



1. Protocolos

1.1 Definición:

*Un protocolo es el conjunto de normas y convenciones requeridas para comunicarse dos o más **entidades** o **sistemas** que intercambian información.*

Entidades: programas de aplicación de usuarios, paquetes de transferencia de ficheros, sistemas de gestión de bases de datos, facilidades de e-mail, etc. (en definitiva, cualquier cosa con capacidad de enviar o recibir información).

Sistemas: computadoras, terminales, sensores remotos (en general, un objeto físicamente diferenciado, que contiene una o más entidades)

Los elementos que definen un protocolo son:

- ❖ Sintaxis: formato, codificación y niveles de señal de datos.
- ❖ Semántica: información de control y gestión de errores.
- ❖ Temporización: coordinación entre la velocidad y orden secuencial de las señales.

1.2 Tipos:

Las características distintivas más importantes de un protocolo son:

- **Directo/indirecto:** los enlaces punto a punto y las redes multipunto de difusión (p.ej. Ethernet) son directos. Si los sistemas se conectan a través de una o más redes conmutadas, no es posible el uso de un protocolo directo, ya que intervienen elementos intermedios.
- **Monolítico/estructurado:** monolítico es aquel protocolo que tiene el control en una sola capa, de todo el proceso de transferencia. En protocolos estructurados, hay varias capas jerárquicas que se coordinan y que dividen la tarea, constituyendo una *arquitectura de comunicaciones*.
- **Simétrico/asimétrico:** los simétricos son aquellos en que las dos entidades que se comunican son paritarias (las tareas y jerarquías de los equipos que se comunican es similar). Un protocolo es asimétrico si una de las entidades tiene mayor cantidad y jerarquía de funciones respecto de la otra (por ejemplo en clientes y servidores o también en modo de respuesta normal NRM en HDLC).
- **Normalizado/no normalizado:** los no normalizados son aquellos creados específicamente para un caso concreto y que no va a ser necesario conectarlos con sistemas externos. En la actualidad, para poder intercomunicar muchas entidades, es necesaria una normalización en los protocolos.

1.3 Funciones:

- a. **Segmentación y ensamblado:** generalmente es necesario dividir los bloques de datos en unidades pequeñas e iguales en tamaño, y este proceso se le llama segmentación. El bloque básico de segmento en una cierta capa de un protocolo se le llama **PDU** ("*Protocol Data Unit*"-Unidad de datos de protocolo).

La conveniencia de la utilización de bloques segmentados puede deberse a:

- La red sólo admite la transmisión de bloques de un cierto tamaño (p.ej. en ATM el bloque tiene un valor fijo de 53 octetos, mientras Ethernet tiene un tamaño máximo de 1526 octetos).
- El control de errores es más eficiente para bloques pequeños (p.ej. efectuando una repetición selectiva de una trama, es mejor si tiene menor cantidad de bits)
- Para evitar monopolización de una red multipunto por parte de un ETD, se emplean bloques pequeños y multiplexación entre las estaciones, resultando un menor retardo para cada una de las estaciones.
- Con bloques pequeños, las necesidades de almacenamiento temporal (buffer del receptor) pueden ser menores.

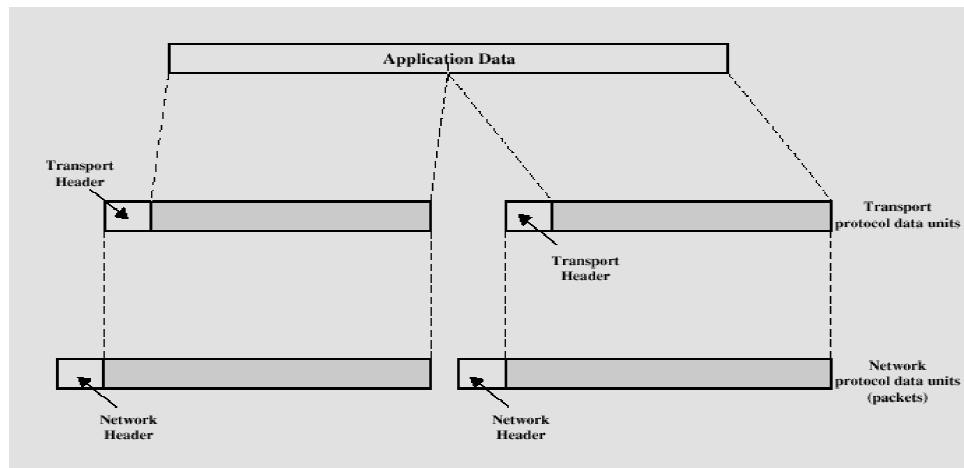


Ilustración 1: Segmentación de los datos

La inconveniencia de la utilización de segmentos pequeños puede deberse a:

- La información de control necesaria en cada bloque. Cuanto menor sea el bloque, mayor es el porcentaje de bits de control sobre los bits totales transmitidos (ineficiencia en la transmisión).
 - Los receptores pueden necesitar interrupciones para recibir cada bloque, con lo que en bloques pequeños habrá más interrupciones.
 - Mayor número de PDU, implican más tiempo de procesamiento.
- b. Encapsulado: se trata del proceso de anexar información de control al segmento de datos recibido de una capa superior. Esta información de control puede ser de direccionamiento entre emisor/receptor, códigos de detección de errores y control de protocolo.
 - c. Control de conexión: Interviene el concepto de transferencia *orientada a conexión* y la transferencia *no orientada a conexión*. Hay bloques de datos sólo de control y otros de datos y control. Cuando se utilizan datagramas, todos los bloques incluyen control y datos ya que cada PDU se trata independiente. En circuitos virtuales hay bloques de control que son los encargados de establecer la conexión del circuito virtual. Hay protocolos más sencillos y otros más complejos, por lo que los protocolos de los emisores y receptores deben de ser compatibles al menos. Además de la fase de establecimiento de conexión (en circuitos virtuales) está la fase de transferencia de datos y la de desconexión. Si se utilizan circuitos virtuales habrá que numerar las PDU y llevar un control en el emisor y en el receptor de los números de secuencias.
 - d. Entrega ordenada: el envío de PDU puede acarrear el problema de que si hay varios caminos posibles, lleguen al receptor PDU desordenados o repetidos, por lo que el receptor debe de tener un mecanismo para reordenar los PDU o descartar los duplicados. Hay sistemas que tienen un mecanismo de numeración con módulo n (algún número), que manejando el concepto de *ventana* hacen que sea imposible que haya dos segmentos en la red al mismo tiempo y con el mismo número.
 - e. Control de flujo: hay controles de flujo de parada y espera o de ventana deslizante. El control de flujo es necesario en variados protocolos y/o capas, ya que el problema de saturación del receptor se puede producir en cualquier capa del protocolo.
 - f. Control de errores: generalmente se utiliza un temporizador para retransmitir una trama una vez que no se ha recibido confirmación después de expirar el tiempo del temporizador. Cada capa de protocolo debe de tener su propio control de errores.
 - g. Direccionamiento: cada estación o dispositivo intermedio de almacenamiento debe tener una dirección única. A su vez, en cada terminal o sistema final puede haber varios agentes o programas que utilizan la red, por lo que cada uno de ellos tiene asociado un puerto.

Además de estas direcciones globales, cada estación o terminal de una subred debe de tener una dirección de subred (generalmente en el nivel MAC).

Hay ocasiones en las que se usa un identificador de conexión; esto se hace así cuando dos estaciones establecen un circuito virtual y a esa conexión la numeran (con un identificador de conexión conocido por ambas). La utilización de este identificador simplifica los mecanismos de envío de datos ya que por ejemplo es más sencillo que el direccionamiento global.

