

1. Capítulo 2: Arquitectura de Protocolos

El intercambio de datos entre sistemas de computación implica procedimientos que, por lo general, son muy complejos.

Imagine que un dispositivo con un sistema operativo, placa de red y conectores definidos, desea enviar datos en algún formato determinado a otro dispositivo remoto, que podría estar en cualquier lugar del planeta, o inclusive podría residir en el espacio. De hecho, el dispositivo origen desea que los datos enviados al destino, no sólo puedan llegar sin errores, sino que también puedan ser interpretados por éste último (el cual probablemente cuente con un sistema operativo, placa de red y conectores).

Sin lugar a dudas hay muchas tareas por hacer: preparar los datos, encontrar un camino, avisarle al sistema destino que se quiere enviar información, acordar velocidad de entrega, coordinar con la aplicación destino, controlar la secuencia, detectar errores, entre otras.

Está claro que debe haber un alto grado de cooperación entre los dos sistemas informáticos. En lugar de implementar la lógica de la solución en un solo módulo, la tarea se divide en subtareas, cada una de las cuales se desarrolla por separado. Cada una de las tareas se ejecuta en una capa distinta, y se respeta un determinado conjunto de reglas.

2.1. Protocolo y Arquitectura de Protocolos

Para iniciar el estudio en detalle de las acciones que tienen lugar en el intercambio de datos entre sistemas de procesamiento de datos, es necesario definir elementos y conceptos involucrados en la comunicación que serán utilizados en este capítulo. Estos son:

- Dispositivos de usuario. Son los dispositivos o sistemas finales que están conectados a una red de comunicación. En este contexto, un dispositivo de usuario de hasta hace poco tiempo era muy generalizado es el computador, por ello en este capítulo se usarán indistintamente ambas denominaciones. Sin embargo, debe quedar claro que en la actualidad el Smartphone o teléfono inteligente ha pasado a ser un elemento clave para conectarse a las redes. También vale la pena aclarar el avance del uso de tabletas, notebooks, netbooks, e inclusive dispositivos sensores en artefactos electrodomésticos, vehículos, entre otros; que tienen la capacidad de transmitir y recibir datos.
- Comunicación entre dispositivos de usuario. Se denomina así al intercambio de información entre dispositivos de usuario. Por lo expuesto en el párrafo anterior, también se referirá a esta comunicación como “comunicación entre computadores”.

- Red de computadores. Se designa así al conjunto de dispositivos de usuario que intercambian datos a través de una red de comunicación, cooperando con la misma en la comunicación. La gran mayoría de las redes de computadores se encuadran dentro de las categorías de LAN, MAN o WAN. No obstante, es necesario remarcar que, sin importar la categoría, una red de computadores comprende el conjunto de dispositivos de usuario más la red de comunicación a la que están conectados.
- Entidad. Es cualquier programa ejecutándose en un dispositivo de usuario que tiene capacidad de intercambiar información con otra entidad. Obsérvese que, de acuerdo a esta definición, no cualquier programa de usuario ejecutándose configura una entidad; es necesario, además, que tenga capacidad para generar y/o consumir información en forma de datos. Ejemplos de entidades son: un programa de correo electrónico, un utilitario para la transferencia de archivos, un motor de bases de datos que envía registros a una aplicación en otro computador, un programa corriendo en un equipo medidor de temperatura que envía datos a un computador.

Obsérvese que, acuerdo a las definiciones precedentes, un dispositivo puede contener varias entidades. Este sería el caso, por ejemplo, de un computador en el que están ejecutándose varias aplicaciones simultáneamente. Sin embargo, pueden existir dispositivos con capacidad para contener una única entidad como, por ejemplo, un terminal no inteligente o un equipo de medición de una magnitud física específica que transfiere a un computador los datos resultantes de la medición.

Para que dos entidades en sistemas diferentes se puedan comunicar deben poder hablar el mismo idioma. Esto les permite saber qué, cómo y cuándo se comunican. Llevar a cabo cada una de estas tareas implica, necesariamente, seguir una serie de convenciones que deben ser aceptadas por las entidades involucradas. Este conjunto de convenciones o convenios se denomina “*protocolo*”. Para avanzar en el estudio de las comunicaciones en las redes de dispositivos, es necesario definir antes los conceptos de protocolo y de arquitectura de protocolos.

Protocolo

En el ámbito de la comunicación de datos, protocolo es el conjunto de reglas que gobierna el intercambio de datos entre dos entidades. En un computador los protocolos se implementan mediante programas. Estos programas o módulos son activados por el sistema operativo del computador cuando una entidad necesita intercambiar datos con otra entidad en otro computador. Cada módulo se comunica con otro usando un paquete de datos, en el que una parte está constituida por los datos propiamente dichos y otra parte es la cabecera en la que se incluye información de control.

Los conceptos asociados a un protocolo son: sintaxis, semántica y temporización.

