de organización de las redes conviene aclarar previamente algunos conceptos. Para ello se considerarán equipos informáticos donde los equipos emisores y receptores son ordenadores con capacidad para mantener una comunicación.

En una primera aproximación se llama **host** o **nodo** a *aquel ordenador que tiene capacidad de interactuar en la red*, es decir, *aquel ordenador capaz de alojar algún tipo de servicio de la misma*. (Técnicamente no es lo mismo host que nodo, aunque mas tarde se precisen estos conceptos, en una primera aproximación pueden identificarse).

Un **sistema aislado** es un ordenador incapaz de comunicarse con el exterior por vía telemática. Un ordenador con software y hardware adecuado para poder operar en red dispondrá de *recursos telemáticos de comunicación* que le hará mas flexible y le permitirá adquirir una mayor capacidad de acción que un sistema totalmente aislado.

En ocasiones, los sistemas aislados pueden efectuar *conexiones temporales*, normalmente a través de redes públicas, para realizar intercambios de información con el exterior. El sistema sólo está conectado temporalmente. Se dice entonces que el sistema está realizando **conexiones remotas temporales**. Por ejemplo, en las conexiones remotas particulares a Internet a través de empresas que ofrecen servicios telemáticos, las estaciones de los usuarios sólo pertenecen a la red cuando se realiza la conexión.

Cuando distintos equipos se conectan a través de una **red de datos** pero sin perder identidad propia se dice que se ha establecido una **red de ordenadores**. Si un usuario solicita un servicio a una red de ordenadores, la solicitud debe presentarse en una máquina concreta y solicitar un servicio determinado ya que *la red distingue todos y cada uno de sus equipos*.

Un **sistema distribuido** está compuesto por una red de ordenadores con una particularidad especial: *la existencia de múltiples ordenadores en la red es totalmente transparente al usuario.* Esto significa que puede realizarse una operación en la red y obtener unos resultados sin saber, a ciencia cierta, qué ordenador de la red ha atendido la petición efectuada. La red se comporta en sí misma como un sistema que gestiona todos los recursos de los ordenadores que posee.

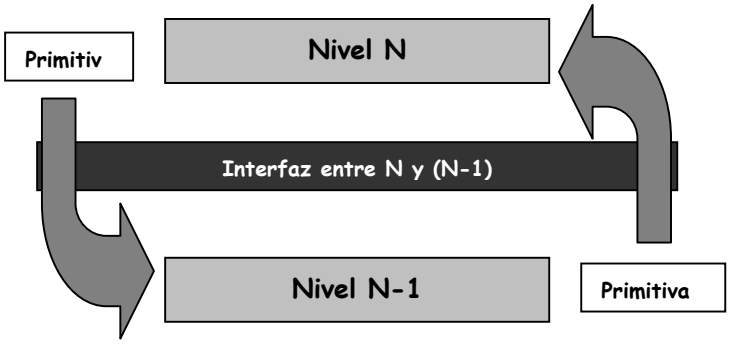
Un **protocolo de comunicaciones** es un *conjunto de reglas perfectamente organizadas y convenidas de mutuo acuerdo entre los participantes en una comunicación, cuya misión es regular algún aspecto de la misma*. Lo habitual es que los protocolos estén dados como normativas o recomendaciones de las asociaciones de estándares.

Con el fin de simplificar la complejidad de cualquier red, los diseñadores estructuran las diferentes funciones que realizan y los servicios que proveen, en diferentes **niveles** o **capas.**

Las capas están *jerarquizadas*: cada capa se construye sobre su predecesora. El número de capas y, dentro de cada una de ellas, el número de servicios y funciones que puede realizar, es variable para cada tipo de red. Sin embargo, en cualquier red, ***la misión de cada capa es proveer servicios a las capas superiores***, haciéndoles transparentes el modo en que estos servicios se llevan a cabo. Así cada capa debe ocuparse exclusivamente de su nivel inmediatamente inferior, al quien solicita servicios, y de su nivel inmediatamente superior, a quien devuelve resultados.

Dos capas consecutivas, como se acaba de decir, mantienen relaciones, es mas, estas relaciones son las únicas que existen en las **redes estructuradas** como sucesión ordenada de capas. El *modo en que cada capa negocia los servicios y se comunica con las capas* está fijado por *normas de intercomunicación entre capas* o **interfaz** de capa

La **interfaz**, entendida como la definición de los servicios y operaciones que la capa inferior ofrece a la superior, se gestiona como una estructura de **primitivas**. Las **primitivas** son llamadas entrantes o salientes en cada una de las capas que sirven para solicitar servicios, devolver resultados, confirmar las peticiones, etc.



La **arquitectura de una red** es el conjunto organizado de capas y protocolos de la misma. Esta organización de la red debe ser lo suficientemente clara como para que los fabricantes de software y hardware puedan diseñar sus productos con la garantía de que funcionarán en comunicación con otros equipos que sigan las mismas reglas.

Obsérvese que no se han incluido en la definición de la arquitectura las interfaces. Ello es debido a que la estructura de capas los oculta totalmente. Un interfaz concreto requiere ser conocido exclusivamente por las dos capas adyacentes a las que separa.

Por último, el concepto de **sistema abierto** fue propuesto inicialmente por la **ISO** (*International Standards Organization*) como *aquel sistema compuesto por uno o mas ordenadores, el software asociado, los periféricos, los procesos físicos, los medios de transmisión de la información, etc. Que constituyen un todo autónomo capaz de realizar un tratamiento de la información.*

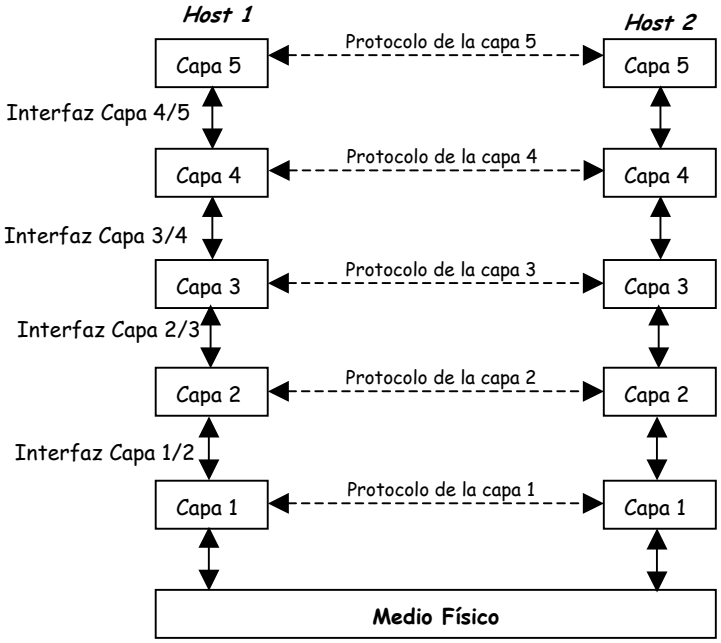
Posteriormente se redefinió como *un sistema capaz de interconectarse con otros de acuerdo con unas normas establecidas.* Por tanto la ***interconexión de sistemas abiertos*** **(OSI** *Open Systems Interconnection*) se ocupará del intercambio de información entre

sistemas abiertos, siendo su objetivo la confección de una serie de normas que permitan la intercomunicación de estos sistemas.

3.2.- Elementos de la arquitectura OSI.-

En 1977 la Organización Internacional de Estandarización **ISO** estableció un subcomité encargado de diseñar una arquitectura de comunicación. El resultado fue el *Modelo de referencia para la Interconexión de Sistemas Abiertos* ***OSI***, adoptado en 1983, que establece unas bases que permiten conectar sistemas ***abiertos*** para procesamiento de aplicaciones distribuidas. Se trata de un marco de referencia para definir estándares que permitan comunicar ordenadores heterogéneos.

Como se ha dicho antes, para reducir la complejidad de su diseño, muchas redes están organizadas como una serie de **capas** o **niveles**, cada una construida sobre la anterior. El número de capas, sus nombres, el contenido que tienen y la función que desempeñan difieren de una red a otra. Sin embargo, en todas las redes, el propósito de cada capa es ofrecer ciertos **servicios** a las capas superiores de modo que no tengan que ocuparse del detalle de la implementación real de los servicios.



La lista de protocolos empleados por un cierto sistema, con un protocolo por capa, se llama **pila de protocolos**.

La capa ***n*** de una máquina (***host 1***) lleva a cabo una conversación con la capa ***n*** de otra máquina (***Host 2***). *Las reglas y convenciones que se siguen en esta conversación* se conocen colectivamente con el nombre de **protocolo de la capa n**. El protocolo es, pues un acuerdo entre las partes que se comunican sobre cómo va a proceder la comunicación.

En la figura se ha representado una red de cinco capas. Las entidades que comprenden las capas correspondientes en las diferentes máquinas se denominan **pares.** En otras palabras, son los pares los que se comunican usando el protocolo.

En realidad los datos no se transfieren directamente de la capa n de una máquina a la capa n de la otra. Mas bien, cada capa pasa datos e información de control a la capa que está inmediatamente debajo de ella (n-1) hasta llegar a la capa mas baja. Bajo esta capa

(capa 1) está el **medio físico** a través del cual ocurre la comunicación real. En la figura se ha representado con líneas punteadas la *comunicación virtual* y con líneas continuas la *comunicación real*.