Algunas veces se hace necesario que un emisor emita hacia varias entidades a la vez y para eso se les asigna un direccionamiento similar a todas.

- h. Multiplexación: es posible multiplexar las conexiones de una capa hacia otra, es decir que de una única conexión de una capa superior, se pueden establecer varias conexiones en una capa inferior (y al revés).
- i. Servicios de transmisión: los servicios que puede prestar un protocolo son:
 - *Prioridad* : hay mensajes (los de control) que deben tener prioridad respecto a otros
 - *Grado de servicio*: hay datos que deben de retardarse y otros acelerarse (vídeo).
 - *Seguridad*.

1.4. Un modelo de tres capas

Se puede analizar un modelo simplificado para la transferencia de un archivo.

En la comunicación intervienen tres agentes: aplicaciones, computadoras y redes. Por lo tanto, es lógico organizar la tarea en tres capas.

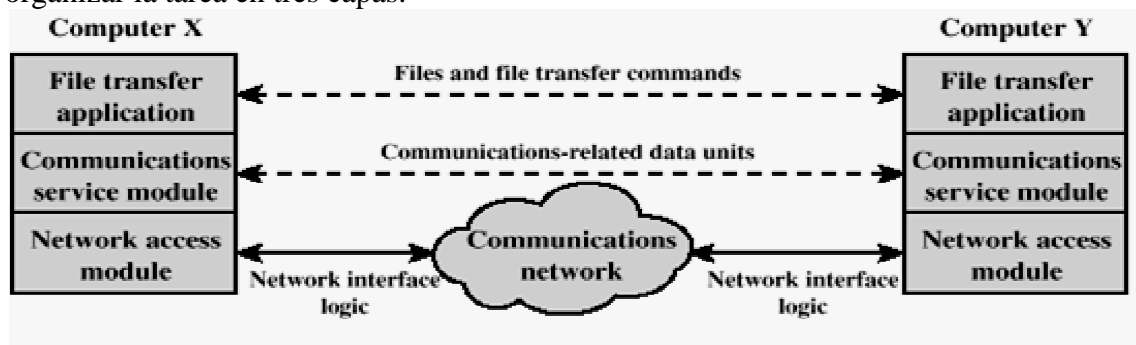


Ilustración 2: Arquitectura simplificada para la transferencia de un archivo

El primer módulo contiene toda la lógica que es exclusiva para la aplicación del ejemplo, o sea transferir el archivo (palabras clave, ordenes de archivo, registros, campos de registros, etc.).

Esta transmisión debe ser segura, pero esa seguridad también será necesaria para otras aplicaciones, como transmisión de documentos, correo electrónico, etc., por lo que no es práctico implementar las herramientas para conseguirlo en cada una de las distintas aplicaciones del módulo I.

Una mejor solución es implementar esas herramientas en el módulo II, para asegurar que los dos computadores estén activos y preparados para la transferencia de datos, efectuar los envíos y seguir su recorrido, independientemente de que tipo de aplicación sea.

Todas estas tareas en el módulo II deben ser independientes del tipo de red al que cada dispositivo final se conecte, de modo que si la red cambia, solo sea problema de ese módulo.

Por ello, ese tipo de tareas las efectúa el módulo encargado del acceso a la red.

El ejemplo simplificado tiene tres capas (Ilustración 3):

Capa de acceso a la red: trata del intercambio de datos entre el computador y la red a la que está conectado. El computador origen debe suministrar a la red la dirección del computador destino, de modo que los paquetes de datos se puedan encaminar correctamente. El software de esta capa dependerá del tipo de red que se use, por lo que contendrá protocolos para conmutación de circuitos, conmutación de paquetes, redes LAN, etc. El software de las capas superiores no se ocupará del tipo de red que se está utilizando.

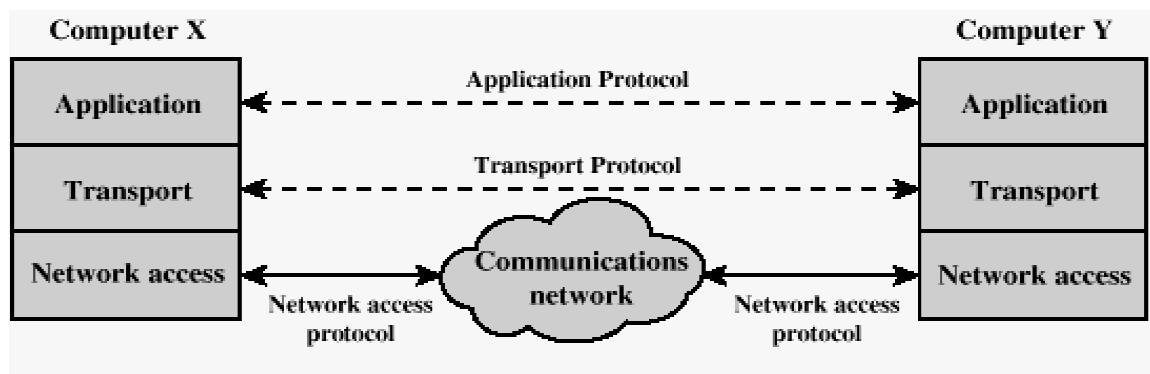


Ilustración 3: Protocolos en una arquitectura simplificada

Capa de transporte: contiene todo el software para un intercambio seguro de datos (que lleguen a destino y en el mismo orden), independientemente de la aplicación.

Capa de aplicación: contiene todo el software para admitir varias aplicaciones de usuario, independientes entre sí.

En la capa de aplicación, cada una de las mismas tendrá una dirección SAP (“*service access point*”- punto de acceso a servicio), que es en realidad una dirección en dos niveles:

- Dirección del computador en la red.
- Dirección de la aplicación dentro del computador.

Esto es así, ya que cada aplicación accede individualmente a la capa de transporte.

La PDU se encapsula luego con la cabecera de red, conteniendo la dirección de la red (“*network address*”) que la transportará hasta el próximo nodo de conmutación de la red de comunicaciones (“*communications network*”), y así sucesivamente hasta llegar a la capa de red del ETD destino.

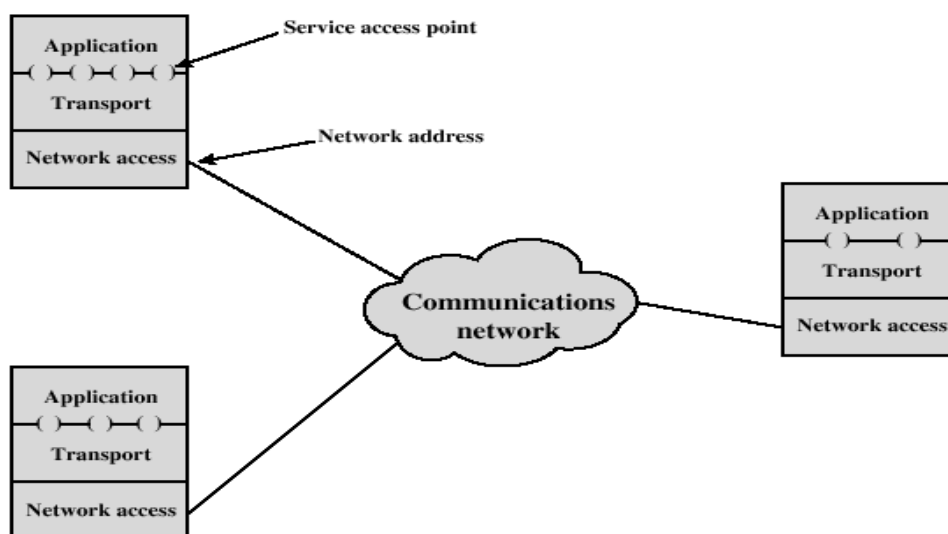


Ilustración 4: Interconexión mediante una red

Se simboliza con (1), (2),...los SAP en la capa de aplicación.