- **Sintaxis.** Incluye aspectos como el formato de los datos y los niveles de señal. Por ejemplo, en un paquete de datos la sintaxis tiene que ver con la cantidad de campos del mismo, cómo se ubican y qué tamaño tienen en bytes.
- **Semántica.** Comprende el manejo de la información de control para la coordinación de la comunicación y el manejo de los errores. Con relación al paquete de datos, la semántica consiste en la interpretación del contenido de los campos para la toma de decisiones por parte del módulo que lo maneja.
- **Temporización.** Está referida a las velocidades de envío y recepción de señales y de los paquetes de datos. Por ejemplo, la velocidad de envío de los paquetes desde una entidad a otra no puede ser mayor que la capacidad de esta última para aceptarlos.

Arquitectura de Protocolos

Para que pueda realizarse la comunicación entre computadores resulta claro que se requiere una gran cooperación entre éstos. Debido a la complejidad y a la gran cantidad de operaciones que involucra esta tarea de comunicación, en lugar de implementar toda la lógica en un único módulo, es más sencillo subdividir esta gran tarea en varias subtareas. De esta manera, cada subtarea se realiza por separado y la suma de todas ellas realiza la tarea total. Para distribuir la tarea en varias subtareas es necesario contar con una estructura que se denomina arquitectura de protocolos. Una muestra simple de esta arquitectura se presenta con el siguiente ejemplo:

Transmisión de un Archivo de Datos

Considérese que un computador necesita transferir a otro un archivo de datos. En primer lugar, existe la necesidad de contar con un camino físico por el que circularán los datos entre ambos computadores. Pero, además, se requieren tareas adicionales como las siguientes:

1. **Activación del camino.** El sistema fuente de información debe activar el camino directo de los datos, tanto en el caso de un enlace directo como cuando existe una red de comunicación. En este último caso se debe proporcionar a la red la identificación del sistema destino al cual entregar los datos.
2. **Preparación del sistema destino.** El sistema fuente debe asegurarse de que el sistema destino (en este caso el computador destino) está preparado para recibir los datos.
3. **Preparación de la aplicación destino.** La aplicación que transfiere el archivo debe asegurarse que la aplicación que lo recibe está preparada para aceptar y almacenar el archivo para un usuario determinado.
4. **Compatibilidad en el formato.** Si los formatos de los archivos de ambos sistemas son incompatibles, uno de los dos deberá hacer una tarea de adecuación.

Transferir un archivo entre dos dispositivos (computadores)

La Figura 2.1.1 se muestra la implementación de dicha transmisión mediante el empleo de tres módulos que se describen a continuación:

- **Transferencia de archivos.** Este módulo podría realizar las tareas 3 y 4 descritas en la presentación del ejemplo. En ese sentido, los módulos (uno en cada sistema) tienen que realizar las tareas de preparación (cuando envía) y de verificación (cuando recibe) de los datos de archivo y órdenes. Las órdenes son, por ejemplo, el tamaño del archivo, el tipo de datos, las señalizaciones de comienzo y fin del archivo.
- **Servicio de comunicación.** Este módulo realizará el intercambio de datos y órdenes; es decir, se encargará de la comunicación propiamente dicha. De esta manera se libera al módulo de transferencia de archivos de los detalles del envío de datos y órdenes; en otras palabras, realiza la tarea 2 del ejemplo. Obsérvese que este módulo se encarga de todo lo relacionado con el intercambio entre los computadores, pero no tiene que ver con el tipo de red que los interconecta.
- **Acceso a la red.** Este módulo se encargará de activar un camino entre los computadores a través de la red. En este sentido es razonable disponer de un módulo adicional de acceso a la red que lleve a cabo la tarea 1 del ejemplo.

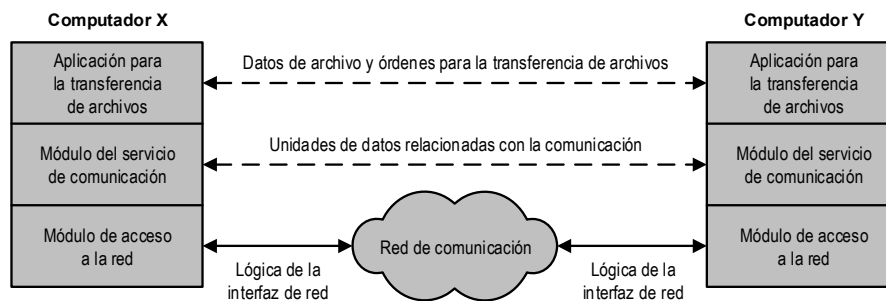


Figura 2.1.1: Transmitir datos entre dos computadores

2.2. Arquitectura de Protocolos simple: modelo de 3 capas

Se puede considerar que en la comunicación de datos por una red de computadores intervienen tres agentes: aplicaciones, computadores y red. Las aplicaciones se ejecutan en computadores y en cada computador pueden correr múltiples aplicaciones simultáneamente. Los computadores se conectan a una red a través de la cual transfieren sus datos. Por lo tanto, la transferencia de datos desde una aplicación a otra implica generarlos y, luego, hacerlos llegar a la aplicación destino a través de la red.