Entre cada par de capas adyacentes hay una **interfaz**. La interfaz define cuáles operaciones y servicios primitivos ofrece la capa inferior a la superior, lo que permite que cada capa ejecute una colección específica de funciones bien conocidas.

Como ejemplo de aplicación puede considerarse cómo proveer la comunicación a la capa superior de la red de cinco capas de la figura adjunta:

1. Se produce un mensaje **M** por un proceso de aplicación que se ejecuta en la capa 5 de la máquina emisora y se entrega a la capa 4 para su transmisión.

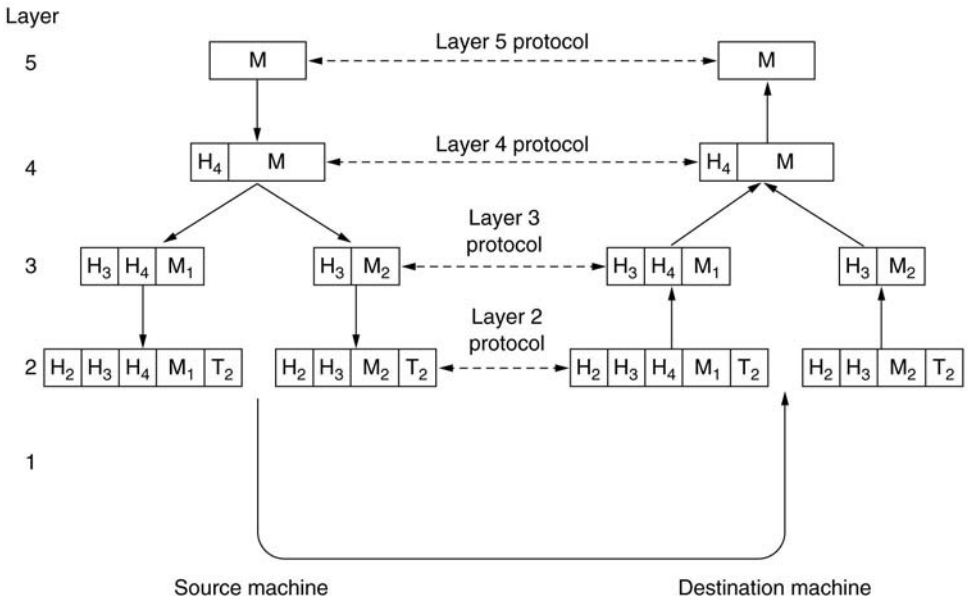
2. La capa 4 coloca un **encabezado** al principio del mensaje para identificarlo y pasa el resultado a la capa 3. El encabezado incluye información de control, como números de secuencia, para que la capa 4 de la máquina receptora pueda entregar los mensajes en el orden correcto si las capas inferiores no mantienen la secuencia.

3. La capa 3 de la máquina emisora divide los mensajes que le llegan en unidades mas pequeñas, anexando un encabezado de la capa 3 en cada paquete. En este ejemplo **M** se divide en dos partes $\mathbf{M_1}$ y $\mathbf{M_2}$

4. La capa 3 decide cual de las líneas que salen usará y pasa los paquetes a la capa 2.

5. La capa 2 no solamente añade un encabezado a cada pieza, sino también un apéndice y entrega la unidad resultante a la capa 1 para la transmisión física.

6. En la máquina receptora el mensaje se mueve hacia arriba, de capa en capa, perdiendo los encabezados conforme avanza. Ninguno de los encabezados para capas inferiores a la ***n*** pasa hasta la capa ***n***.



Es importante observar la relación entre la comunicación virtual y la real y la diferencia entre protocolos e interfaces.

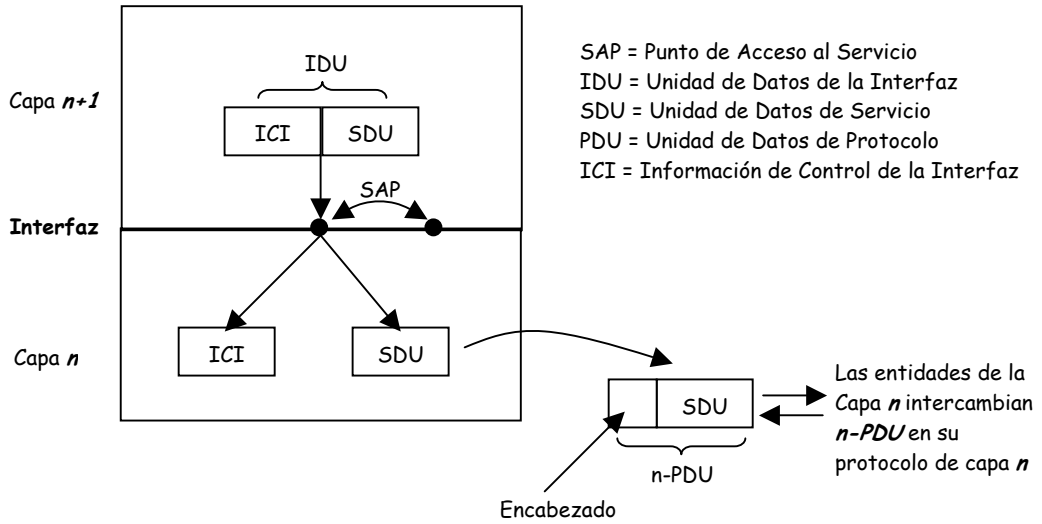
3.3.- Interfaces y servicios.-

La función de cada capa es proporcionar **servicios** a la capa que está encima de ella. Los elementos activos de cada capa generalmente se llaman **entidades**. Una entidad puede ser de software (como un proceso) o de hardware (como un circuito integrado inteligente de entrada / salida). Las entidades de la misma capa en máquinas diferentes se llaman **entidades pares.** Las entidades de la capa ***n*** implementan un servicio que utiliza la capa superior (***n+1***). A la capa ***n*** se llama entonces **proveedor del servicio** y la capa ***n+1*** es el **usuario del servicio**.

A su vez la capa ***n*** puede usar los servicios de la capa ***n-1*** con el fin de proveer su propio servicio, pudiendo ofrecer varias clases de servicio.

Los servicios están disponibles en los **Puntos de Acceso al Servicio** (*SAP.- Service Access Points*). Los SAP de la capa ***n*** son los lugares en los que la capa ***n+1*** puede tener acceso a los servicios ofrecidos. Cada SAP tiene una ***dirección*** que lo identifica de manera única.

Para que dos capas intercambien información tiene que haber un acuerdo sobre el conjunto de reglas relativas a la interfaz. En una interfaz típica, la entidad de la capa ***n+1*** pasa una **Unidad de Datos de la Interfaz** (*IDU.- Interface Data unit*) a la entidad de la capa ***n*** a través del SAP. La IDU consiste en una **Unidad de Datos de Servicio** (*SDU.- Service Data Unit*) y cierta información de control. La SDU es la información que se pasa mediante la red a la entidad par y, después, hasta la capa ***n+1***. La información de control es necesaria para ayudar a la capa inferior a efectuar su trabajo, pero no forma parte de los datos mismos.



Para que se transfiera la SDU, la entidad de la capa ***n*** puede tener que fragmentarla en varios pedazos, a cada uno de los cuales se les da un encabezado y se envía como una **Unidad de Datos de Protocolo** (*PDU.- Protocol Data Unit*) independiente, que podría ser un paquete. Las entidades pares usan los encabezados de las PDU para acarrear su protocolo de par. Los encabezados indican cuáles PDU contienen datos y cuáles contienen información de control: proveen números de secuencia, cuentas, etc.

En lo que a servicios se refiere, las capas pueden ofrecer dos tipos diferentes de servicios a las capas que se encuentran sobre ellas: los **servicios orientados a la conexión** y los **servicios sin conexión.**

- En el **servicio orientado a la conexión** el usuario establece primero una conexión, la utiliza y, posteriormente la libera. El aspecto esencial de una conexión es que actúa como un tubo: el emisor empuja objetos (bits) por un extremo y el receptor los saca, en el mismo orden, por el otro extremo.

  Entre la conexión y la liberación se produce el intercambio de datos de usuario. Los bloques de datos se reciben en el destino siguiendo el mismo orden en que se emitieron en el origen. Todos los paquetes siguen la misma ruta, la conseguida en el establecimiento de la conexión. Por tanto, los paquetes de datos no necesitan especificar la dirección de destino.

  Los servicios orientados a la conexión tienen las siguientes variantes:

  - **Secuencia de mensajes.-** En estos servicios se establecen fronteras que definen y determinan cada mensaje. La secuencia de mensajes es equivalente a la sincronización de bloque estudiada anteriormente.

  - **Secuencia de bytes**.- En estos servicios no hay contornos entre mensajes. Cada mensaje es una secuencia de caracteres, dejando al receptor la responsabilidad de su interpretación.

- En el **servicio sin conexión** cada mensaje lleva la dirección completa de destino y, cada uno, se encamina a través del sistema de forma independiente de todos los demás. Normalmente cuando se envían dos mensajes al mismo destino, el primero que se envió

será el primero en llegar, pero es posible que el primero se retrase tanto que sea el segundo el que llegue antes. Con un servicio orientado a la conexión esto es imposible. Cada servicio se puede caracterizar por una **calidad del servicio**. Algunos servicios son confiables en el sentido de que nunca pierden datos. Usualmente un servicio confiable se implementa haciendo que el receptor *acuse recibo* de cada mensaje, de modo que el emisor esté seguro de que el mensaje ha sido recibido.