Se destaca que solo en la capa inferior existe una comunicación **física** entre protocolos gemelos, mientras que en las capas superiores, la comunicación entre protocolos paritarios es **lógica**.

El archivo a transferir está en la SAP1 de A, y se desea enviar a la SAP2 de B.

La comunicación entre capas paritarias se efectúa a través de unidades de información llamadas PDU (“*protocol data unit*”-unidades de datos de protocolo).

La cabecera de transporte es invisible para la PDU de red, o sea que a esa capa no le interesa el contenido de la PDU de transporte, ni accede a ella.

La cabecera que se agrega en cada capa, contiene información de control que será usada por el mismo protocolo, en la misma capa, en el destino.

A esta operación se la llama “encapsulado” de los datos.

La cabecera de transporte contiene:

- SAP destino: la capa de transporte destino sabrá a quien van los datos.
- Numero de secuencia: las PDU de transporte pueden llegar a destino desordenadas.
- Código de detección de errores: CRC sobre el contenido del resto de la PDU. El destino repite el cálculo para verificar.

La cabecera de red contiene:

- Dirección del computador destino dentro de la red (no del puerto).
- Petición de facilidades (p.ej., prioridades)

La red acepta las PDU de red, y las transmite a B como paquetes de datos. En el destino, cada PDU recibida es examinada en su cabecera (solamente), y si todo es correcto, esa cabecera es descartada y el resto de la PDU pasado al nivel superior. En cada capa se repite la operación.

El protocolo debe definir las reglas, convenios, funciones utilizadas, etc., para la comunicación por medio de red.

Cada capa del protocolo le pasa datos a la siguiente capa y ésta le añade datos propios de control y luego pasa el conjunto a la siguiente capa. Por tanto, cada capa forma unidades de datos que contienen los datos tomados de la capa anterior junto a datos propios de esta capa, y al conjunto obtenido se le llama **PDU** (unidad de datos del protocolo).

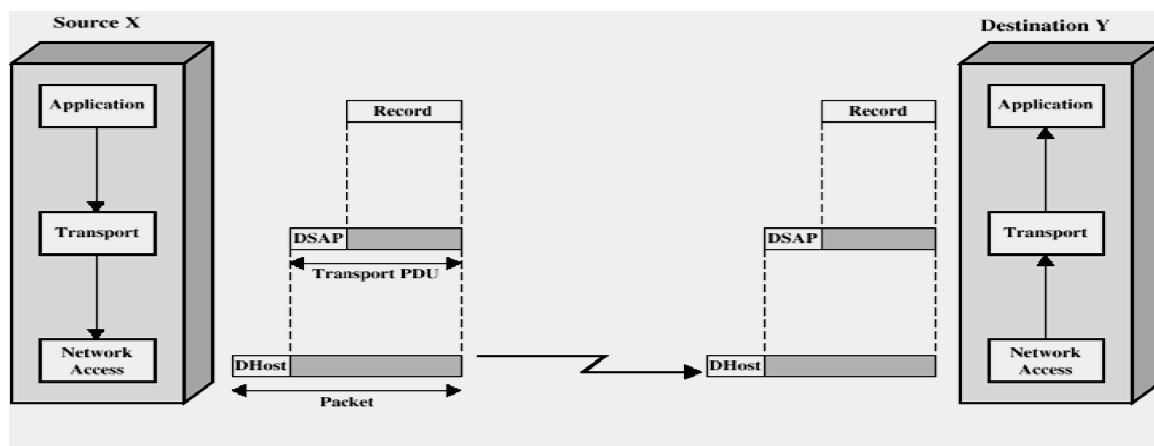


Ilustración 5: Operación de una arquitectura de protocolos.

2.OSI (“Open System Interconnection”-Interconexión de Sistemas Abiertos)

La ISO (“International Standarization Organization”) formó el comité para desarrollar la arquitectura en el año 1977, aunque el estándar fue publicado como ISO 7498 recién en 1984.

2.1. El modelo

El sistema de comunicaciones del modelo OSI estructura el proceso en siete capas que interaccionan entre sí. Una capa proporciona servicios a la capa superior siguiente y toma los servicios que le presta la siguiente capa inferior.

Algunas consideraciones importantes a tener en cuenta:

- Entre los dispositivos fuente y destino, se establece una “conexión lógica” entre las capas paritarias, que deben tener en cada una un protocolo común para poder comunicarse. La única “conexión física” se da a nivel de la capa física, que es la que transmite realmente los bits a través del medio.

- Una capa debería estar definida de modo tal que los cambios en ella, no requieran cambios en las otras. (Por ejemplo, modificación de protocolos por avances en arquitecturas, software, hardware, etc.)
- Cada capa agrupa lógicamente funciones afines.
- Cada capa poseerá solamente fronteras con su capa superior e inferior, de modo de poder definir, de ser necesario, la interfaz normalizada que se requiera entre ellas.

Los objetivos planteados por la I.S.O al desarrollar el modelo O.S.I han sido los siguientes:

- a) Estandarizar la comunicación entre sistemas.
- b) Eliminar los impedimentos técnicos para la comunicación entre sistemas.
- c) Eliminar la necesidad de describir las operaciones internas de los sistemas.
- d) Definir los puntos de interconexión para el intercambio de información entre los sistemas.
- e) Acotar la variedad de opciones, para aumentar la capacidad de los sistemas, evitando costosas conversiones y transformaciones entre productos dispares (“compatibilidad”).
- f) Proporcionar un punto de partida razonable en el caso de que los estándares no cubran todas las necesidades, y se necesiten nuevos desarrollos.