Por lo expuesto, parece natural organizar toda la tarea dividiéndola en tres capas independientes: de acceso a la red, de transporte y de aplicación. Cada capa se define a partir de las funciones que cumple en la comunicación.

Capa de acceso a la red

Está relacionada con el intercambio de datos entre el computador y la red a la que está conectado. Las tareas y cuestiones inherentes a esta capa son las siguientes:

- Dirección del computador destino. El computador emisor deberá proporcionar a la red la dirección del computador destino, de tal forma que ésta pueda encaminar y entregar los datos al destino indicado.
- Servicios de red. El computador emisor necesitará hacer uso de algunos de los servicios provistos por la red como, por ejemplo, la gestión de prioridades cuando el tipo de aplicación así lo requiera.

Además, las características del software de esta capa dependerán de la red que se use, puesto que cada tipo de red responde a una estándar diferente. Por ese motivo resulta muy conveniente separar las funciones que tienen que ver con el acceso a la red en una capa independiente. De esta forma, el resto del software de comunicación que esté por encima de la capa de acceso a la red no tendrá que ocuparse de las características específicas de la red que se use. Es decir, funcionará independientemente del tipo de red particular a la que esté conectado el computador.

Capa de Transporte

Comprende las tareas necesarias para asegurar que los datos de una aplicación lleguen a la otra en forma segura y en el mismo orden en que fueron enviados, independientemente de la naturaleza de las aplicaciones. Debido a que este servicio se proveerá a todas las aplicaciones que corren en el computador, tiene sentido concentrar todos estos procedimientos en una capa separada.

Capa de Aplicación

Especifica la lógica necesaria para admitir varias aplicaciones de usuario. Para cada tipo de aplicación, en esta capa se requerirá un módulo independiente que responda a sus propias necesidades de comunicación. El módulo de cada aplicación debe realizar básicamente dos funciones:

- Procesamiento. Comprende procesar los datos realizando las tareas específicas de la aplicación.
- Organización para la comunicación. Una vez completado el procesamiento y, a continuación, transmitir a otra aplicación los datos generados, el módulo debe organizar dichos datos y la información de control necesaria. Por el contrario, cuando una aplicación requiera de otra aplicación remota un envío de datos, debe generar el pedido correspondiente. En ambos casos (envío o pedido de envío de datos) debe pasar la información a la capa de transporte que se encargará de su comunicación.

Las Figuras 2.2.1 ilustra esta sencilla arquitectura. Se puede observar tres computadores conectados a una red. Cada computador contiene un módulo de software por capa. Cada uno de estos módulos lleva a cabo las funciones de la capa correspondiente: de acceso a la red, de transporte y de aplicación.

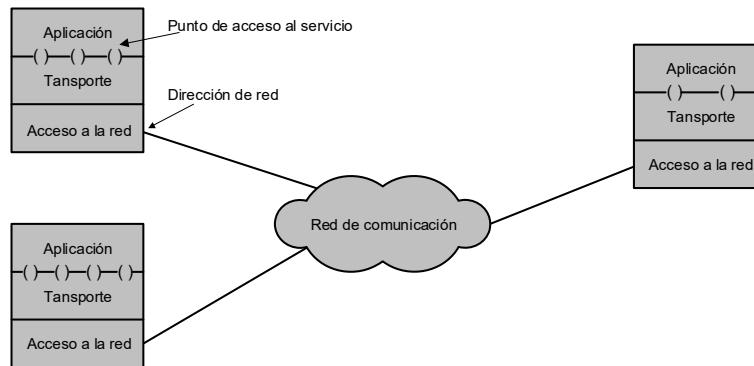


Figura 2.2.1: Arquitectura de protocolos de tres capas

En la arquitectura propuesta hay dos aspectos claves que se deben resolver: la implementación del direccionamiento y el manejo de la información de control.

Direccionamiento

Una cuestión importante a definir en cualquier sistema de comunicación es cómo se realizará la conexión entre las aplicaciones para que el intercambio de los datos se lleve a cabo correctamente. Esto implica que cada aplicación debe ser perfectamente identificable en la red, para lo que es necesario contar con un esquema de direccionamiento apropiado. En este sentido, debe tenerse presente que una red tiene conectados varios computadores y, a su vez, en cada computador están corriendo varias aplicaciones. Por lo tanto, para que pueda realizarse una conexión independiente entre cualquier par de aplicaciones, cada aplicación deberá tener una dirección única en toda la red. Esto implica que se necesitan dos niveles de direccionamiento:

- Dirección de red. Cada computador conectado a la red deberá tener una dirección de red; esto permite a la red entregar los datos al computador apropiado.
- Dirección de aplicación. Además, cada aplicación en el computador deberá tener una dirección que sea única dentro del mismo; esto permitirá al módulo de transporte entregar los datos a la aplicación apropiada. Las direcciones de las aplicaciones se denominan puntos de acceso al servicio o SAP (Service Access Point). En resumen, a través de un SAP el módulo de transporte entrega los datos a una aplicación específica y, a su vez, una aplicación accede individualmente a los servicios proporcionados por la capa de transporte.