Estos servicios proporcionan capacidades de comunicación sin necesidad de realizar la conexión con el destinatario. El emisor envía paquetes de datos al receptor confiando en que la red será lo suficientemente inteligente como para conducir los datos por las rutas adecuadas. Cada paquete lleva la dirección de destino y, en algunos casos, el receptor ha de enviar un **acuse de recibo** al emisor para informarle del éxito de la comunicación.

Existen varios tipos de servicios sin conexión:

- **Servicio de datagrama sin confirmación.-** El emisor no necesita confirmación del receptor de que los paquetes enviados les llegan correctamente. Un ejemplo es el protocolo IP.
- **Servicio de datagrama con confirmación.**- El receptor envía confirmaciones al emisor. Un ejemplo es el correo electrónico con acuse de recibo.
- **Servicio de petición y respuesta.-** Es un servicio propio de gestión interactiva basado en que a cada petición le sigue una respuesta.

	Servicio	Ejemplo
Orientado a la conexión	Flujo de mensaje confiable	Secuencia de páginas
	Flujo de bytes confiable	Ingreso remoto
	Conexión no confiable	Voz digitalizada
Sin conexión	Datagrama no confiable	Correo electrónico (propaganda
	Datagrama con acuse de recibo	Correo registrado
	Petición / Respuesta	Consulta de Base de Datos

3.4.- Primitivas de servicios.-

Un servicio se especifica de manera formal con un conjunto de operaciones o **primitivas** disponibles para que un usuario u otra entidad acceda al servicio. Estas primitivas ordenan al servicio que ejecute alguna acción o que informe de una acción que haya tomado una entidad par.

Las primitivas suelen dividirse en cuatro clases:

Primitiva	Significado
Petición	Una entidad quiere que el servicio haga un trabajo

Indicación	Se le informa a una entidad acerca de un suceso
Respuesta	Una entidad quiere responder a un suceso
Confirmación	Ha llegado la respuesta a una petición anterior

La utilización de las primitivas puede ilustrarse en el modo de establecerse y liberarse una conexión: La entidad que inicia la conexión efectúa una petición de conexión *CONNECT.request* que se concreta en el envío de un paquete. A continuación, el receptor recibe una indicación de conexión *CONNECT.indication* que le anuncia que en algún lugar una entidad quiere establecer una comunicación con él. La entidad que recibe la comunicación usa la primitiva de respuesta a la conexión *CONNECT.response* para indicar si quiere aceptar o rechazar la conexión propuesta. En cualquier caso, la entidad que emite la petición inicial averigua que sucedió por medio de la primitiva de confirmación de conexión *CONNECT.confirm.*

Las primitivas pueden tener parámetros y la mayoría de ellas los tiene: Los parámetros de una *petición de conexión* pueden especificar la máquina a la que se va a conectar, el tipo de servicio deseado y el tamaño máximo del mensaje a usar en una conexión.

Los parámetros de una *indicación de conexión* podrían contener la identidad de quien llama, el tipo de servicio deseado y el tamaño de mensaje máximo propuesto. Si la entidad llamada no está de acuerdo, con el tamaño máximo propuesto, podría presentar una contrapropuesta en su primitiva de *respuesta*, que se pondría a disposición del originador de la llamada en la *confirmación*. Los detalles de esta **negociación** son parte del protocolo.

Los servicios pueden ser **confirmados** o **no confirmados**. En un servicio confirmado existe una *petición*, una *indicación*, una *respuesta* y una *confirmación*. En un servicio no confirmado únicamente hay una *petición* y una *confirmación.* El servicio *CONNECT* (conexión) es siempre un servicio confirmado por que el par remoto debe estar de acuerdo con el establecimiento de la conexión. Sin embargo, la transferencia de datos puede ser confirmada o no confirmada dependiendo si el emisor necesita acuse de recibo o no.

Para concretar el concepto de servicio se considera el ejemplo de un servicio simple, orientado a la conexión, con las siguientes primitivas de servicio:

1. *CONNECT.request.*- Petición para establecer una conexión.
2. *CONNECT.indication.*- Envía una señal a la parte llamada.
3. *CONNECT.response.*- La usa el receptor para aceptar o rechazar la llamada.
4. *CONNECT.confirm.*- Indica al emisor si se aceptó o no la llamada.
5. *DATA.request.*- Petición de envío de datos.
6. *DATA.indication.*- Señal de llegada de los datos.
7. *DISCONNECT.request.*- Petición para liberar la conexión.

8. *DISCONNECT.indication*.- Indica al par la petición.

En este ejemplo, *CONNECT* es un servicio confirmado pues requiere una respuesta específica, mientras que *DISCONNECT* es un servicio no confirmado.

Los ***servicios*** y los ***protocolos*** son dos conceptos distintos. Un *servicio* es un conjunto de (operaciones) primitivas que ofrece una capa a la capa inmediatamente superior. El **servicio** define cuales son las operaciones que la capa está preparada para ejecutar en beneficio de sus usuarios, pero no dice cómo se instrumentan dichas operaciones.

El servicio se refiere pues a la interfaz entre dos capas, siendo la capa inferior la que provee el servicio y la capa superior la que hace uso de él.

Por su parte, un **protocolo** es un conjunto de reglas que gobiernan el formato y el significado de los marcos, paquetes o mensajes que se intercambian entre las entidades pares dentro de una capa. Las entidades usan protocolos con el fin de instrumentar sus definiciones de servicios.

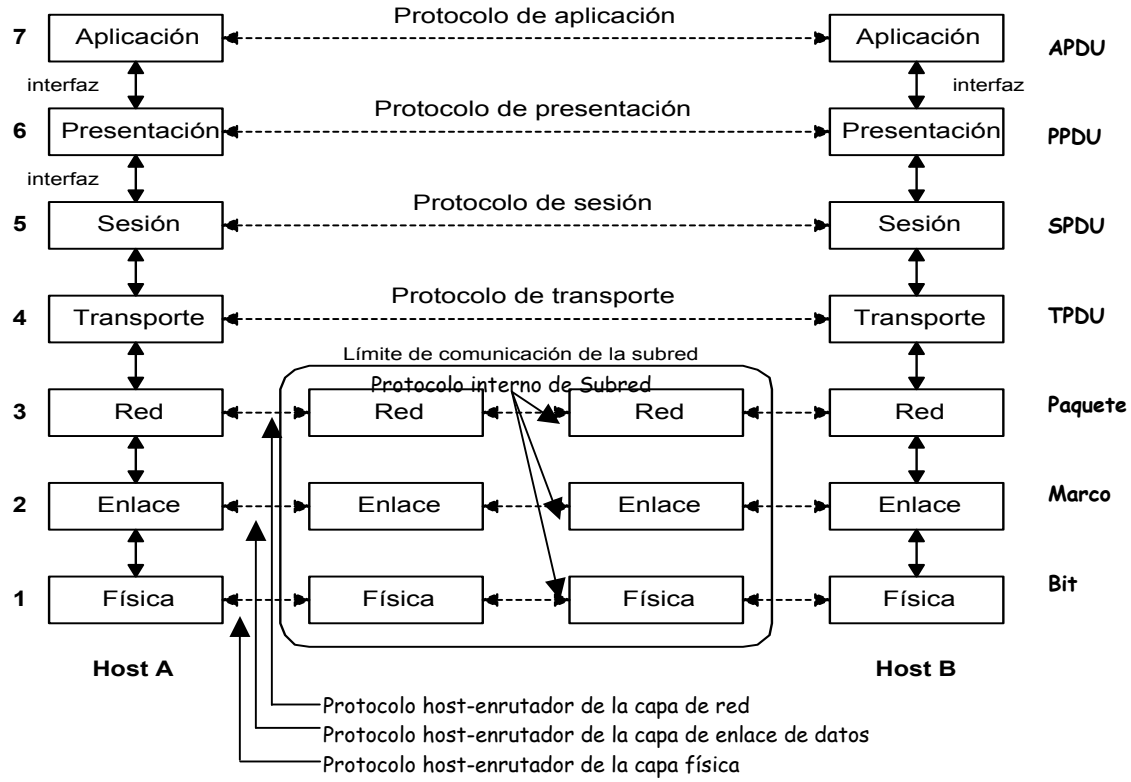
3.5.- Características generales de los niveles del modelo OSI.-

El modelo OSI define una arquitectura de comunicación estructurada en ***siete niveles o capas verticales***. Cada capa ejecuta un subconjunto de las funciones que se requieren para comunicar con el otro sistema. Para ello se apoya en los servicios que le ofrece la capa inmediata inferior y ofrece sus servicios a la capa que está por encima de ella. Idealmente, los cambios que se realicen en una capa no deberían afectar a su capa vecina mientras no se modifiquen los servicios que le ofrece.

La tarea del subcomité **ISO** fue definir el conjunto de capas y los servicios proporcionados por cada una. Los principios aplicados para llegar a las siete capas fueron los siguientes:

- Se debe crear una capa siempre que se necesite un nivel diferente de abstracción
- Cada capa debe ejecutar una función bien definida.
- La función de cada capa se debe elegir pensando en la definición de protocolos estandardizados internacionalmente.
- Los límites de cada capa deben elegirse de modo que minimicen el flujo de información a través de las interfaces.
- La cantidad de capas debe ser suficiente para no tener que agrupar funciones distintas en la misma capa, y lo bastante pequeña para que la arquitectura no se vuelva inmanejable.