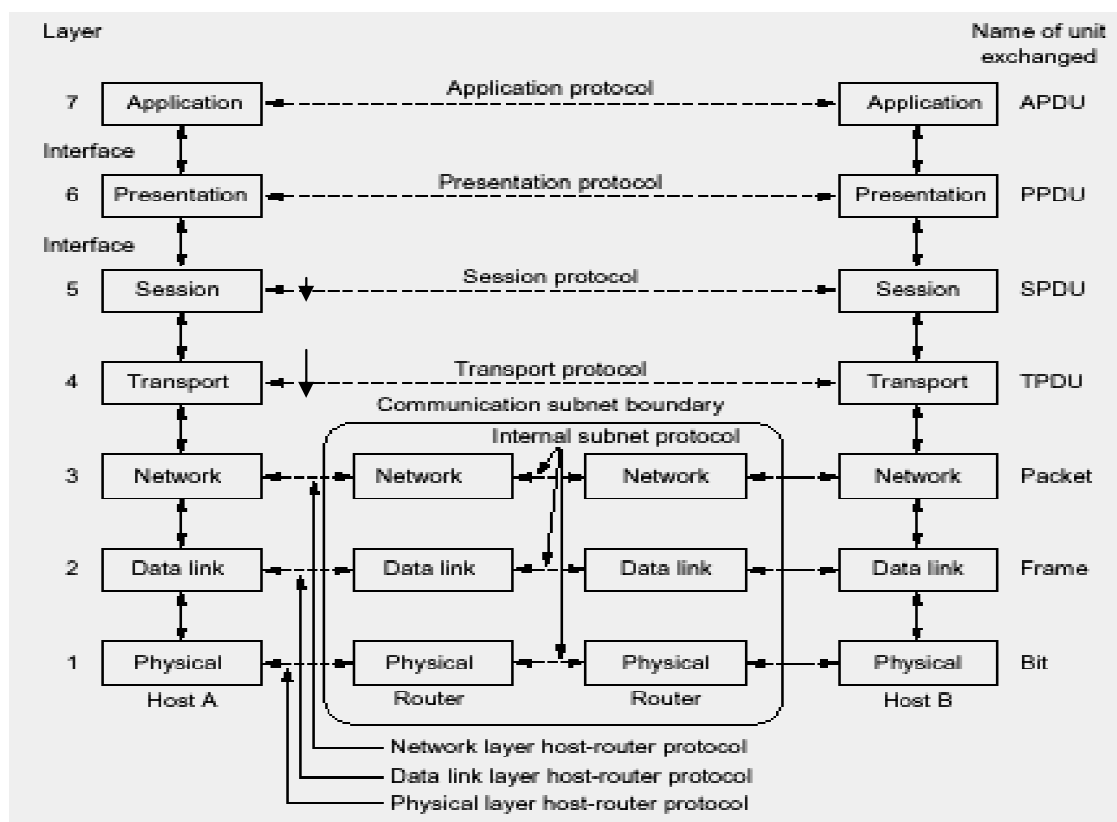


Ilustración 6: Capas de la arquitectura de protocolos OSI (entre dos Host y un Router intermedio).

De esta manera, el problema se divide en sub-problemas más pequeños y por tanto más manejables. Para comunicarse dos sistemas, ambos tienen el mismo modelo de capas.

La capa más alta del sistema emisor se comunica con la capa más alta del sistema receptor, pero esta comunicación se realiza vía capas inferiores de cada sistema. La única *comunicación física* real entre capas de ambos sistemas se produce en la capa física. La comunicación entre capas equiparitarias por encima de la física, es una *comunicación lógica*.

Los datos parten del emisor y cada capa le adjunta datos de control hasta que llegan a la capa física. En esta capa son pasados a la red y recibidos por la capa física del receptor. Luego irán siendo desencapsuladas las PDU, separando los datos de control de cada capa y pasados a una capa superior el segmento resultante. Al final, los datos de usuario llegan a la capa superior.

Cada capa tiene la facultad de poder fragmentar los datos que le llegan en trozos más pequeños para su mejor manejo. Luego serán reensamblados en la capa paritaria de la estación de destino previamente a su paso a la capa superior en el receptor.

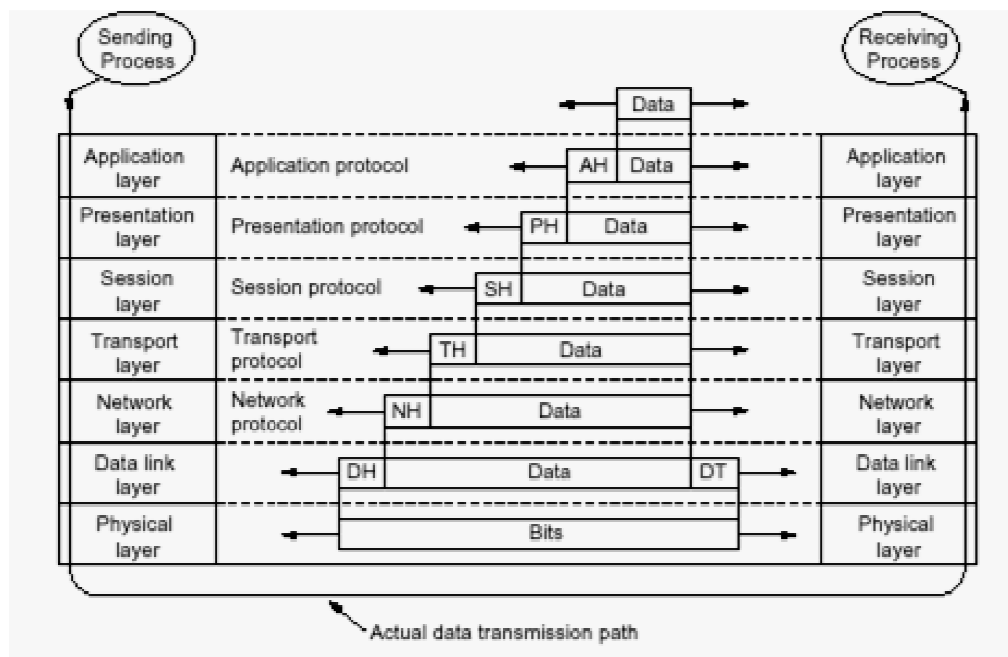


Ilustración 7: Proceso de transmisión en el modelo OSI.

2.2. Normalización dentro del modelo OSI

- El proceso de descomposición del problema de comunicaciones en capas, hace posible la normalización de cada capa en forma independiente y la posible modificación de una capa sin afectar a las demás.
- Funciones similares deberían agruparse en la misma capa.
- En cada capa se pueden desarrollar uno o más protocolos.
- Es preciso el empleo de normalizaciones para que dos sistemas puedan conocerse y poder comunicarse con plena exactitud, sin ambigüedades.
- Al estar (en teoría) perfectamente definidas las interfaces entre las capas, el reemplazo o modificación en el software de una capa no debe afectar a las otras capas.
- Dentro de cada capa se suministra:
 - El servicio proporcionado a la capa superior siguiente.
 - El SAP o sea el direccionamiento de acceso a ese servicio
 - El protocolo requerido, a la capa paritaria en el otro sistema.

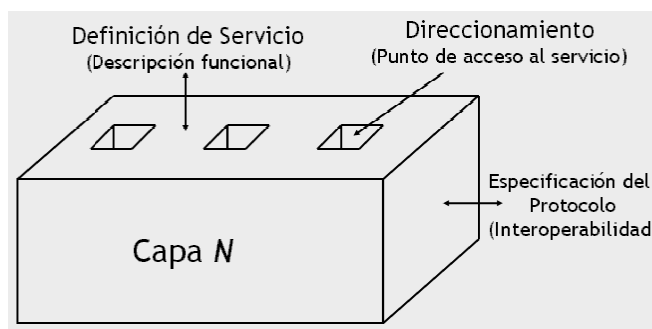


Ilustración 8: Concepción de protocolos en capas.



2.3. Primitivas de servicio y parámetros

Las capas inferiores (N-1) suministran a las superiores (N+1) una serie de funciones o primitivas y una serie de parámetros. La implementación concreta de estas funciones está oculta para la capa superior, la cual sólo puede utilizar las funciones y los parámetros para comunicarse con la capa inferior (paso de datos y control).

2.4. Las capas de OSI

- a. **CAPA FÍSICA:** Abarca la interfaz física entre los dispositivos y las reglas por las cuales se pasan los bits de uno a otro.

Tiene cuatro características importantes:

- *Mecánicas:* relaciona las propiedades físicas de la interfaz con el medio de transmisión. Normalmente incluye la especificación de un conector que une a una o más señales del conductor, llamados circuitos de intercambio.
- *Eléctricas:* relaciona la representación de los bits (por ejemplo, en términos de niveles de tensión) y la tasa de transmisión de datos.
- *Funcionales:* Especifica las funciones realizadas por los circuitos de intercambio de la interfaz física, entre un sistema y el medio de transmisión de datos.
- *De Procedimiento:* especifica la secuencia de eventos por los que se intercambia un flujo de bits a través del medio físico.