Manejo de la Información de Control

Para poder realizar y controlar esta operación es necesario transmitir información de control junto con los datos de la aplicación de usuario, como se muestra en la Figura 2.2.2. Supóngase que la aplicación emisora genera un bloque de datos destinado a la aplicación destino. Como primera medida, pasa dicho bloque a la capa de transporte. Esta capa puede fraccionar el bloque de datos en unidades más pequeñas para hacerlas más manejables. A cada una de estas pequeñas unidades la capa de transporte añadirá una cabecera que contendrá información de control que maneja el protocolo. Esta fracción de los datos generados por la capa superior más la información de control adosada como cabecera se denomina unidad de datos de protocolo o PDU (Protocol Data Unit). En el caso descripto se denominará PDU de transporte.

Información de Control en la Cabecera del PDU de Transporte

Debe recordarse que en la cabecera de la PDU generada por el módulo origen en una determinada capa, se incluye la información que será utilizada por el módulo destino que opera en la misma capa. Así, la cabecera de cada PDU de transporte contiene información que será usada por el módulo de transporte en el computador destino. La información que se debe almacenar en la cabecera es, por ejemplo:

- SAP Destino. Cuando el módulo de transporte destino reciba la PDU de transporte, tomará conocimiento a cual aplicación van dirigidos los datos.
- Número de secuencia. Ya que el módulo de transporte está enviando una secuencia de PDU's como resultado del fraccionamiento de los datos de la aplicación, éstas se numerarán secuencialmente para que, si llegan desordenadas, el módulo de transporte destino sea capaz de ordenarlas.
- Código de detección de error. El módulo transporte emisor debe incluir un código que es función del contenido del resto de la PDU. El módulo de transporte receptor realiza el mismo cálculo y compara los resultados con el código recibido. Si hay discrepancia se concluirá que ha habido un error en la transmisión, en cuyo caso el módulo receptor intentará corregir el error o, simplemente, descartará la PDU y solicitará al otro extremo la retransmisión de la misma.

El siguiente paso en la capa de transporte es pasar cada una de las PDU a la capa de red, con la instrucción de que sean transmitidas al computador destino. Para satisfacer este requerimiento, el módulo de acceso a la red debe pasar los datos a la red con una solicitud de transmisión. Como antes, esta operación requiere el uso de información de control. En este caso, el módulo de acceso a la red añade la cabecera de acceso a la red a los datos provenientes del módulo de transporte, creando así la PDU de acceso a la red.

Información de Control en la Cabecera de la PDU de Acceso a la Red

Por ejemplo, dicha cabecera debe contener la siguiente información:

- Dirección del computador destino. La red debe conocer a qué computador de la red entregará los datos.
- Solicitud de recursos. El protocolo de acceso a la red puede solicitar a la red que realice algunas funciones, como, por ejemplo, gestionar prioridades de envío de los datos.

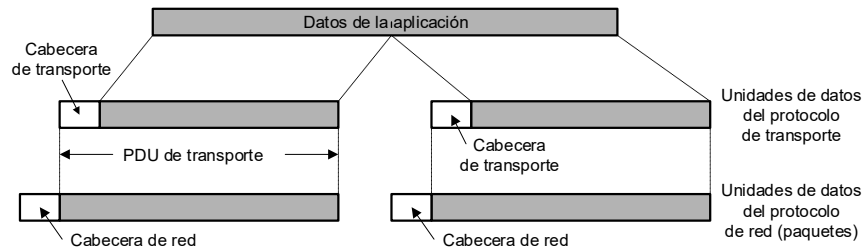


Figura 2.2.2: Información de control en el modelo de tres capas

2.3. Aplicación del Modelo de Tres Capas

En el siguiente ejemplo se describen todos los conceptos vertidos respecto del modelo de tres capas que se está considerando. En la Figura 2.3.1 se muestra la interacción entre los módulos para transferir un bloque de datos. Supóngase que el módulo de transferencia de archivos en el computador X está transfiriendo un archivo, registro por registro, al computador Y. Cada registro se pasa al módulo de la capa de transporte. En un sistema de computación, esta operación se puede describir como una orden o una llamada a un procedimiento. Posibles argumentos de este procedimiento serán el registro del archivo, el SAP destino y la dirección del computador Y. La capa de transporte añade el SAP correspondiente e información de control adicional que se agregará al registro para formar la PDU de transporte. Ésta se pasa a la capa inferior de acceso a la red mediante la llamada a otro procedimiento. En este caso, los argumentos para esta llamada serán la dirección del computador Y junto con la PDU del protocolo de transporte.