El modelo OSI no es en sí una arquitectura de red porque no especifica los servicios y protocolos exactos que se han de utilizar en cada capa.



Los siete niveles que configuran el modelo **OSI** suelen agruparse en dos bloques. Los tres niveles inferiores (nivel físico, nivel de enlace y nivel de red) constituyen el **bloque de transmisión**. Son niveles dependientes de la red de conmutación utilizada para la comunicación entre los dos sistemas. Por el contrario, los tres niveles superiores (nivel de sesión, de presentación y de aplicación) son **niveles orientados a la aplicación** y realizan funciones directamente vinculadas con los procesos de aplicación que desean comunicarse. El nivel intermedio (**nivel de transporte**) enmascara a los niveles orientados a la aplicación, el funcionamiento detallado de los niveles dependientes de la red.

Niveles OSI orientados a la Red

3.5.1.- Capa Física.-

La capa física tiene que ver con la transmisión de bits por un canal de comunicación. Las consideraciones de diseño tienden a asegurarse de que, cuando el emisor envía un bit 1, el receptor reciba un bit 1 y no un bit 0. Las tareas de diseño deben contestar a las preguntas siguientes:

- Voltaje necesario para representar un 1 y voltaje necesario para representar un 0.
- Microsegundos que dura la emisión de un bit
- Tipo de transmisión, en una o en dos direcciones
- Establecimiento de la conexión inicial y forma de interrupción de la conexión, cuando emisor y receptor han acabado.
- Puntas que tiene el conector de la red y para que sirve cada una

Algunas de las normas dentro de este nivel son:

Norma	Contenido

X.24	Definiciones relativas a los circuitos de unión establecidos entre dos equipos sobre redes públicas de datos
V.10	Características eléctricas de los circuitos de intercambio de doble corriente asimétrica, ,para uso general en Teleinformática
V.11	Vomo V.10 pero para corriente simétrica
V.24/v.28	Características funcionales/eléctricas para circuitos de enlace entre dos equipos
V.35	Recomendación CciTT para transmisión de datos a 48 kbps por medio de circuitos en grupo binario de 60 a 108 KHz.
ISO 2110	Características mecánicas para el conector de V.24
EIA-232	Estándares a nivel físico, eléctrico y funcional de EIA

Son funciones del nivel físico:

- Establecer los caminos físicos
- Detectar errores en las señales
- Proporcionar a los niveles superiores independencia frente al medio físico
- Hacerse cargo de las particularidades electromecánicas del medio físico

3.5.2.- Capa de Enlace de Datos.-

La tarea principal de la **capa de enlace de datos** es tomar un medio de transmisión en bruto y transformarlo en una línea que parezca libre de errores de transmisión a la capa de red.

La capa de enlace de datos cumple esta tarea al hacer que el emisor divida los datos de entrada en **marcos de datos**, que transmita los marcos en forma secuencial y procese los **marcos de acuse de recibo** que devuelve el receptor.

Dado que la capa física únicamente acepta y transmite una corriente de bits sin preocuparse por su significado o su estructura, corresponde a la capa de enlace *crear y reconocer los límites de los marcos*, lo cual consigue añadiendo patrones especiales de bits al principio y al final de cada marco.

La aparición de una ráfaga de ruido en la línea puede destruir un marco por completo. En estos casos, el software de la capa de enlace de datos del emisor debe ofrecer la capacidad de retransmitir el marco. Corresponde pues a esta capa resolver los problemas provocados por marcos dañados, perdidos o duplicados.

También debe tenerse en cuenta, en la capa de enlace de datos, el evitar que un transmisor veloz *sature de datos* a un receptor lento. Debe emplearse algún mecanismo de regulación de trafico para que el emisor sepa cuanto espacio de almacenamiento temporal (*buffer*) tiene el receptor en ese momento. Con frecuencia esta regulación de flujo y el manejo de errores están integrados.

Pertenecen a este nivel:

Norma	**Contenido**

HDLC	(*High-Level Data Link Control):* Protocolo de alto nivel, orientado al bit (especificado por ISO 3309), para el control de enlace de datos, en modo síncrono
LAP-B	*(Link Access Procedure-Balanced):* Subconjunto del protocolo HDLC, definido por OSI, para acceso al enlace a redes X.25
IEEE 802.2-7	Para LAN

Son funciones del nivel de enlace:

- Coordinar la comunicación
- Recuperar los fallos
- Compartir el circuito físico entre varias transmisiones
- Sincronizar las tramas
- Establecer la transparencia de la división en tramas para los niveles superiores

3.5.3.- La capa de Red.-

La capa de red se ocupa de controlar el funcionamiento de la subred, esto es, proporciona los medios para establecer, mantener y liberar la conexión a través de una red, donde existe una malla compuesta de enlaces y nodos, entre sistemas abiertos que contienen entidades de aplicación en comunicación, así como los medios funcionales y de procedimiento para el intercambio de unidades de datos del servicio de red entre entidades de transporte por conexiones de red.

La capa de red es la responsable de las funciones de conmutación y encaminamiento de la información; proporciona los procedimientos precisos necesarios para el intercambio de datos entre el origen y el destino, por lo que es preciso que conozca la topología de la red para determinar la ruta mas adecuada.

Pertenecen a este nivel:

Norma	Contenido
X.25	Interconexión de redes públicas de equipos terminales de datos (ETD) y equipos de comunicación de datos (ECD) para terminales con funcionamiento a modo paquete, conectados a una red pública de transmisión de datos, con línea dedicada
X.32	Interface entre un ETD y un ECD para terminales que transmiten en modo paquete y acceden a la red pública X.25 a través de una red telefónica conmutada
X.3	Servicio complementario de ensamblado y desensamblado de paquetes de una red pública de datos
X.28	Interconexión entre ETD/ECD para el acceso a un ETD asíncrono al servicio de ensamblado y desensamblado de paquetes (DEP), en una red pública de datos
X.29	Procedimientos de intercambio de información de control y de datos de usuario entre un DEP y un ETD modo paquete u otro DEP
ISO 9542	Protocolo de encaminamiento para la LAN

Las funciones de este nivel son, entre otras:

- El encaminamiento de las conexiones
- La respuesta a configuraciones diferentes

- El mantenimiento de la secuencia en el envío de informaciones sucesivas

3.5.4.- La capa de Transporte.-

La capa de transporte efectúa la transferencia de datos entre entidades de sesión y las libera de toda otra función distinta de la de conseguir una transferencia de datos segura y económica.

Su misión básica es la de aceptar datos de la capa de sesión, dividirlos en unidades mas pequeñas si es necesario, pasarlos a la capa de red y asegurar que todos los trozos lleguen correctamente al otro extremo. Además, esto debe hacerse de forma eficiente, de manera que aísle a las capas superiores de los cambios inevitables en la tecnología del hardware.

La capa de transporte determina también qué tipo de servicio proporcionará a la capa de sesión y, finalmente, a los usuarios de la red. El tipo de servicio se determina al establecer la sesión.

La capa de transporte es una verdadera capa de extremo a extremo, del rigen al destino. En otras palabras, un programa en la máquina emisora sostiene una conversación con un programa similar en la máquina receptora, haciendo uso de los encabezados de mensajes y de los mensajes de control. En las capas bajas, los protocolos se usan entre cada máquina y sus vecinas inmediatas, y no entre las máquinas origen y destino, que pueden estar separadas por varios enrutadores.

La diferencia entre las capas 1 a la 3 es que están encadenadas, frente a las capas 4 a la 7, que son de extremo a extremo. Muchos nodos están multiprogramados, lo que implica que múltiples conexiones entran y salen de cada nodo. En este caso, el **encabezado de transporte** (H4 de una figura anterior) es la opción para saber cual mensaje pertenece a qué conexión.

Además de multiplexar varias corrientes de mensajes por un canal, la capa de transporte debe cuidar de establecer y liberar conexiones a través de la red. Esto requiere alguna clase de asignación de nombres, de modo que un proceso en una máquina pueda describir con quien quiere conversar.

También debe existir un **mecanismo de control de flujo** para regular el flujo de información para evitar que un nodo rápido pueda saturar a uno lento. El control de flujo es distinto desempeña un papel clave en la capa de transporte siendo distinto el control de flujo entre nodos que el control del flujo entre enrutadores.

Dentro de este nivel se encuentran:

Norma	Contenido
X.214 (ISO 8072)	Servicio de Transporte
X.224 (ISO 8073)	*Especificación del protocolo de transporte*

Son funciones del nivel de transporte:

- Controlar el flujo de la información

- Establecer varias comunicaciones simultáneas para diferentes sesiones de nivel sesión, si fuera necesario
- Comprobar que la comunicación llega libre de errores al receptor
- Fragmentar la información que le llega de niveles superiores para darle aspecto de tramas de un tamaño adecuado para la red subyacente

Niveles OSI orientados a la Aplicación

3.5.5.- La capa de Sesión.-

La capa de sesión permite a los usuarios de máquinas diferentes establecer sesiones entre ellos. Una sesión permite el transporte ordinario de datos, como también lo hace la capa de transporte, pero, además, proporciona servicios mejorados que son útiles en algunas aplicaciones.

Uno de los servicios de la capa de aplicación es el **control de diálogo**, permitiendo que el tráfico vaya en las dos direcciones al mismo tiempo o en una sola dirección en un instante determinado, estableciendo un control de los turnos.