Son ejemplos de protocolos de la capa física el EIA RS-232 (V.24), y algunas partes de los estándares RDSI y de LAN.

- b. **CAPA DE ENLACE DE DATOS:** Mientras la capa física proporciona solamente un servicio bruto de flujo de datos, la capa de enlace de datos intenta hacer el enlace físico seguro, proporcionando los medios para activar, mantener y desactivar el enlace.

Su principal servicio a las capas superiores es la detección de errores y control de flujo. Así, si dos sistemas están enlazados por una conexión “punto a punto” (conexión directa, sin una red), las capas superiores a la de enlace de datos podrán suponer una conexión libre de errores.

Son ejemplos de protocolos de capa de enlace de datos: HDLC, SDLC, LAPB, LAPD, LLC, etc.

Si la conexión no es directa, sino a través de algún tipo de red, las capas superiores no quedan liberadas de los controles, ya que habrá múltiples enlaces simultáneos e independientes con dispositivos intermedios.

- c. **CAPA DE RED:** Proporciona los medios para la transferencia de información entre sistemas finales, a través de algún tipo de red de comunicaciones. Libera a las capas superiores de la necesidad de tener conocimiento sobre la transmisión de los datos y de las tecnologías de conmutación utilizadas. En esta capa, el dispositivo emisor está envuelto en un diálogo con la red, para especificar la dirección destino y solicitar ciertas facilidades. La capa de red se ocupa de la gestión de comunicaciones intermedias, pudiendo tener ellas muchas variantes, por ejemplo

- Enlace directo punto a punto entre las estaciones: como se mencionó, no sería necesaria la capa de red, ya que la gestión del enlace la puede manejar la capa 2.
- Enlace entre dos sistemas pertenecientes a la misma red: los paquetes creados por un sistema final, pasan a través de uno o más nodos de la red, que actúan como retransmisores entre los sistemas finales (Ilustración 9. Cada nodo de la red implementa las capas 1 a 3 del modelo. La capa 3 del nodo realiza funciones de conmutación y de encaminamiento. Dentro del nodo existen dos capas de enlace de datos y dos capas físicas, que actúan independientemente, vinculando sus respectivos extremos con la capa superior de red del nodo. Las cuatro capas superiores son protocolos “extremo-extremo” entre los sistemas finales conectados, independientemente de los nodos intermedios. (Ej: protocolo X-25 de paquetes)

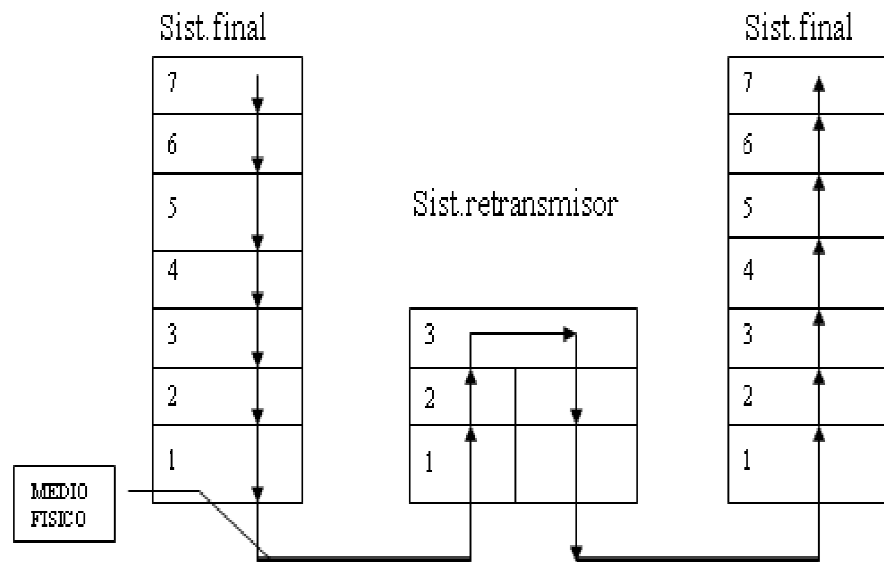


Ilustración 9: Concepción de protocolos en capas.

- d. **CAPA DE TRANSPORTE:** esta capa permite que los datos se entreguen libres de errores, en secuencia, sin pérdidas o duplicados. Esta capa también puede brindar a su capa superior servicios de red adicionales: tasas de error aceptables, retardo máximo, prioridad, seguridad, etc. El tamaño y complejidad del protocolo de transporte dependerá de lo seguras o inseguras que sean las redes integrantes, y de los servicios de red requeridos. I.S.O. ha desarrollado una familia de cinco estándares de protocolos de transporte, orientados a los diferentes servicios disponibles.
- e. **CAPA DE SESION:** proporciona los mecanismos para controlar el diálogo entre aplicaciones en sistemas finales. Las cuatro capas inferiores de O.S.I. brindan un medio para el intercambio rápido y seguro de datos, aunque en muchos casos estos servicios son insuficientes, por ejemplo en:
 - Una aplicación de acceso a un terminal remoto, podría requerir un diálogo semiduplex.
 - Una aplicación de procesamiento de transacciones, podría requerir confirmaciones, recuperaciones y copias de seguridad.
 - Una aplicación de procesamiento de mensajes, podría requerir interrumpir el proceso para efectuar otra tarea, y luego retomarlo donde se dejó.

Todas estas capacidades se podrían implementar dentro de cada aplicación específica, a nivel de la capa 7, pero siendo herramientas ampliamente utilizadas para estructurar el diálogo, es lógico organizarlas en una capa separada, la capa de sesión. Esta capa tendrá mayor o menor utilidad según la aplicación en uso. Sus servicios fundamentales son:

 - Disciplina del diálogo: full-dúplex o semi-duplex.
 - Agrupamiento: en el flujo de datos transmitidos, se pueden establecer marcas para separar los datos por grupos.
 - Recuperación: la capa de sesión tiene un mecanismo de “puntos de comprobación”, a partir de los cuales retransmite los datos ante un eventual fallo.
- f. **CAPA DE PRESENTACION:** proporciona la sintaxis (formato) de los datos a utilizar entre las aplicaciones. NO se ocupa del significado de los datos, sino que su función es aceptar tipos de datos del nivel de aplicación y negociar con su nivel homólogo en el otro extremo, la sintaxis de representación (p.ej. ASCII, Teletipo, Videotex, etc.). O sea que básicamente éste nivel cuenta con muchas tablas de sintaxis, disponiendo de funciones muy limitadas (las principales: compresión y encriptado de datos).
- g. **CAPA DE APLICACION:** proporciona un medio a los programas de aplicación para que accedan al entorno OSI. A diferencia del nivel de presentación, éste nivel SI tiene en cuenta el significado de los datos. Contiene funciones de administración y financieras, en varios lenguajes de

programación, como también mecanismos para aplicaciones distribuidas. Residen en este nivel aplicaciones de uso general, como transferencia de archivos, correo electrónico, acceso a terminales remotas, etc.

3. Arquitectura de protocolos TCP / IP

(TCP/IP: “Transmission control protocol/Internet protocol- Protocolo de control de transmisión/Protocolo Internet”)

3.1. INTRODUCCIÓN:

Hasta mediados de los años 70, las redes no estaban diseñadas como para compartir recursos. O bien los administradores eran reacios a compartir, alegando problemas de seguridad, o bien las redes eran directamente incompatibles.