A tener en cuenta:

- El campo de datos de la PDU de una capa es la PDU de la capa superior.
- En la cabecera de las PDU del computador origen sólo se incluye información que utilizará el protocolo de la capa del mismo nivel en el computador destino. Así:
 - En la capa de transporte, de la información que le envía la capa de aplicación sólo incluye el SAP destino, al que le suma otra de generación propia como es el número de secuencia y código de error. El resto de la información de control es pasada al módulo de la capa de acceso a la red.

- De forma similar se procede en la capa de acceso a la red. De la información enviada por la capa de transporte sólo utiliza la dirección de Y, a la que le agrega la dirección de X y la solicitud de recursos. Nótese que la cabecera de transporte no es visible al nivel de acceso a la red; en otras palabras, a dicho nivel no le concierne el contenido concreto de la PDU de transporte.
- La red acepta la PDU de red del computador origen y la transmite al destino. El módulo de acceso a la red en el destino recibe la PDU, elimina la cabecera, y pasa la PDU de transporte adjunta al módulo de la capa de transporte. La capa de transporte examina la cabecera de la PDU de transporte y, en función del campo en la cabecera que contenga el SAP, entregará el registro correspondiente a la aplicación pertinente. En el caso del ejemplo 2, lo entrega al módulo de transferencia de archivos de Y.

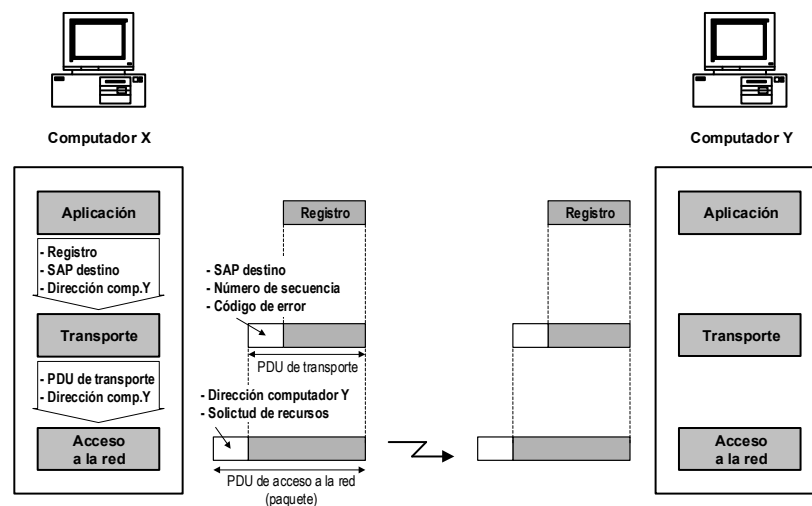


Figura 2.3.1: Aplicación del modelo de tres capas

2.4. Modelo OSI de ISO

El modelo OSI surgió de la necesidad de intercambiar datos entre sistemas heterogéneos en los cuales el hardware, software y el formato de los datos que cada uno genera, tienen características diferentes, con lo cual la comunicación se hace compleja. Por esta razón la ISO - International Organization for Standardization - creó en 1977 un subcomité ad-hoc con el fin de definir un modelo de interfaz común para que sistemas heterogéneos puedan comunicarse. Al año siguiente este subcomité presentó una estructura estratificada en capas que recibió el nombre de Modelo de Referencia para la Interconexión de Sistemas Abiertos (Referencial Model Open System Interconnection), más conocido como modelo OSI.

En 1983 el modelo OSI fue aprobado por la ISO como estándar internacional para la interconexión de sistemas abiertos. El término “abierto” significa que cualquier sistema que adopte las normas creadas por la ISO podrá comunicarse con cualquier otro sistema que se rija por las mismas normas. Por tanto, un sistema abierto es aquel que sigue las normas OSI de la ISO para comunicarse con otros sistemas.

Estructura de Capas

Debido a que comunicar dos dispositivos entre sí constituye un problema sumamente complicado de resolver, el subcomité de la ISO propuso dividir esta compleja tarea en varias tareas menores. Concretamente, ideó un conjunto jerárquico de capas y asignó a cada capa una tarea específica, de manera que, realizando sucesivamente cada tarea específica, al final se resuelve todo el problema de comunicación planteado. El esquema de funcionamiento de las capas es el siguiente:

- Cada capa utiliza los servicios suministrados por la capa inmediata inferior y produce servicios para la capa inmediata superior.
- La capa N no sabe, ni le interesa saber, cuántas capas existen ni cómo realizan sus tareas, solo le importa qué le ofrece la capa adjunta como servicio.