Un servicio de sesión relacionado es el **manejo de fichas**. Para algunos protocolos es esencial que en ambos lados de una comunicación no se intente la misma operación al mismo tiempo. Para controlar estas actividades, la capa de sesión proporciona fichas que se pueden intercambiar de manera que el que posea la ficha sea quien pueda realizar la operación crítica.

Otro servicio de sesión es la **sincronización**, de modo que, en una transferencia de datos, se inserten *puntos de verificación* en la corriente de datos, que permitan que después de cada interrupción únicamente se deban repetir los datos que se transfirieron después del último punto de verificación.

Las recomendaciones asociadas a este nivel son:

Norma	Contenido
X.215 (ISO 8326)	Servicio de sesión
X.225 (ISO 8327)	*Especificación del protocolo de sesión*

Las tareas realizadas por este nivel incluyen:

- Apertura y cierre de la sesión para llevar a cabo la conexión
- Intercambio de información en un sentido u otro
- Control de la sincronización del diálogo, evitando emisiones simultáneas de ambos comunicantes si el tipo de comunicación no lo permite.
- Identificación del usuario, mediante solicitud de palabras clave (contraseñas) y envío encriptado de ellas.
- Inserción de puntos de control en la transmisión para, en caso de fallo, retransmitir únicamente desde el último chequeo efectuado.

3.5.6.- La capa de Presentación.-

La capa de presentación realiza ciertas funciones que se realizan con suficiente frecuencia como para justificar la búsqueda de una solución general en lugar de dejar al usuario que resuelva sólo sus problemas. En particular, y a diferencia de todas las capas inferiores, que se interesan únicamente en mover bits de manera confiable del emisor al receptor, la capa de presentación *se ocupa de la semántica y la sintaxis de la información que se transmite*.

En particular, las diferentes computadoras tienen códigos diferentes para representar las cadenas de caracteres (por ejemplo, ASCII, Unicode, etc.), enteros (Complemento a uno, complemento a dos, etc.) y demás. Para hacer posible la comunicación entre ordenadores con representaciones diferentes, las estructuras de datos por intercambiar se pueden definir de forma abstracta, junto con un código estándar que se use "en el cable". La capa de presentación maneja estas estructuras de datos abstractas y las convierte de la representación que se utiliza dentro del ordenador a la representación estándar de la red y viceversa.

A este nivel corresponden:

Normas para vídeoText, TeleFax y TeleTex

X.225 del CCITT.

Son tareas desarrolladas por la capa de presentación:

- Codificación de datos y criptografía
- Compresión de los mensajes antes de su envío
- Manejo de distintos tipos de terminales virtuales en el mismo terminal físico, para poder ser usados con diferentes aplicaciones

3.5.7.- La capa de Aplicación.-

La capa de aplicación contiene varios protocolos que se necesitan con frecuencia y que permiten definir un **terminal virtual de red** abstracto, que los editores y otros programas puedan manejar. Para cada tipo de terminal real se debe escribir un programa que establezca la correspondencia entre las funciones del terminal virtual de red y las funciones del terminal real. Por ejemplo, cuando un editor mueva el cursor del terminal virtual a la esquina superior izquierda de la pantalla, este *software* debe emitir la secuencia apropiada de órdenes al terminal real para poner el cursor en dicho lugar. Todo este *software de terminal virtual* se encuentra en la capa de aplicación.

Otra función de la capa de aplicación es la transferencia de archivos. Los diferentes sistemas de archivos tienen convenciones diferentes para nombrar los ficheros, formas diferentes de representar líneas de texto, etc. La transferencia de un archivo entre dos sistemas diferentes requiere la resolución de éstas y otras incompatibilidades.

Además de la transferencia de ficheros, pertenecen a la capa de aplicación el correo electrónico, la carga remota de trabajos, la búsqueda en directorios, etc.

Pertenecen a este nivel, entre otras:

Norma	Contenido

X.400	Describe el modelo básico del sistema de tratamiento de mensajes en la aplicación de *Correo Electrónico*
X.500	*Servicio de* Directorio *en la aplicación de Correo Electrónico*

Las tareas que realiza la capa de aplicación son, entre otras:

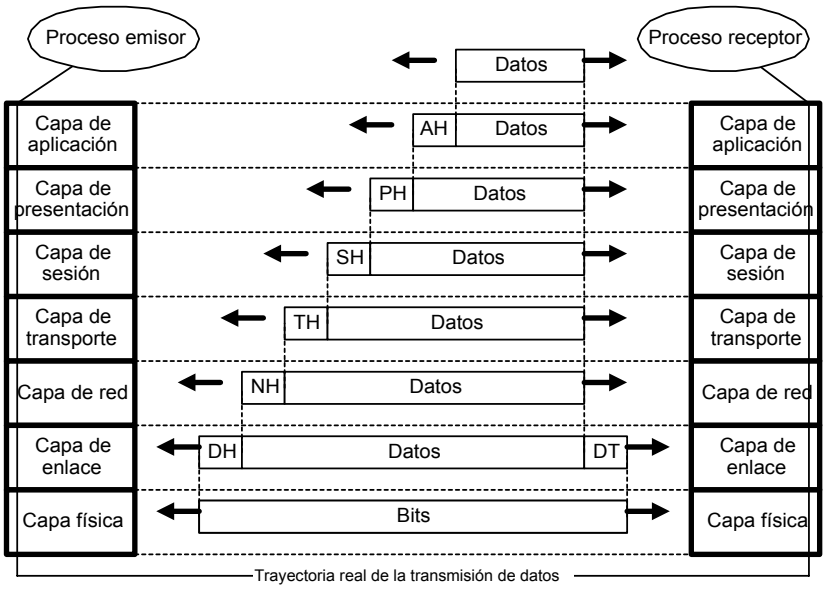
- Procesos de gestión del sistema, esto es, procesos que supervisan la actuación de los elementos de la red, por ejemplo, las prioridades de acceso
- Procesos de gestión de las aplicaciones. Controlan la utilización de la red por parte de las aplicaciones, incluyendo el acceso a partes del sistema, el bloqueo de recursos, la contabilidad y la facturación por el uso de los recursos.
- Procesos del sistema, como el acceso a ficheros de la red, la sincronización entre tareas, la activación de procesos derivados, etc.
- Procesos de aplicación. Son programas de usuario que consumen recursos de la red para su funcionamiento, como consultas a bases de datos, o procesadores de información situada en la red. Requieren, en ocasiones, protocolos específicos que se sitúan sobre el propio nivel de aplicación. Son también procesos de aplicación el correo electrónico o las utilidades para transferencia de ficheros.

3.5.8.- Transmisión de datos en el modelo OSI.-

La figura adjunta muestra un ejemplo de cómo se pueden transmitir datos empleando el modelo OSI. El proceso remitente tiene algunos datos que quiere enviar al proceso receptor, por lo que entrega los datos a la capa de aplicación, la cual añade entonces al principio el encabezado de aplicación **AH** (que puede ser nulo) y entrega el elemento resultante a la capa de presentación.

La capa de presentación puede transformar este elemento de diferentes maneras y, posiblemente, añadir al principio un encabezado, entregando el resultado a la capa de sesión. Conviene resaltar que la capa de presentación no sabe cuál porción de los datos entregados a ella por la capa de aplicación es la **AH** (si es que existe) y cuáles son en verdad los datos del usuario.

Este proceso se repite hasta que los datos alcanzan la capa física, donde son transmitidos realmente a la máquina receptora. En esta máquina se retiran los distintos encabezados, uno por uno, conforme el mensaje se propaga hacia arriba por las diferentes capas hasta que por fin llega al proceso receptor.



Transmisión de datos en el modelo OSI

La idea clave en todo este proceso es que aunque la transmisión real de los datos es vertical, cada capa se programa como si fuera horizontal. Así, por ejemplo, cuando la capa de transporte emisora recibe un mensaje de la capa de sesión, le añade un encabezado de transporte y lo envía a la capa de transporte receptora. Desde su punto de vista, el hecho de que, en realidad, debe dirigir el mensaje a la capa de red de su propia máquina es un tecnicismo sin importancia.

3.7.- Otras arquitecturas de red.-

3.7.1.- El modelo de referencia TCP/IP.-

La capacidad de conectar entre sí múltiples redes sin interrupciones propició el diseño de esta arquitectura, que posteriormente se popularizó con el nombre de modelo de referencia TCP/IP, por las iniciales de sus dos protocolos primarios.

Otro de los objetivos principales del diseño de esta arquitectura fue que la red fuera capaz de sobrevivir a la pérdida del hardware de subred sin que las conversaciones existentes se interrumpieran, es decir, que las conexiones permanecieran intactas mientras la máquina origen y la máquina destino estuvieran funcionando, aún si alguna de las máquinas intermedias o de las líneas de transmisión en el trayecto de una a otra dejara de funcionar de forma repentina.

El modelo ofrece pues una estructura flexible permitiendo el uso de aplicaciones con requerimientos divergentes, que abarquen desde la transferencia de ficheros hasta la transmisión de palabra hablada en tiempo real.

El desarrollo de TCP/IP tenía como objetivo conseguir la independencia frente a diversos parámetros:

- El hardware
- El Sistema Operativo
- La capa de enlace de la red
- El medio físico utilizado

Además, como consecuencia de la utilización mayoritaria de la red telefónica convencional, debe soportar altas tasas de error, y, por último, debe cumplir el requisito de poder elegir entre diferentes caminos, adaptándose a las condiciones de la red externa al sistema informático. Para ello utiliza la *descomposición de los datos en paquetes*, que pueden viajar por rutas diferentes de un sistema conmutado y ser ensamblados en el ordenador de destino para reconstruir el mensaje.

