Ampliación y aclaración de conceptos:

Caracterización de Redes y Arquitecturas de comunicaciones

Sumario

Documento realizado por M.ª Ángeles Caro Bartolomé y Jesús Manuel Marín Navarro

1 Definiciones

1.1 Host

Llamaremos host a un ordenador capaz de interactuar en red, capaz de alojar algún servicio de la misma. Serán un host un PC, un servidor, etc.

1.2 Sistemas aislados

Un sistema aislado es un ordenador incapaz de comunicarse con el exterior.

A un sistema aislado se le debe añadir el hardware (tarjeta de red, etc) y el software (driver, protocolos de red, etc) necesarios para poder operar en red.

En ocasiones, los sistemas aislados pueden efectuar conexiones temporales (por ejemplo, cuando nos conectamos usando un módem).

1.3 Redes de ordenadores

Varios ordenadores conectados entre sí pero sin perder su identidad propia.

1.4 Sistema distribuido

Es una red de ordenadores en la que cierta o ciertas tareas se comparten y se realizan entre todos. P.ej: cálculos astronómicos o investigaciones médicas en las que se necesita gran capacidad de cálculo, este cálculo lo realizan muchos ordenadores de manera independiente y luego se comunican para intercambiarse datos o resultados y seguir trabajando (hay algunos salvapantallas que te permiten esto).

En los sistemas distribuidos, la existencia de éstos múltiples hosts es transparente al usuario que no sabe si la tarea que realiza se está llevando a cabo en el ordenador ante el cual está sentado o en una máquina remota.

(En una red cada ordenador trabaja por separado, en un sistema distribuido todos los ordenadores trabajan para el mismo fin).

2 Conceptos previos sobre arquitecturas de redes

2.1 Protocolo de comunicaciones

Un protocolo es un conjunto de reglas perfectamente organizadas y convenidas de mutuo acuerdo entre los participantes de una comunicación cuya misión es la de regular algún aspecto de dicha comunicación.

Es habitual que los protocolos se estandaricen y sean dados como normativas o recomendaciones de asociaciones de estándares. Los fabricantes que se ajustan a estas normativas están seguros de ser compatibles entre sí.

Estándar de facto: Protocolo que todo el mundo utiliza a pesar de que ninguna asociación lo haya normalizado.

Estándar real: hecho por alguna asociación de normalización, a partir de estándares de facto en muchas ocasiones. En los primeros años era muy habitual que las organizaciones de estandarización sacaban un documento con el estándar basándonse en el protocolo que ya estaba todo el mundo usando, lo que ocurría era:

1) Fabricante crea protocolo.

2) Protocolo se extiende.

2.1) Cuando todo el mundo lo usa se convierte en un estándar de facto.

3) Asociación de normalización ---> crea el estándar (la RFC)

No siempre se crea un estándar a partir de uno de facto, sobre todo en los últimos años en los que las asociaciones de estandarización trabajan a la vez que los fabricantes y no a remolque de éstos como en los años 80 y 90.

Algunas asociaciones en el mundo de las telecomunicaciones son IEEE, ISO, w3c, etc.

2.2 Capa o nivel

En una comunicación entre dos equipos hay que tener en cuenta muchos factores y ponerse de acuerdo en muchos puntos. Con el fin de simplificar la complejidad del asunto, se han agrupado conjuntos de funciones en “capas”.

Las capas están jerarquizadas, la capa N proporciona una serie de servicios a la capa que tiene por encima sin que ésta sepa los pormenores de cómo se da dicho servicio. A su vez, esta capa N se sirve de la capa inferior N-1, usando los servicios que ésta le ofrece.

El número de capas y las funciones dentro de cada capa depende de la arquitectura que tomemos.

En una capa o nivel se agrupan funciones similares.

2.3 Interfaz entre capas

De cada capa se definen qué servicios ofrece a la capa superior y qué servicios puede coger de la inferior pero no se sabe cómo se hacen esos servicios. Es decir, una capa es “opaca”, de ella sólo se ve su “interfaz” con el exterior, lo que da y toma de las capas anterior y posterior.