El esquema descrito presenta la gran ventaja de que, al independizar el modo en que cada capa resuelve su problema, se consigue:

- Simplificar la resolución del problema completo al dividirlo en varios problemas menores.
- Flexibilizar el diseño de protocolos de cada capa puesto que no interesa cómo éstos resuelven los problemas, sino que, a partir de determinados servicios que recibe de una capa, deben operar de alguna manera para entregar otros servicios determinados a la otra capa adyacente.

Capas del Modelo OSI

La arquitectura del modelo OSI está formada por siete capas. Para determinar este número la ISO tuvo en cuenta las siguientes pautas:

- Cada capa deberá tener una función bien definida, claramente diferente de las otras. Esto fija un número mínimo de capas.
- El número máximo de capas no debe ser tan alto a fin de no dificultar el trabajo de descripción e integración de cada capa.
- Demarcar el límite entre dos capas adyacentes en el punto donde la transferencia de servicios sea pequeña, minimizando las interacciones entre las capas.
- Cada capa solo tiene interfaces con la capa superior y con la inferior.

- Garantizar que la modificación de los protocolos que realizan las funciones de una capa no afecte a las otras capas.

En la Figura 2.4.1 se muestra el esquema de siete capas del modelo OSI de la ISO. Es importante tener presente que un modelo de referencia de capas como el presentado no necesariamente describe una arquitectura de comunicación, puesto que no especifica los servicios ni los protocolos que se utilizarán en cada capa. Solo indica qué funciones se le asignan a cada capa, pero no cómo las debe realizar. El modo en cómo se realizan las funciones de una capa corresponde a los protocolos. No obstante, la ISO también ha generado normas que deben seguir los protocolos para todas las capas, pero ello no significa que estas normas formen parte del modelo. Ellas se han publicado como normas internacionales independientes.

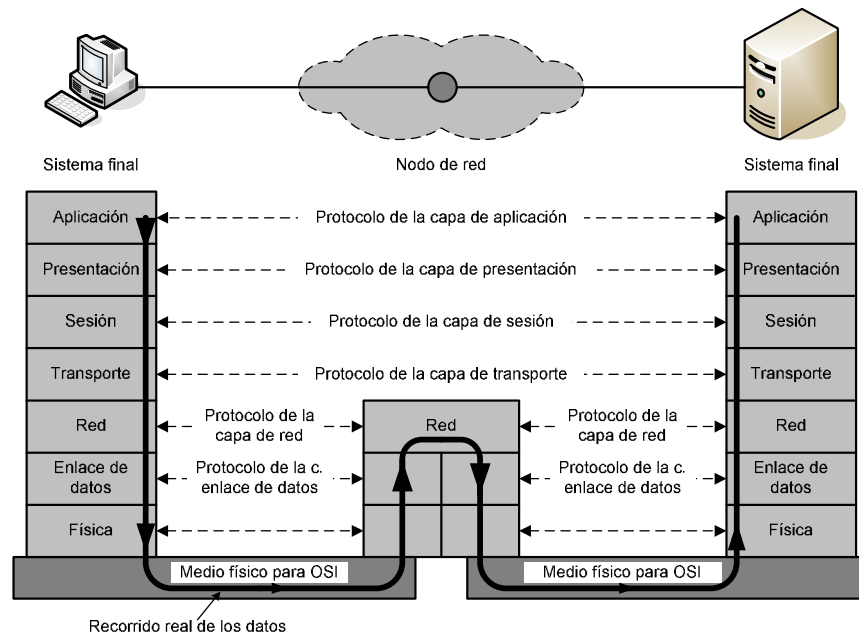


Figura 2.4.1: Capas del modelo OSI

Capa Física

En esta capa se define la interfaz física entre los dispositivos y, además, cómo debe realizarse la transmisión de las señales a través del medio de transmisión. Concretamente, las consideraciones de diseño del hardware están relacionadas con asegurar que cuando un extremo envía un bit de valor 1 en el otro debe recibirse un 1 y no un 0. Por lo tanto, para lograr lo señalado, en esta capa se deben realizar básicamente las siguientes especificaciones:

- Mecánicas. Están referidas a los elementos que intervienen en la conexión física de la interfaz. Normalmente dentro de estas características se incluye la especificación de conectores que vinculan los circuitos electrónicos con los cables. Especifica dimensiones físicas y cantidades, por ejemplo, número, tamaño y distribución de pines de un conector.

- Eléctricas. Está referidas a las señales que generan los circuitos electrónicos para representar los bits: código, velocidad de transmisión, niveles de corriente o tensión, tiempo de duración de los pulsos y separación de los mismos.
- Funcionales. Tienen que ver básicamente con las funciones de cada circuito de la interfaz física y el modo de transmisión. Este último está referido a si la transmisión será en ambos sentidos, uno por vez (half dúplex) o simultáneamente (full dúplex).
- De procedimientos. Están relacionadas con la secuencia de eventos que se deben llevar a cabo para el intercambio del flujo de señales. Concretamente, con la sincronización entre los circuitos electrónicos de los extremos para establecer la conexión inicial y luego interrumpirla cuando los dispositivos han decidido finalizar la transmisión.