Al revertirse esa tendencia, surgió la necesidad de acordar un conjunto de tecnologías y normas comunes para que las redes puedan comunicarse.

Comenzaron a desarrollarse los protocolos y servicios en capas, modelando lentamente la arquitectura O.S.I., que en los años 80 se daba por descontado que sería la estructura triunfante a nivel comercial sobre otras competidoras como S.N.A. desarrollada por IBM, o TCP/IP desarrollada por el Departamento de Defensa de USA.

En los años 90, se vio que no fue así, triunfando TCP/IP en forma contundente, entre otras muchas razones por:

- TCP/IP especificó y puso en uso sus protocolos con gran anterioridad a que se finalizaran las normas ISO. Las empresas e instituciones oficiales tenían urgencias y no podían esperar, por lo que utilizaron las alternativas disponibles. Al salir OSI, el costo y riesgo de cambiar de plataforma, frenó el despegue comercial de OSI. La aparición de los estándares de cualquier tecnología, necesariamente debe producirse luego de una considerable investigación que garantice los resultados buscados, pero antes de que la industria efectúe inversiones tan sustanciales que se fueren estándares de facto.

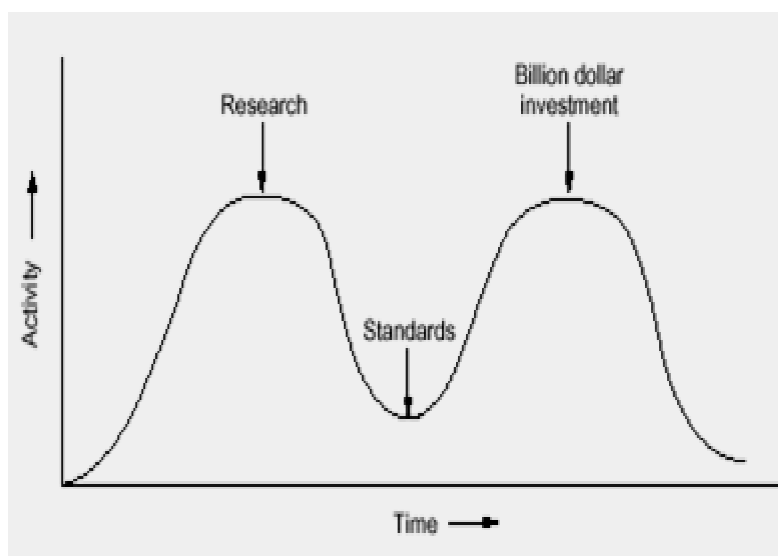


Ilustración 10: El apocalipsis de los dos elefantes.

- Los protocolos iniciales de TCP/IP fueron desarrollos del Departamento de Defensa (que es el principal consumidor de software del mundo), que al no poder esperar a OSI, compra productos alternativos, creando un mercado para ellos, e impulsando su crecimiento explosivo.
- Internet, la red de redes, se construyó sobre TCP/IP. El crecimiento de Internet y la w.w.w. consolidó el triunfo.

3.2. COMPARACION ENTRE OSI Y TCP/IP:

Parten de las mismas premisas:

- La tarea de comunicación es tan compleja y diversa que no se puede realizar en una única unidad.

- Separación en módulos jerárquicos que se pueden comunicar con sus paritarios en otra unidad.

Las diferencias conceptuales más importantes entre ambos se podrían resumir en:

- Concepto de capa:** OSI agrega un paso que TCP/IP no posee, consistente en que protocolos de características comunes se deben agrupar en un mismo nivel (éste concepto otorga a OSI buena utilidad didáctica). En cambio, TCP/IP entiende que indicar que los protocolos dentro de una cierta capa realizan cierta tarea, no resulta siempre deseable o posible, por lo que define protocolos dentro de un mismo nivel, aun realizando estas funciones distintas. Su premisa es que los protocolos de una capa, serán todos aquellos que compartan los servicios de soporte de los protocolos de una capa adyacente inferior común.
- Uso de las capas:** OSI implica que las interfaces entre capas están bien definidas, por lo que en una capa se puede reemplazar un protocolo por otro sin afectar las capas adyacentes. (Para conseguir respetar su propio modelo, OSI recurre a conceptos dudosos como la definición de capas o subcapas “nulas”) TCP/IP entiende que esto no es siempre deseable o posible, por lo que no establece como obligatorio el uso de todas las capas.
- Modelos:** mientras OSI tiene un modelo establecido, TCP/IP no lo tiene, aunque es usual estudiarlo como una arquitectura de cinco capas, relativamente independientes entre sí, para poder establecer la siguiente analogía con el modelo OSI:

APLICACION	APLICACION
	PRESENTACION
TRANSPORTE	SESION
	TRANSPORTE
INTERNET	RED
ACC. A LA RED	ENL.DE DATOS
FISICA	FISICA

TCP/IP

OSI

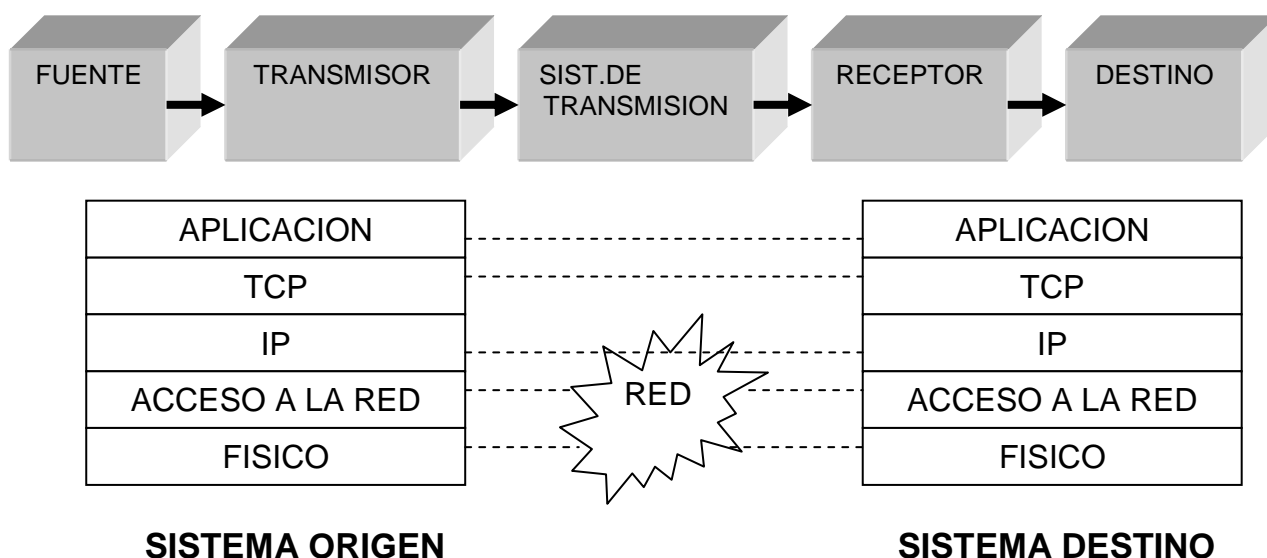


Ilustración 11: Modelos OSI y TCP/IP comparados.



Se puede entonces agregar, a partir de éste “modelo”, para las capas de TCP/IP:

a) CAPA FÍSICA:

Interfaz física entre el dispositivo transmisor de datos y el medio de transmisión o red. Especifica las características del medio de transmisión, naturaleza de las señales, velocidad de los datos, etc.

b) CAPA DE ACCESO A LA RED:

Responsable del intercambio de datos entre el sistema final y la red.