2.4 Arquitectura de red

Una arquitectura de red debe definir:

- un determinado número de capas,

- las funciones que realiza cada una de esas capas
- las interfaces entre capas (es decir, cómo interactúan las capas adyacentes entre sí)
- los protocolos de comunicación entre capas homónimas.

Las capas, su número y función deben quedar claras para que los fabricantes puedan hacer productos que den los servicios que indica cada capa. La arquitectura no dice cómo implementar el servicio, solo da las características de la interfaz: es decir, cómo dar dicho servicio a las capas adyacentes.

Por ejemplo, si tú fabricas una tarjeta de red que cumple el estándar 802.11n, IEEE no define cómo tiene que ser la tarjeta de red, lo que define es cómo debe enviar la señal, los bits, la trama, etc, para que sea compatible con otras tarjetas compatibles con el estándar WiFi 802.11n.

Los protocolos indican cómo se comunican las capas de igual nivel entre sí: la capa N del emisor se comunicará con la capa N del receptor utilizando el protocolo de capa N que defina la arquitectura en cuestión. Cada capa tendrá sus protocolos para comunicarse con su capa homónima en el receptor.

Por ejemplo, si tengo dos tarjetas de red Ethernet, éstas se comunicarán usando el estándar 802.3, a nivel 2 según el protocolo Ethernet y a nivel 3 según el protocolo IP. Si dejo de usar IP en una de las dos tarjetas, también tendré que dejar de usarlo en la otra.

2.5 Ventajas de los modelos por capas

Flexibilidad, facilidad de mantenimiento: tanto de reparación de fallos como de mejoras.

Podemos implementar un producto que implemente varias de las capas y sólo tendremos que ser compatibles en cuanto a las interfaces externas, por arriba y por abajo. También tendremos que implementar los protocolos que usemos de dichas capas para podernos comunicar con otros productos que también implementen dicha arquitectura.

Podemos cambiar la implementación de una capa (por mantenimiento, reparación de fallos y/o mejoras) que esto no afectará a las capas adyacentes ya que verán los mismos servicios: la interfaz no cambia.

Suponemos dos PCs que se comunican usando dos tarjetas de red Ethernet de distinto fabricante.

Las dos tarjetas tienen que enviar los mismos voltajes la una a la otra y por los pines adecuados. Esto es porque los dos fabricantes se ajustan a la norma de transmisión por un cable Ethernet, que es un estándar de la organización IEEE. Es decir, a nivel físico, las dos tarjetas hablan el mismo protocolo, siguen las mismas normas de nivel físico.

Además de que los voltajes, intensidad, etc sean los mismos en las dos tarjetas, tenemos que ponernos de acuerdo en utilizar un mismo protocolo de comunicaciones, por ejemplo: IP. Las dos tarjetas tiene que tener dirección IP, es un requisito que las dos usen el protocolo IP.

Suponemos que un PC es de Windows y el otro tiene Ubuntu. Aún así, las dos son capaces de comunicarse, ¿por qué? Porque se rigen por las mismas normas y hablan los mismos protocolos de comunicaciones.

Si cambiamos de tarjeta de red en uno de los PCs y ponemos la de otro fabricante, nosotros, en nuestro sistema operativo, seguimos viendo los mismos parámetros: se configuran igual. Esto es porque las dos tarjetas, aunque internamente sean distintas, ofrecen las mismas características al nivel superior (que es IP).