La principal ventaja del modelo TCP/IP es que puede interconectar redes muy heterogéneas formadas por máquinas diferentes, o bien redes de ordenadores cuyas características se desconocen. Dos máquinas distintas que utilizan protocolos TCP/IP pueden comunicarse mediante un **encaminador** (enrutador o router).

El modelo TCP/IP se construye de forma estructurada con las siguientes capas:

La capa de interred

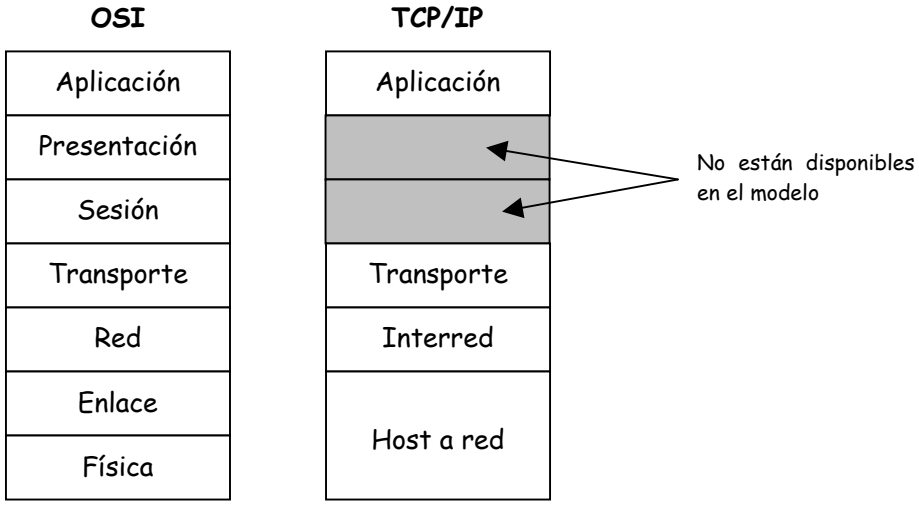
Los requerimientos anteriores condujeron a una red de conmutación de paquetes, basada en una *capa de interred carente de conexiones.*

Esta capa (**capa de interred**) es el eje que mantiene unida toda la arquitectura.

La misión de esta capa es permitir que los **nodos** inyecten paquetes en cualquier red y los hagan viajar de forma independiente a su destino (que podría estar en una red diferente). Los paquetes pueden llegar incluso en un orden diferente a aquel en que se enviaron, en cuyo caso, corresponde a las capas superiores reacomodarlos para una entrega ordenada.

La capa de interred define un formato de paquete y protocolo oficial llamado **IP** (*Internet Protocol* o **Protocolo de Interred**). El trabajo de la capa de interred es entregar paquetes IP a donde se supone que deben ir. La consideración mas importante es claramente el **ruteo** de los paquetes y, también, evitar la congestión.

La capa de interred del modelo TCP/IP es muy parecida en funcionalidad a la capa de red del modelo OSI como se ve en la siguiente figura:

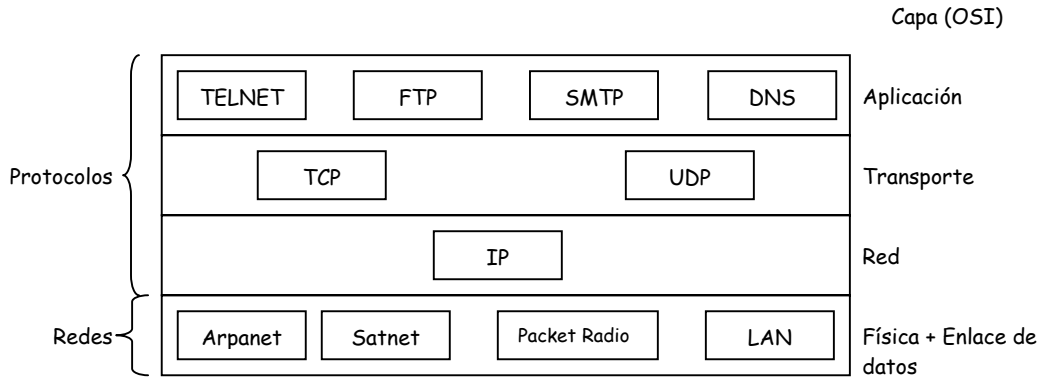


La capa de transporte

La capa que está sobre la capa de interred en el modelo TCP/IP se llama usualmente **capa de transporte**. Esta capa se diseñó para permitir que entidades pares en los nodos origen y destino lleven a cabo una conversación, lo mismo que la capa de transporte del modelo OSI.

En esta capa se definieron dos protocolos de extremo a extremo. El primero, **TCP** (*Transmission Control Protocol*) o **Protocolo de Control de la Transmisión** es un protocolo confiable orientado a la conexión, que permite que una corriente de bytes originada en la máquina emisora se entregue sin errores en cualquier otra máquina de la interred. Este protocolo fragmente la corriente entrante de bytes en mensajes discretos y pasa cada uno a la capa de interred. En el destino, el proceso TCP receptor reensambla los mensajes recibidos para formar la corriente de salida.

El segundo protocolo de esta capa es el **UDP** (*User Datagram Protocol* o **Protocolo de datagrama de usuario**) que es un protocolo sin conexión, no confiable, para aplicaciones que no necesitan la asignación de secuencia ni el control del flujo del TCP y que desean utilizar los suyos propios.



La relación entre los protocolos IP, TCP y UDP se muestra en la figura anterior.

La capa de aplicación

El modelo TCP/IP no tiene capas de sesión ni de presentación. Encima de la capa de transporte está la **capa de aplicación** que contiene todos los protocolos de alto nivel, entre los que se encuentran el *terminal virtual* (TELNET), el de *transferencia de archivos* (FTP) y el de *correo electrónico* (SMTP). Posteriormente se han añadido otros muchos protocolos, como el *servicio de nombres de dominio* (DNS) para relacionar los nombres de los nodos con sus direcciones de la red; el NNTP para transferir noticias o el HTTP que se utiliza para recuperar páginas en la *World Wide Web*; etc.

La capa del nodo de red

Bajo la capa de interred se encuentra un gran vacío, el modelo TCP/IP no indica lo que aquí sucede aparte de indicar que el nodo se conecta a la red utilizando algún protocolo tal que pueda enviar por la red paquetes de IP. Este protocolo no está definido y varía de un nodo a otro y de red a red.

3.7.- Introducción a las arquitecturas de redes de área local.-

Una red de área local (**LAN** *Local Area Network*) es un sistema de comunicaciones constituido por un hardware (cableado, terminales, servidores, etc.) y un software (acceso al medio, gestión de recursos, intercomunicación, etc.) que se distribuyen por una extensión limitada (edificio, grupo de edificios, etc.) en el que existen una serie de recursos compatibles (discos, impresoras, bases de datos, etc.), a los que tienen acceso los usuarios para compartir información de trabajo.

La interconexión entre varias LAN, o entre LAN y WAN se realiza por medio de *repetidores* (repeaters), *puentes* (bridges), *encaminadores* (routers), *pasarelas* (gateways) o *conmutadores* (switches).

Toda red de área local viene caracterizada por:

- **Modo de transmisión / modulación** (banda base o banda ancha)
- **Protocolo de acceso** (TDMA, CSMA/CD, Token Passing, FDI)
- **Soporte físico** (par trenzado, coaxial, fibra óptica, etc.)
- **Topología** (bus, anillo, estrella, malla)

Características que se detallan a continuación.

3.7.1.- Soporte físico de las LAN.-

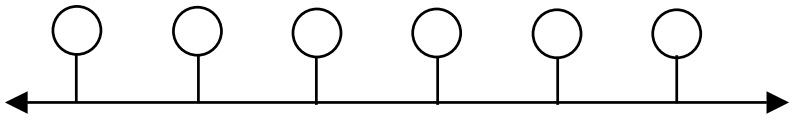
Para constituir una LAN se utiliza como elemento básico un **sistema físico** consistente en un cableado que distribuye las señales entre todos los equipos conectados a la misma. Este cableado utilizado (coaxial, par trenzado, fibra óptica, etc.) presenta una serie de características – ancho de banda, facilidad de conexión, etc. Que determinan, entre otras cosas, la velocidad a la que puede circular la información, el número de estaciones de trabajo que pueden conectarse y la distancia máxima las que éstas pueden estar.

3.7.2.- Topologías de las LAN.-

Existen básicamente cuatro topologías diferentes para la construcción de una red de área local:

Topología en BUS

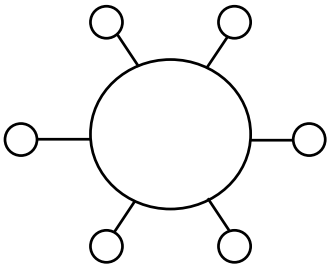
Es la topología mas simple, en la que un único tendido, mediante derivaciones, da servicio a todos y cada uno de los terminales, por lo que, en caso de fallo, una parte de la red queda siempre sin servicio.



Suele emplearse para esta topología cable coaxial, y el ejemplo mas típico de la misma lo constituyen las redes Ethernet. La estructura puede complicarse añadiendo ramificaciones hasta llegar a formar un árbol (**topología en árbol**).