Capa de Enlace de Datos

La tarea principal de esta capa consiste en lograr que la comunicación por un enlace físico entre dos dispositivos sea sin error. Esto implica que, a partir de un medio de transmisión real (con ruido, interferencias y atenuación que producen errores en la transmisión), la transferencia de datos se realice como si se tratara de un medio ideal, es decir, sin errores de transmisión.

Las funciones básicas de la capa de enlace están relacionadas con el control del enlace punto a punto entre dispositivos que están en los extremos del mismo. Realizar el control del enlace implica un conjunto de tareas coordinadas entre los dispositivos que comprenden básicamente el acceso al medio de transmisión y la iniciación, mantenimiento y terminación del intercambio de datos. A continuación, se identifican las principales tareas involucradas en el control del enlace:

- Sincronización de trama. Los datos enviados por un dispositivo emisor son fraccionados en tramas que son grupos de bytes. El principio y fin de cada trama deben ser reconocidos por el receptor, para lo cual el emisor agrega bytes especiales en ambos extremos.
- Control de flujo. La estación que envía los datos no debe hacerlo más rápidamente que la capacidad de la estación receptora para almacenarlos. Debe emplearse, entonces, un mecanismo de regulación de tráfico que permita que el transmisor conozca el espacio de memoria disponible (buffer) que en ese momento tiene el receptor para ese fin.
- Control de error. Los bits erróneos de la transmisión deben ser corregidos en cada trama. Para ello se generan en el transmisor bits especiales - mediante un código de control de error - que son agregados al final de la trama, a partir de los cuales el receptor puede detectar y corregir los errores producidos en el enlace.

Controles en más de una capa:

Hay dos controles, importantes en la comunicación de datos entre sistemas, que por conveniencia se realizan en varias capas que son el control de error y el

control de flujo de los datos. Cada capa resuelve estos problemas de acuerdo a las funciones asignadas de acuerdo al nivel que ocupa en la estructura jerárquica. El resultado final es que el flujo de datos transmitido soporta distintos tipos de controles con el fin de minimizar o anular los problemas señalados.

Capa de Red

Básicamente, esta capa se especifica sobre la transferencia de información entre dos sistemas finales a través de una red. Libera así a las capas superiores de la tarea de la transmisión de datos teniendo en cuenta la tecnología propia de la red. Las funciones de esta capa están relacionadas con dos tareas básicas:

- El dispositivo emisor establece un diálogo con la red para especificarle la dirección del dispositivo destino y solicitar ciertos servicios, como, por ejemplo, la gestión de prioridades según el tipo de dato que transmite (voz, video o datos).
- La red debe realizar las acciones necesarias para que los paquetes de datos que llevan el mensaje atraviesen la red evitando en lo posible y solucionando cuando ocurran, congestiones de tráfico en enlaces y dispositivos internos de la red.

Existe un amplio abanico de posibilidades para que los servicios de comunicación intermedios sean gestionados por la capa de red. Se tratarán a continuación tres casos representativos siguiendo un orden de complejidad creciente:

- Dispositivos conectados a través de un enlace punto a punto. Es el extremo más sencillo. En este caso no se necesita capa de red ya que la capa de enlace de datos puede proporcionar las funciones necesarias de gestión.
- Dispositivos conectados a través de una única red. Puede ser una red de conmutación de paquetes o de conmutación de circuitos. Un ejemplo de esta última situación es el estándar X.25 cuya capa de paquete tiene las funciones típicas de una capa de red. Los paquetes creados por los sistemas finales pasan a través de uno o más nodos de la red que actúan como retransmisores entre los dos dispositivos. Los nodos de la red implementan las capas 1 a 3 de la arquitectura. La capa 3 del nodo realiza las funciones de ruteo y conmutación; es decir, decide sobre la ruta más conveniente y, luego, por ella envía los datos.
- Dispositivos conectados a redes distintas. Es el extremo de mayor complejidad. Este caso no está contemplado en el modelo OSI.

En resumen, las funciones principales de la capa de red son:

- Gestión de servicios de red. Solicitados por los dispositivos de usuario como, por ejemplo, prioridades.
- Ruteo. Los protocolos de la capa de red en los nodos deben decidir cómo encaminar los paquetes desde el origen, pasando por los distintos nodos de conmutación de la red, hasta el destino. El camino que van a seguir los paquetes

a través de la red depende de varios factores, como carga existente en la red y estado de los enlaces.

- Control de tráfico. A esta capa también le cabe la función de control de tráfico con el fin de evitar la congestión en los enlaces de la red. Para poder realizar esta función, el nodo debe tener información sobre el estado de los enlaces de la red.