3 Arquitecturas de red existentes

3.1 SNA

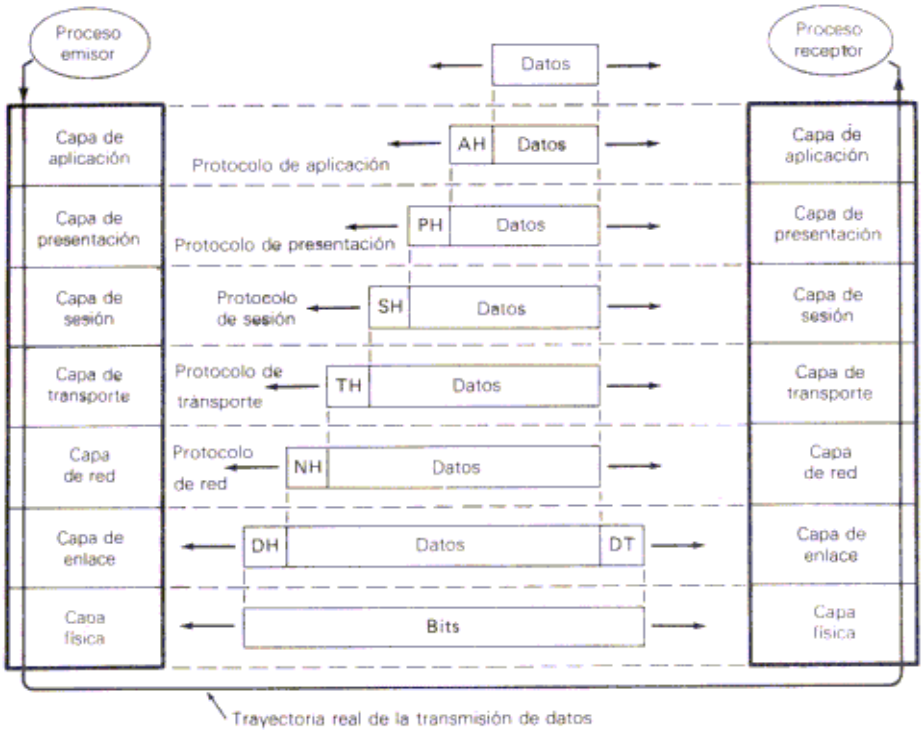
Fue la primera arquitectura importante, la de las máquinas de IBM.

Tenía 7 capas y era bastante compleja: tenía muchas funciones que repartía en las 7 capas. Nació en el 74 en un intento de facilitar la fabricación de productos compatibles con IBM (que eran casi los únicos hasta el momento).

Antes los productos de IBM tenían una estructura cerrada, no se sabía cómo comunicar mi equipo con un IBM (que era la mayor parte del mercado). Después de sacar este estándar "de facto" los equipos IBM eran "abiertos", cualquiera que implementara los protocolos y las capas de SNA podía comunicarse con un equipo IBM.

3.2 OSI

Se basa en las 7 capas de SNA. Es complejo, tiene muchas funciones que reparte entre sus 7 capas.



.

Las capas 5, 6 y 7 están orientadas a la aplicación, es decir, implementan funciones para que no tenga que implementarlas cada aplicación por separado: por ejemplo, un navegador se puede servir de un protocolo de capa 7 como http para comunicarse en vez de inventarse dicha aplicación otro protocolo; Se puede servir de las funciones de encriptación que le ofrece la capa 6 en vez de inventarse dicho navegador la forma de enviar datos por Internet de manera segura, etc.

Estas tres capas implementan tareas que facilitan la vida a las aplicaciones.

Las capas 1, 2, 3 y 4 están orientadas a la red, a las comunicaciones. Se encargan de que los mensajes que les entregan lleguen tal cual al otro extremo, sin errores y en un tiempo razonable.

3.3 TCP/IP

El modelo TCP/IP no es una arquitectura como tal ya que no especifica el número de capas, sus funciones, las interfaces entre capas, etc, de manera muy ortodoxa.

En esta arquitectura primero se hicieron los protocolos de comunicaciones y luego se enmarcaron éstos en varios niveles:

- el nivel de aplicación, donde se desarrollaron protocolos como http, ftp, smtp, etc.
- el nivel de transporte, donde se utilizaban TCP y UDP.
- El nivel de internet, que usa IP.
- El nivel físico, del cual este modelo no dice nada (es decir, estos protocolos pueden funcionar en teoría sobre cualquier medio de transmisión, de hecho existe IP sobre palomas mensajeras).

Cuando tenemos unos datos que transmitir de emisor a receptor, estos datos son entregados a la capa de aplicación.

Esta capa debe añadir al paquete una información para comunicarse con el nivel de aplicación del receptor, para ello añade a los datos una cabecera y le entrega el paquete (datos+cabecera) completo al nivel inferior.

Suponemos que estamos rellenando un formulario en una página web. Los datos que nosotros escribimos necesitan ser encapsulados en un paquete http para que puedan ser enviados por Internet y que el servidor web los entienda.

El nivel de transporte necesita también añadir una cabecera para así poder transmitir información de control necesaria para el nivel de transporte del receptor.