Topología en ANILLO

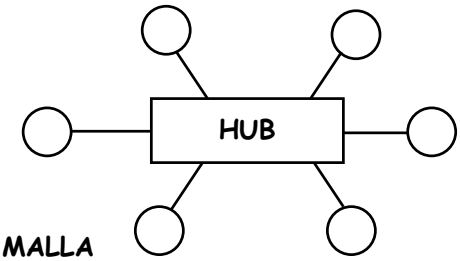
Es una variante de la topología en bus, en la que éste se cierra sobre sí mismo, por lo que, en caso de rotura, se puede acceder a las estaciones aisladas por el otro semianillo.



En la práctica, la mayoría de las topologías en anillo (lógica) acaban siendo una estrella física. Pueden emplearse cables de pares, coaxiales o fibra óptica. Esta topología encuentra su ejemplo mas significativo en las redes Token Ring.

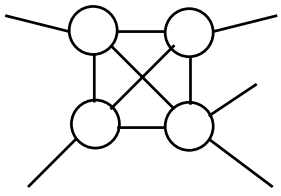
Topología en ESTRELLA

En esta topología, un elemento central (**HUB**) sirve de puente entre todos los terminales de la LAN, proporcionando la conmutación entre ellos. Aisla unos elementos del fallo de otros, pero presenta como punto crítico el nodo central, que, en caso de fallo, deja la red sin servicio. El coste del cableado es elevado al requerir conexiones puntop a punto para todos los elementos, aunque éste se minimiza empleando cable par trenzado.



Topología de MALLA

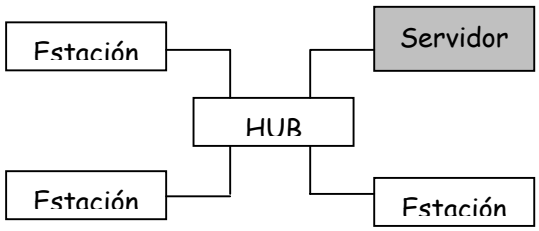
Es la topología que ofrece un mayor nivel de seguridad. Los nodos de la red se unen entre sí formando una estructura en la que al menos existen dos rutas posibles en cada nodo; así, si hay un fallo en una de ellas la información puede hacerse circular por la otra. Es una topología adecuada para cubrir, por ejemplo, un país completo. En particular, es la red que utiliza Telefónica para su red Iberpac.



Topologías Física y Lógica

Todas las configuraciones vistas hasta ahora son llamadas **topologías físicas** porque describen como está extendido el cableado. Además, cada red designa una **topología lógica** que describe la red desde la perspectiva de las señales que viajan por ella. Un diseño de red puede tener distinta tipología física y lógica, esto es, que la forma en que esté cableada la red no tiene por qué reflejar necesariamente la forma en que viajan las señales por ella.

La figura:

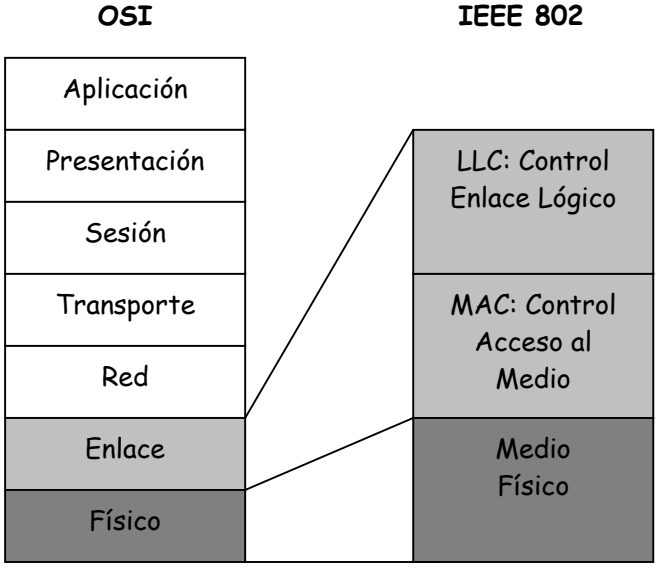


Muestra una disposición física de topología en estrella. Cada estación envía y recibe señales por el mismo cable. En el concentrador (HUB) se mezclan las señales de todas las estaciones y son transmitidas a todas ellas, esto es, actúa como si estuviera definida una topología en bus, por tanto, es una *topología física en estrella* que funciona como una *topología lógica en bus*.

Muchas redes utilizan este modelo ya que es fácil de modificar la situación de cada estación (sólo hay que desconectar un cable) sin perjuicio de la red entera y, además, incrementa las posibilidades de detección de problemas en la red

3.7.3.- Los niveles en la LAN.-

Puede establecerse una correspondencia entre el modelo OSI y los estándares del IEEE (IEEE – 802) para redes locales tal y como se indica en la siguiente figura.



EL MEDIO FÍSICO

El nivel físico define las características mecánicas, eléctricas, funcionales y de procedimiento necesarias para conseguir que las tramas de bits que la capa física recibe del nivel de enlace, su capa inmediatamente superior, puedan ser emitidas por los medios de transmisión en forma de señales.

La capa utiliza una gran cantidad de recursos propios de las transmisiones de señales:

- Medios de transmisión de la señal (cable par trenzado, coaxial, etc.)
- Transmisiones analógicas a través de líneas telefónicas utilizando módems con diferentes técnicas de modulación
- Transmisiones digitales a través de redes digitales de transmisión de datos, utilizando técnicas de modulación digital (impulsos codificados, modulación delta, etc.)
- Técnicas de multiplexación en el tiempo y en la frecuencia
- Técnica de concentración de canales
- Técnicas de conmutación: de circuitos, de mensajes, de paquetes
- Transmisión en serie o en paralelo
- Transmisión síncrona o asíncrona

EL NIVEL DE ENLACE

La capa de enlace asegura una conexión libre de errores entre dos ordenadores de la misma red. Fundamentalmente organiza los bits en forma de tramas y los pasa a la capa física para que sean transmitidos al receptor.

La capa de enlace tiene dos funciones:

- El control de acceso al medio (MAC)
- El control de enlace lógico (LLC)

Cada una de estas funciones da origen a una subcapa.

La subcapa MAC

La subcapa de control de acceso al medio es muy importante en las redes de área local, ya que la mayoría de ellas utiliza un canal común - *canal de acceso múltiple* - como base para sus comunicaciones, a diferencia de redes de área extendida que suelen utilizar enlaces punto a punto.

La principal función de esta subcapa consiste en establecer la forma para determinar quien tiene derecho sobre ese canal compartido por todos los equipos conectados a la misma red.

El subnivel MAC facilita pues al subnivel de Control de Enlace lógico (LLC) un medio de comunicación "aparentemente propio". El subnivel MAC depende de la topología del medio, puesto que ésta influye en la política de acceso, facilitando al LLC y capas superiores un servicio independiente totalmente del medio (tanto topológica como tecnológicamente).

Para establecer esta política de acceso, el subnivel MAC parte de las siguientes hipótesis:

- **1°.- Modelo de estación.-** Formado por n estaciones independientes, de forma que en una de ellas, una vez establecida la trama del mensaje a enviar, se espera hasta que no se haya transmitido con éxito. Las estaciones son por tanto independientes y el trabajo se genera a ritmo constante.
- **2°.- Hipótesis de un sólo canal.-** Se supone que hay un sólo canal utilizado por todas las estaciones, aunque pueden asignarse prioridades a la hora de transmitir, dando mas importancia a unas estaciones que a otras.

- **3°.- Hipótesis de colisión.-** Si dos estaciones transmiten sendas tramas simultáneamente en el mismo canal, se producirá una colisión que generará una interferencia de la señal. Todas las estaciones pueden entonces detectar las colisiones habidas en el canal. Si dos tramas colisionan, ambas deberán ser retransmitidas por las estaciones que las generaron.
- **4°.- Tiempo continuo y ranurado.-** En tiempo continuo, la transmisión de la trama puede comenzar en cualquier instante, sin existir ningún organizador del tiempo de la red. En tiempo ranurado, el tiempo de la red se divide en intervalos (*ranuras*), de forma que las estaciones utilizan dichas ranuras para transmitir sus tramas.
- **5°.- Detección de portadora.-** Cada estación puede escuchar si hay o no señal portadora en el canal. Si no la hay, la estación podrá transmitir si lo desea. En caso contrario, deberá esperar hasta que se desocupe el canal. En el caso de que no exista detección dela portadora, la estación que emite la trama sólo puede saber si el canal estaba libre cuando se puso ésta en el canal.

Distintas combinaciones de estas hipótesis conducen a sistemas distintos de establecimiento de las características de acceso al medio de transmisión. Una vez elegida una solución concreta, se dice que se ha establecido un **sistema de contienda.**

El subnivel MAC participa además en el formato del mensaje en dos maneras:

- Inserta los delimitadores (de inicio DI y de fin DF) del mensaje
- Añade campos orientados al control de acceso (CA).

Los métodos de acceso al medio (MAC) mas utilizados son el *método Aloha*, por razones históricas, método *de acceso múltiple con detección de actividad y colisión* (CSMA/CD Carrier Sense Multiple Access/Collision Detect) y el *método de acceso por paso de testigo* (Token Passing).

El método Aloha

El protocolo aloha permite que cualquier estación que tenga datos que transmitir lo haga inmediatamente.