El emisor comunica a esta capa la dirección destino, para que la red pueda encaminar los datos, brindando de ser necesario, ciertos servicios (p.ej. prioridades). Hay protocolos para distintos tipos de redes disponibles: conmutación de circuitos, conmutación de paquetes (p.ej. X-25), redes LAN (p.ej. Ethernet), etc. Esta capa, entonces, contiene las funciones relacionadas con el acceso y encaminamiento a través de la red de acceso del usuario origen, de modo que las capas superiores se desentiendan del tipo de red que éste use.

c) CAPA INTERNET:

En general, los dispositivos están conectados a redes distintas, por lo que serán necesarios procedimientos y normas (Protocolo Internet) para que los datos atraviesen las distintas redes interconectadas. (No confundir con Internet, la red de redes).

El IP se implementa tanto en los dispositivos finales como en los intermedios (“Router”: dispositivo intermedio de red, con capacidad de procesamiento y encaminamiento, que retransmite datos entre un tipo de red y otro).

Las capas superiores se desentienden de las distintas redes a enlazar para el flujo de la información.

d) CAPA DE TRANSPORTE:

Se agrupa en ésta capa a los procedimientos que brindan seguridad en el intercambio de datos, independientemente de la naturaleza de la aplicación de la capa superior.

Asegura que todos los datos llegan a destino, sin duplicar, en orden y libres de errores.

El protocolo más usado es el TCP, que se implementa sólo en los dispositivos origen y destino, no en los intermedios.

e) CAPA DE APLICACIÓN:

Contiene todo el software necesario para las aplicaciones de usuario, dividido en módulos dentro de la capa (p.ej. para transferir un archivo)

Nota: observar en la Ilustración 11, de acuerdo a lo comentado en las definiciones de las capas:

- Capas 1 y 2: hay interacción entre sistemas finales y la red.
- Capas 4 y 5: hay interacción solo entre sistemas finales (son protocolos extremo a extremo).
- Capa 3: es híbrida, ya que hay relación entre los sistemas finales y la red para el encaminamiento, pero también hay funciones adicionales de intercambio entre los dos sistemas finales.

3.3. FUNCIONAMIENTO DE TCP e IP:

La red está formada por múltiples sub-redes, vinculadas por dispositivos de encaminamiento (“routers”). IP está implementado en las computadoras y “routers”, retransmitiendo bloques de datos a través de ellos.

TCP está implementado solo en los sistemas finales, guardando el registro de los bloques de datos, para asegurar la entrega segura a la aplicación adecuada.

El direccionamiento, como en OSI, requiere dos niveles:

- Dirección internet global: identifica al computador destino entre todos los conectados a la red.
- Puerto: identifica la dirección de la aplicación dentro del computador destino (equivalente al concepto de SAP “Punto de acceso de servicio” en el modelo OSI).



Sea un ejemplo en el cual, la aplicación X en el puerto 1 de A quiere enviar un mensaje a la aplicación X en el puerto 2 de B. (Ilustración 12)

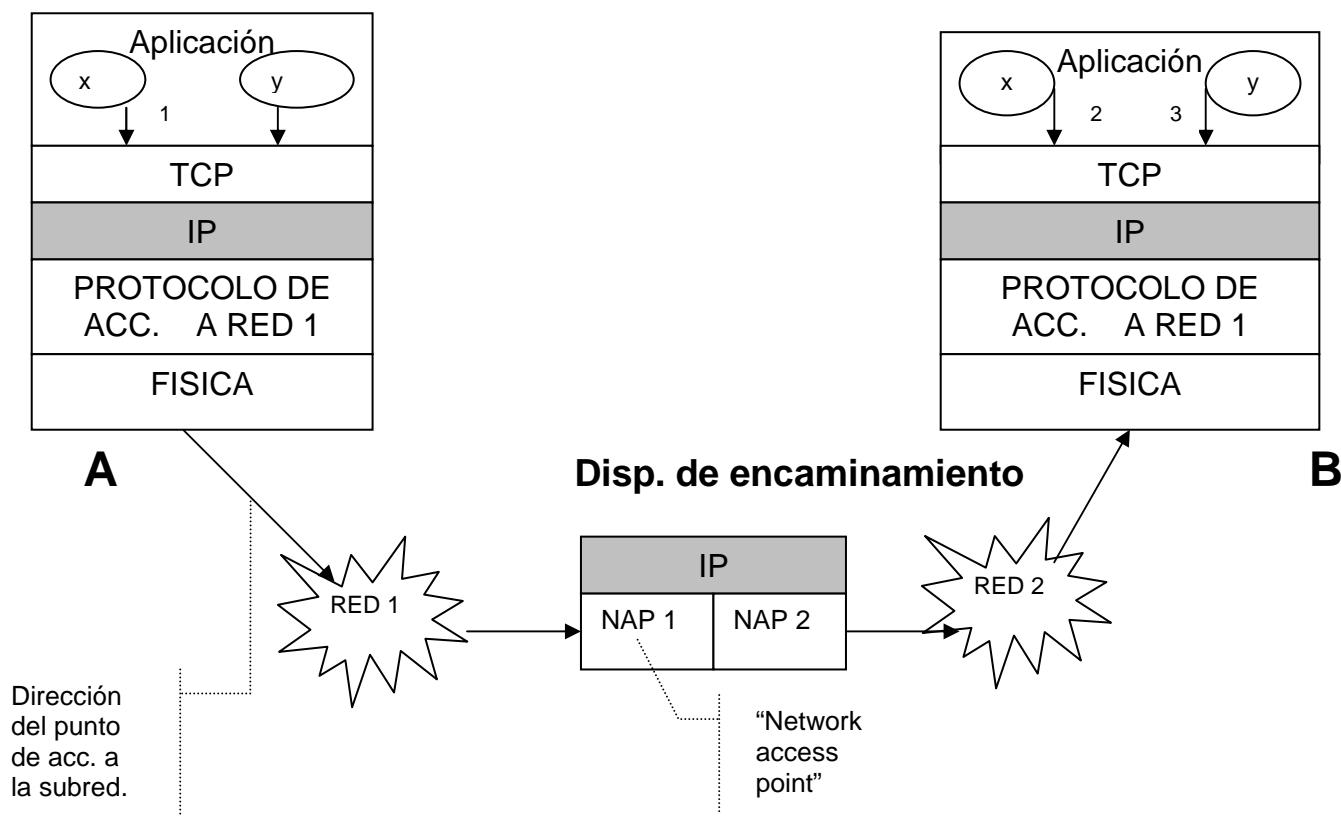


Ilustración 12: Modelos OSI y TCP/IP comparados.

La aplicación de A ordena a TCP enviar los datos a B, puerto 2.

TCP ordena a IP enviar a B (no especifica el puerto, ya que a la capa IP de destino no le incumbe).

IP ordena a la capa de acceso a la red (p.ej. podría ser a una Ethernet) enviar a través de la sub-red 1 hacia el primer router, para que retransmita hacia el próximo, con rumbo al destino (en el gráfico por simplicidad se puso un solo router).