Es decir, el paquete http no puede ser enviado así como así, necesita que tenga una dirección IP y un número de puerto para que llegue al servicio correcto en el servidor correcto. Para esto, el paquete http se "encapsula" añadiéndole una cabecera de N4 que tiene información necesaria para el N4 del receptor.

Sucesivamente cada capa va añadiendo su cabecera, que será leída e interpretada sólo por la capa homónima en el receptor.

El receptor recibirá el paquete completo, con las cabeceras de todas las capas: el nivel 1 comprobará la señal que recibe y le pasará un chorro de unos y ceros al nivel 2 para que lo interprete.

El nivel 2, leerá la cabecera de nivel 2 y, entre otras funciones, comprobará que el paquete no tiene errores. Luego pasará el resto del paquete (todo menos la cabecera de N2) al nivel 3, que hará lo mismo: leerá e interpretará la cabecera de N3 y le pasará el resto al N4. Etc...

3.3.1 Correspondencia entre los niveles OSI y TCP/IP

3.3.2 Arpanet

Era una red de ordenadores del departamento de defensa de los EEUU en los setenta.

El departamento necesitaba una arquitectura que permitiera la comunicación entre sus ordenadores.

En esta red Arpanet se utilizaron los protocolos IP, TCP y UDP que actualmente se usan en Internet.

Estos protocolos fueron también utilizados por Unix para conectar sus sistemas y se utilizaron en los primeros ordenadores que formaron parte de Internet (al principio universidades en EEUU). Luego surgieron otros protocolos aparentemente "mejores" pero se optó por seguir utilizando éstos porque "funcionaban" y, además, lo hacían en Unix, que tenía fama de fiable.

4 Modelo OSI

4.1 Nivel 1 o físico

Trabaja con bits. Lo que le importa es que los ceros y unos que salen de un lado del cable o medio de transmisión que usemos, lleguen al otro lado.
Define las características mecánicas y eléctricas como:

- medio de transmisión: si es cable de pares, o fibra, coaxial, radio, satélite, etc.

- conectores: tipo, forma, tamaño, nº de pines y lo que significa cada uno.
- Parámetros de voltaje o intensidad que se van a usar, duración de los pulsos
- modulación
- Tipo de transmisión: serie o paralelo, síncrona o asíncrona, analógica o digital...
- Tipo de multiplexación (es decir, si viajan más de una señal por el mismo medio, cómo se reparten el medio): en tiempo (esto lo hacen los móviles, cada uno emite en un slot de tiempo, no lo hacen todos a la vez), frecuencia (así se hace en el cable de pares por el que viaja la señal de teléfono y la de ADSL, cada una a una frecuencia diferente; o también en las diferentes emisoras de radio, cada una emite en una frecuencia distinta, pero todas a la vez) o longitud de onda (se usa en fibra óptica).
- etc...

Por ejemplo, el nivel físico de Ethernet en la norma 802.3e define que:

se usa cable de pares

Conectores RJ45

No hay multiplexación

La transmisión es digital, síncrona y serie (no paralelo).

4.2 Nivel 2 o de enlace

Trabaja con tramas. Agrupa los bits en tramas y las envía por el medio de transmisión.

Algunas de sus funciones:

Que la transmisión sea sin errores, para ello añade un campo en su cabecera que puede ser un CRC o un checksum de la trama.

Agrupa los datos que le pasa el nivel 3 en tramas con el tamaño adecuado al medio de transmisión que usemos. P.ej. El nivel 3 entrega al nivel 2 un paquete de 2000 bits y el medio de transmisión soporta tramas de máximo 1500 bits, el nivel 2 se encargaría de reagrupar los bits y transmitir en dos paquetes, por ejemplo, uno de 1500 y otro de 500 bits.

Cada nivel 2 establecerá su tamaño máximo de trama o MTU (Maximum Transfer Unit). En Ethernet es 1500 bytes.