Ello puede provocar colisiones con otras estaciones que también han iniciado una transmisión. El sistema de contienda generado es el siguiente: Cuando se produce la colisión, la estación la escucha sin mas en el canal; si lo que escucha no es lo que ella puso, es que alguien mas ha puesto una señal, y, por tanto, se ha producido una colisión. En este caso, las estaciones esperan un cierto tiempo y vuelven a transmitir las tramas que colisionaron.

El rendimiento de este sistema es muy bajo y especialmente crítico al crecer el número de estaciones, al incrementarse, en gran medida, las probabilidades de colisión.

El método CSMA

Este tipo de métodos (*Carrier Sense Multiple Access* o Acceso Múltiple con Escucha de Portadora y Detección de Colisión) permiten el acceso múltiple a un único canal, comprobando si el canal está libre por detección en él de señal portadora. Es el protocolo de acceso que utilizan las redes Ethernet

Los protocolos CSMA llevan asociado un **índice de persistencia *p*** cuyo valor está representado por un número real comprendido entre 0 y 1, que indica la probabilidad de envío. El funcionamiento de este protocolo es el siguiente:

Cuando un terminal desea transmitir se pone a la escucha del canal para determinar si está o no está libre. En el caso de que el canal esté libre puede efectuar la transmisión, pero si está ocupado, debe esperar a que éste se libere, detectándolo automáticamente si permanece a la escucha. Cuando efectivamente se libere la estación emitirá su trama con probabilidad p.

Existe también un protocolo **CSMA no persistente.** Cuando la estación escucha el canal, si éste se encuentra ocupado, deja de escuchar, y después de un tiempo aleatorio intenta de nuevo la transmisión.

Los protocolos CSMA se ocupan de disminuir el número de colisiones tanto como sea posible. No obstante con las técnicas del CSMA estas son inevitables.

La técnica CD (Collision Detect) del protocolo **CSMA/CD** implica que las estaciones permanezcan a la escucha mientras transmiten sus tramas. Si reconocen una colisión (es decir si lo que transmiten no es lo mismo que escuchan) suspenden inmediatamente la transmisión.

El método Paso de Testigo

El método de acceso por Paso de Testigo (*Token Passing*) se utiliza en diferentes redes (con pequeñas variantes) que disponen de un anillo lógico: Token Ring, Token Bus y FDI. Al contrario que en el método anterior, un terminal de la red puede transmitir en un tiempo fijado. Esto es: únicamente tiene derecho a utilizar el medio momentáneamente la estación – en cada momento sólo una – que dispone del **testigo** (que suele ser un byte formado por 8 unos (11111111), utilizándose técnicas de relleno de bit para evitar que esta secuencia aparezca en un mensaje), resolviéndose de esta manera el problema de congestión de acceso.

La política de acceso se establece haciendo que el testigo vaya pasando de manera secuencial de una estación a otra, controlando a su vez el tiempo máximo de pertenencia, dando así posibilidad a todas las estaciones de hacer uso del medio. (Formando un anillo lógico).

La mecánica de utilización del testigo es la siguiente:

El método de paso por testigo se vale de una trama especial o *testigo* (**token**) que va a ser monitorizado por cada ordenador, para dar a éstos permiso o no de transmisión. En definitiva, los ordenadores conectados al anillo lógico no pueden transmitir datos hasta que no obtengan el permiso para hacerlo.

Si el testigo está libre (no existe ninguna estación que esté transmitiendo), cualquier ordenador que tenga necesidad de transmitir, pasará el testigo al estado de *ocupado* e iniciará la comunicación insertando los datos detrás del testigo. En ese momento, el propietario del testigo es la estación que está transmitiendo, siendo ésta la que dispone del control absoluto del anillo. La trama resultante pasará por cada terminal, regenerándose en el camino hacia el terminal destinatario de los datos.

Una vez la trama ha llegado al ordenador destino, se copia en la memoria de éste, pasando a retransmitir la trama sobre la red, cambiando una serie de bits de forma que la máquina que envió la información comprueba que el terminal destino la recibió correctamente. De ser este el caso, el terminal se encarga de liberar el testigo de forma que otros ordenadores puedan realizar sus comunicaciones.

En el caso de que el terminal destino no hubiera recibido correctamente la trama, la máquina origen de la comunicación volvería a retransmitir la trama.

En resumen, sus principales características son:

- Durante el periodo de pertenencia del testigo no se prescribe que un subconjunto de estaciones no pueda hacer uso de otras técnicas (polling, CSMA/CD, etc.) de acceso al medio.
- Responde igualmente bien tanto en situaciones de carga elevada como en situaciones de baja utilización.
- Proporciona un reparto equitativo de la capacidad del medio.
- El retardo máximo en el acceso puede ser acotado determinísticamente (tiempo máximo de pertenencia del token multiplicado por el número de estaciones)
- El coste de los nodos (adaptadores al medio) a utilizar es bajo debido a la sencillez de los mismos.

La subcapa LLC

La subcapa de **Control del Enlace Lógico**, en colaboración con la subcapa MAC, se encarga de garantizar la comunicación entre emisor y receptor, sin errores de las tramas construidas con la información recibida del nivel de red. Proporciona tres servicios:

- Servicio sin conexión y sin confirmación
- Servicio sin conexión y con confirmación
- Servicio con conexión

Servicios que ya han sido comentados anteriormente.

La **funcionalidad** del subnivel LLC de una red de área local es pues transferir la unidad de servicio de datos correspondiente (SDU) al subnivel (o subniveles) de enlace lógico del terminal o terminales destino de los datos (DTE) sin errores. Por ello, formatea la unidad de servicio de datos con:

- Un **campo de dirección** (CDIR), para determinar el destino o destinos del mensaje.
- Un **campo de control** (CC) para indicar el tipo de mensaje o realizar un control de flujo.
- Un **campo de bits de redundancia cíclica** (CRC) para detección de errores de transmisión del mensaje del nivel de enlace lógico.

El formato de un mensaje del nivel de enlace lógico es:

<CDIR> <CC> <SDU_LLC> <CRC>

En caso de recepción errónea del mensaje en el destino, se corrige el error con una retransmisión. Los mensajes del subnivel LLC incorporan un **conjunto de bits redundantes**

(código cíclico, normalmente de 32 bits) que permite, con una elevada fiabilidad, detectar los errores en la transmisión.

Obsérvese que ni los delimitadores ni el campo de control de acceso, quedan cubiertos por este código cíclico, que sólo afecta a los campos <CDIR> <CC> <SDU_LLC>.Cuando se detecta algún error en la transmisión se puede recuperar mediante una petición de retransmisión (ARQ), como se realiza habitualmente en los protocolos de nivel de enlace.

Con el subnivel LLC son posibles tres tipos de direccionamiento:

- **Individual** en donde el destinatario es único.
- **Grupo** en donde el destinatario es un subconjunto de las estaciones de la red.
- **Broadcast** en donde todas las estaciones de la red son destino

Protocolos de las Redes de Área Local

Tomando como referencia el modelo OSI, los protocolos situados por encima del nivel de enlace que se disponen en una LAN no están normalizados por el IEEE ni por ningún otro organismo. De forma que los fabricantes han diseñado sus propios protocolos. Por ejemplo, NOVELL ha creado los, protocolos IPX para el nivel de red, SPX para el nivel de transporte y NCP que implementa los niveles de sesión, presentación y aplicación.

La función de este tipo de protocolos es equivalente a la que ofrecen los protocolos del modelo OSI. Dentro de los protocolos mas extendidos en las empresas, destacan los ya indicados de Novell para sus redes NetWare y los creados para ARPANET, la precursora de la actual Internet: TCP/IP.

La mayor diferencia entre estos dos estándares "de facto" es que en un entorno NetWare los servidores y los clientes son **dedicados** (de ficheros, de aplicaciones, de impresión, etc.). Los servidores de Novell NetWare no funcionan como clientes y los clientes no funcionan como servidores. En un entorno TCP/IP los ordenadores pueden ser clientes, servidores, y clientes y servidores a la vez. Existen además otras diferencias entre estas dos arquitecturas de comunicaciones, ya que fueron desarrolladas para dos entornos muy diferentes.

Según el protocolo, se empleará una u otra técnica (contienda o selección) de acceso al medio, ya que la red local es un recurso compartido. Las técnicas basadas en una estrategia de contienda, aunque mucho mas adecuadas para demandas a ráfagas, resultan con frecuencia mas difíciles de controlar, ya que la asignación de recursos varía con el tiempo, por actuar bajo criterios de demanda. Son técnicas de asignación dinámicas, a diferencia de la selección, que utiliza técnicas fijas o estáticas.

El problema del acceso a un recurso informático en un sistema de procesamiento distribuido, o el de comunicación entre sistemas distintos conectados a través de una LAN, es básicamente un problema de acceso y utilización de un recurso de comunicaciones. Ya que el tráfico a que se destinan las redes locales suele ser un tráfico a ráfagas, las técnicas mas eficientes suelen ser las de contienda. Para un número de usuarios reducido,

puede no estar justificado el mayor rendimiento que se obtiene en el uso del recurso, con las técnicas de contienda, frente a la mayor sencillez de las técnicas de selección.

Debe considerarse además, que las técnicas de selección, aunque puedan producirse grandes retardos en obtener todo el servicio solicitado, el acceso al recurso es determinístico y el tiempo de espera máximo está acotado y es conocido. En las de contienda, el acceso es aleatorio. El tiempo de espera depende de la carga del sistema y puede llegar a ser muy grande al no estar acotado. Sin embargo, la probabilidad de que esto ocurra en un sistema adecuadamente dimensionado es muy pequeña.