Las PDU que intervienen, tienen el siguiente formato:

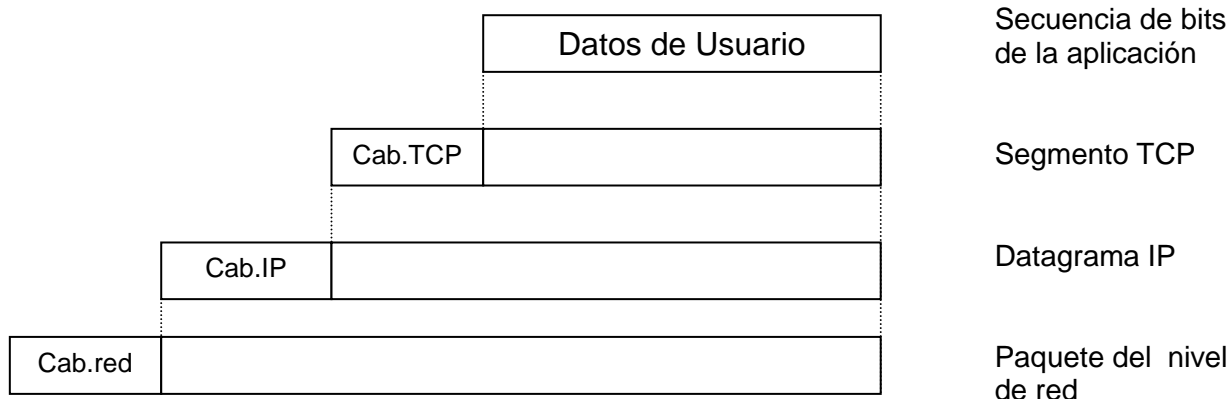


Ilustración 13: Modelos OSI y TCP/IP comparados.

TCP puede fragmentar los datos de usuario.

La información de control que agrega en su cabecera será usada por TCP paritaria en destino (puerto destino, número de secuencia y CRC)

IP en su cabecera incluye información de control (p.ej. dirección del computador destino).

Cada datagrama IP se presenta a la capa de acceso a la red, para ser transmitido.

La última capa agrega una cabecera, con entre otros campos, dirección de la sub-red destino y solicitud de facilidades.

Se tiene entonces un paquete que se transmite por la primera sub-red hacia el primer router. El dispositivo de encaminamiento recibe el paquete, elimina la primera cabecera (la de red), y examina la cabecera IP. Como ésta contiene la dirección destino, el módulo IP del router, luego de consultar sus tablas de encaminamiento, re-direcciona ese datagrama como un nuevo paquete (con una nueva cabecera de red) a través de la sub-red 2 hacia B. Al llegar al dispositivo final, o sea B, cada capa analiza su cabecera paritaria, efectúa los controles, que de ser correctos, dan como resultado que esa cabecera sea eliminada y el resto de la información a la capa superior, hasta llegar a la aplicación de destino. Se remarca que TCP/IP no requiere obligatoriamente el uso de todas las capas, ya que hay aplicaciones que pueden invocar directamente los servicios de la capa que requieren. (p.ej. si no hay interconexión de redes, no necesita TCP y solicitan directamente a la capa de acceso de red).

3.4. PRINCIPALES PROTOCOLOS TCP/IP:

Surgidos como desarrollos militares del Departamento de Defensa de EEUU para aplicaciones sobre su red de comunicaciones ARPANET, constituyen la base de la actual Internet.

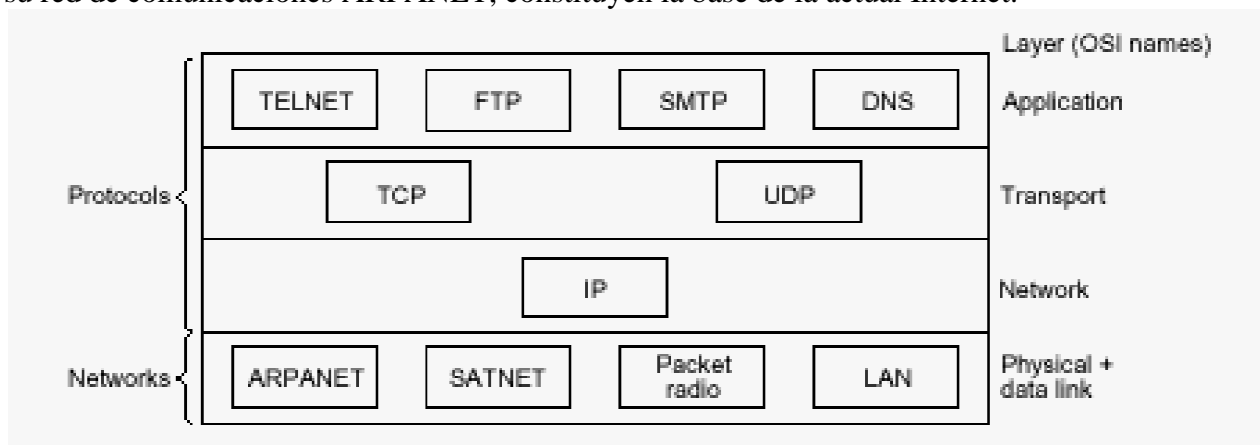


Ilustración 14: Principales protocolos de la suite TCP/IP

- SMTP ("Simple Mail Transfer Protocol"-Protocolo de transferencia simple de correo). Correo electrónico básico, para transferir mensajes entre computadores independientes, incluyendo lista de mensajes, acuse de recibo y reenvío. El protocolo SMTP no especifica cómo se crea el mensaje, por lo que se requiere un programa de correo electrónico o un editor de texto local. SMTP acepta mensajes creados, y mediante TCP lo transfiere a otro módulo SMTP en otro computador.
- FTP ("File Transfer Protocol"-Protocolo de transferencia de archivos), transfiere archivos de un sistema a otro bajo órdenes del usuario, tanto de textos como de datos binarios. A la solicitud de transferir por parte de un usuario, TCP establece la conexión con el destino, solicitando usuario y contraseña, permitiendo identificar archivo y acciones a realizar antes de habilitar la transferencia del material.
- TELNET, que da la capacidad de conexión desde una terminal o PC con un computador remoto, y trabajar en él como si el usuario estuviera en ese sitio. El tráfico también se canaliza por TCP.
- HTTP ("Hyper Text Transfer Protocol"-Protocolo de transferencia de hipertexto). Designa recursos de Internet que son la base de la WWW ("World Wide Web"), pudiendo utilizarse en cualquier aplicación cliente-servidor que suponga la aplicación de hipertextos. Los datos transferidos por el protocolo pueden ser texto propiamente dicho, hipertexto, imágenes, audio, video y en general, cualquier información accesible a través de Internet.



Algunos consideran poco adecuado el nombre de HTTP, ya que en realidad no se trata de un protocolo para “transferir “hipertextos, sino de un protocolo para transmitir información con tal eficiencia, que hace que los datos “salten” ante el usuario en forma casi instantánea.

- e) DNS (“Domain Name System”- Sistema de Nombres de Dominio). En general, los programas en Internet usan cadenas de caracteres ASCII en lugar de las direcciones binarias de los hosts, buzones de correo y demás recursos. Se debe tener un mecanismo que permita convertir las cadenas ASCII en direcciones IP binarias y viceversa., lo que se consigue con un esquema de nombres jerárquico basado en dominio y una base de datos distribuida para implementar éste esquema de nombres. Para relacionar un nombre con una dirección IP, un programa de aplicación llama a un procedimiento de biblioteca llamado *resolvidor*, pasándole el nombre como parámetro. El resolvidor envía un paquete UDP a un servidor DNS local, que entonces busca el nombre y devuelve la dirección IP al resolvidor, que entonces lo devuelve al solicitante. Con la dirección IP, el programa puede entonces establecer una conexión TCP con el destino, o enviarle paquetes UDP.