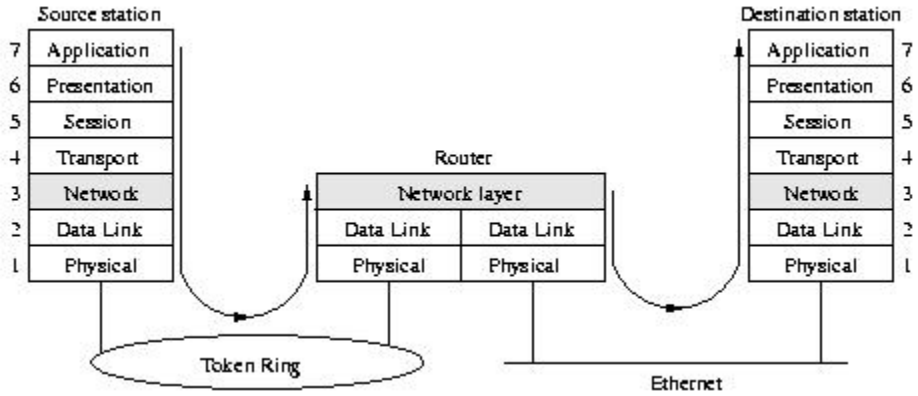
Otra función del nivel 2 es que determina en qué momento podemos acceder a un medio compartido porque está libre. P.ej. En esta capa se define cómo se va a organizar la comunicación de varias estaciones asociadas a un punto de acceso WIFI. Todas ellas no pueden transmitir a la vez, el N2 decide quién transmite, en qué orden, cuánto transmite, etc.

4.3 Nivel 3 o de red

Trabaja con paquetes.
Funciones:

- Se encarga de la identificación inequívoca del destino. P.ej. Con la dirección IP, es un número que identifica a una estación y que está en la cabecera del nivel 3.
- Se encarga del encaminamiento o enrutamiento, es decir, decidir por dónde se alcanza un destino en caso de que haya más de una ruta para llegar a él.

En este dibujo vemos la comunicación entre dos estaciones, atravesando un router. Éste permite pasar de una red Token Ring, con sus características y su cabecera, a otra Ethernet, con distinta MTU, distinto direccionamiento MAC, etc:



Como se ve en el dibujo, los PCs tienen todos los niveles de la comunicación, leen e interpretan todas las cabeceras de todos los niveles. Sin embargo, un router sólo comprueba los niveles 1, 2 y 3. Es decir, sólo comprueba que lo que acaba de recibir sera correcto en cuanto a voltajes, intensidad, etc (N1), que sea correcta la trama de N2 que no tenga errores, que no sea mayor que lo que permite la norma. También comprueba la dirección Ip del paquete (nivel 3) pero no comprueba si lo que enviamos de un sitio a otro está escrito con http o está encriptado, éstas serían funciones de niveles superiores que, los routers, no comprueban.

4.4 Nivel 4 o de Transporte

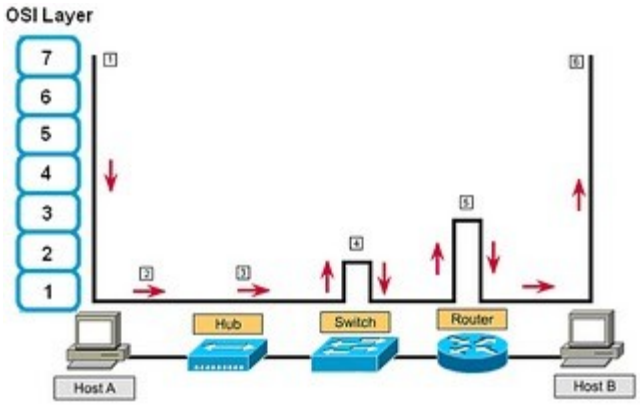
Trabaja con segmentos.
Funciones:

- Se encarga de la comunicación extremo a extremo (peer to peer).

Este nivel sólo está presente en los PCs, ni en switches ni en routers. A este nivel le es transparente el hecho de que, para alcanzar un destino, haya que atravesar 1 o 50 routers. Se comunica con el PC o equipo en el otro extremo directamente. La cabecera de N4 no la leen los equipos de comunicaciones, solo las estaciones finales.

En el siguiente esquema se ve la comunicación entre los hosts A y B que, como estaciones finales, implementan los 7 niveles OSI. La comunicación entre A y B atraviesa un hub (equipo que sólo implementa el nivel 1 de OSI), un switch o conmutador (implementa las capas 1 y 2) y un router (implementa las capas 1, 2 y 3). El hub recibe ceros y unos y los reenvía, le da igual lo que quieran decir o que haya un error. El switch recibe 0s y 1s pero también comprueba la cabecera de N2: comprueba, antes de reenviar la trama, que ésta no tenga errores mirando el CRC de la trama

recibida. El router comprueba las cabeceras de niveles 2 y 3: comprueba el CRC de N2, la dirección IP a nivel 3, etc.



4.5 Nivel 5: Sesión

Funciones:

- Resincronización. P.ej. Si estamos usando FTP para transferir un fichero entre dos extremos, en caso de pérdida de la conexión en mitad de la transferencia, no será necesario comenzar de nuevo por el principio. Existen unos puntos de resincronización o de control de manera que se vuelve al último de estos puntos y se continúa desde ahí.
- Manejo de testigo o token. El que tenga el testigo es el que puede transmitir. P.ej. El host que tenga el token es el que puede modificar una base de datos compartida entre varios hosts.

4.6 Nivel 6: Presentación.

Es el primer nivel al que le importa lo que quiera decir la información, el primero que se ocupa de la semántica. Para este nivel, el mensaje a transmitir tiene que tener sentido.
Se encarga de:

- compresión
- encriptación
- determinar el código a usar en el mensaje (p.ej: ASCII, EBCDIC...)

4.7 Nivel 7: Aplicación.

En este nivel se definen protocolos genéricos que puedan usar luego las aplicaciones de manera que cada aplicación no tenga que redefinir sus propios protocolos.
Protocolos abiertos que pueden usar las aplicaciones son:

- HTTP, Protocolo para la Transferencia de Hiper Texto (HyperText Transfer Protocol)lo usa el Internet Explorer, el Mozilla, el Chrome, etc para conectar un cliente web a un servidor web.

- SMTP, es un protocolo para la comunicación entre servidores y clientes de correo electrónico; lo usarán clientes de correo como Outlook, Eudora, Thinderbird, etc, para comunicarse con los servidores donde está nuestro correo y también lo usan los servidores de correo entre sí para intercambiarse los correos de sus usuarios.
- FTP, protocolo de transferencia de ficheros File Transfer Protocol. Lo usan clientes y servidores FTP como Filezilla o incluso desde la misma ventana de comandos de Windows (la ventana negra), probad a escribir ftp o ftp 192.168.8.2 en el IES). Con este protocolo podemos enviar o recibir ficheros.

4.8 Por qué OSI no triunfa

OSI surge una década más tarde que los protocolos TCP, IP, etc, cuando ya llevaban estos protocolos mucho tiempo funcionando sin problemas (¿para qué inventar algo nuevo?).

Los prototipos de fabricantes que eran compatibles con este modelo eran caros y malos. El hardware que usaba TCP/IP era más simple y barato, y funcionaba bien.

Respecto a OSI, se tiene la idea de que los gobiernos tratan de controlar las redes que estaban creciendo rápidamente sobre Unix (sistema abierto) con lo que el mercado tiene una razón más para rechazar los prototipos basados en OSI.

OSI se queda como una arquitectura teórica para enmarcar otros protocolos pero no se implementan actualmente equipos que sigan sus especificaciones y protocolos.

El modelo OSI se estudia actualmente para ver cómo se hace una arquitectura en capas y se utiliza para enmarcar los protocolos que se usan como IP, TCP, UDP, etc. Así, IP es un protocolo de nivel 3 porque lleva a cabo la mayor parte de las tareas que el modelo OSI puso en el nivel 3. Igual para Ethernet, que es un protocolo de N2 o TCP y UDP, que son de N4.

Por otro lado, como lo que se usa es la arquitectura TCP/IP últimamente aparecen en muchos libros los 3 niveles más altos de la torre OSI fusionados en uno sólo, llamado nivel de aplicación.

Finalmente, lo que se usa es una mezcla de los dos con lo que hay los siguientes niveles:

N1: físico

N2: de enlace (Ethernet, Wifi)

N3: de red (IP)

N4: transporte (TCP, UDP)

Nivel de aplicación (http, https, ftp, smtp, telnet…)