Capa de Transporte

La función principal de la capa de transporte consiste en que los datos generados por la aplicación en un computador, lleguen a la aplicación que corre en otro computador, libres de errores, en orden, sin pérdidas ni duplicaciones. Se trata, básicamente, de proporcionar una “comunicación de extremo a extremo” entre dos entidades finales. Una característica distintiva de esta capa, en relación con las capas ya vistas, es que no está involucrada con la transmisión en la red sino con las necesidades de comunicación de los sistemas finales. Se dice que los protocolos de la capa de transporte, en dos sistemas finales que intercambian datos, mantienen una comunicación virtual. La capa de transporte también está involucrada en la solicitud de los servicios de red necesarios según el tipo de aplicación. A continuación, se especifican con mayor detalle las funciones más importantes de esta capa:

- Secuenciamiento. Referida al ordenamiento de los segmentos que se transmiten.
- Segmentación de datos
- Control de flujo. A diferencia del control de flujo que involucra a la capa de enlace de datos, la capa de transporte realiza este control teniendo en cuenta las especificaciones de las capas superiores de tal manera de no saturar los procesos que corren en el sistema (aplicaciones).
- Tipo de servicio o conexión. Tiene que ver con la decisión que se debe tomar respecto al tipo de servicio o de conexión a establecerse de acuerdo a las necesidades de la capa superior. Por ejemplo, define el modo en que deben enviarse los datos; esto es, tamaño y velocidad de transferencia de los paquetes, cantidad de conexiones lógicas que se deben establecer. Respecto a este último servicio puede ocurrir, por ejemplo, que una aplicación requiera comunicarse en forma simultánea con varias aplicaciones que corren en otros computadores. En este caso, el protocolo de la capa de transporte debe proveer tantos canales lógicos como aplicaciones destino haya y velocidades de transferencia requeridas.

Capa de Sesión

A esta capa le incumbe que los usuarios de diferentes computadores puedan establecer sesión entre ellos. Una sesión implica el intercambio de datos entre dos aplicaciones que concierne a la capa de transporte; pero, además, necesita de otros servicios adicionales a los proporcionados por esta capa, con el fin de asegurar que la operación se realice sin inconvenientes. Las funciones típicas de la capa de sesión son:

- Control de diálogo entre aplicaciones. Se explica con el siguiente caso: Una sesión es, por ejemplo, cuando el usuario de una PC se conecta en modo terminal a un

sistema de tiempo compartido para transferir un archivo. En este caso los servicios de la capa de sesión consisten en realizar el control del diálogo; por ejemplo, que en un instante dado la comunicación vaya en ambas direcciones simultáneamente y, en otro instante, sólo en una dirección. La capa de sesión, en este caso, otorga y controla los turnos de emisión de los datos entre los dispositivos.

- En el caso de la sesión en tiempo compartido entre un computador central y varios terminales, la capa de sesión es la encargada de administrar ese diálogo punto-multipunto. Esto es, al comienzo del diálogo con un terminal dado debe restituir el estado en quedó la comunicación la última vez para que la misma se reanude sin problemas. Luego controla el diálogo durante el intervalo de tiempo como se señaló en el párrafo anterior y, una vez finalizado, almacena el estado de la comunicación hasta el próximo intervalo. Así, para cada terminal.
- Sincronización. Considérense, por ejemplo, los problemas que podrían ocurrir si se tratara de hacer la transferencia de un archivo de gran tamaño, en una red con poca confiabilidad. Si no existe sincronización, después de abortar cada envío del archivo por la caída de la conexión, tendría que iniciarse de nuevo la transferencia completa del mismo y, probablemente, se encontraría nuevamente con otra caída de la conexión. Para eliminar este problema, la capa de sesión proporciona una forma para insertar puntos de verificación en el flujo de datos con el objeto de que, después de cada caída, solamente tengan que repetirse los datos que se encuentren después del último punto de verificación.

Capa de Presentación

Esta capa se ocupa de los aspectos de sintaxis y semántica de la información que se transmite. En esto se diferencia de las capas inferiores las que únicamente están interesadas en la transferencia confiable de los datos de un lugar a otro. Las funciones típicas asociadas a la capa de presentación son:

- Codificación. La mayor parte de los programas de aplicación no intercambian conjuntos de bits binarios aleatorios, sino, más bien, nombre de personas, datos, cantidades de dinero, números de facturas, etc. Estos datos se representan por conjuntos de caracteres, números enteros, números de punto flotante y estructuras de datos constituidas por elementos más simples. Puede ocurrir que diferentes computadores tengan diferentes códigos para representar los conjuntos de caracteres como, por ejemplo, ASCII o EBCDIC. Con el fin de hacer posible la comunicación entre computadores con representaciones diferentes, se puede definir una representación estándar. Entonces, la tarea de la capa de presentación es manejar la representación estándar y convertirla a la propia de cada computador y viceversa.
- Seguridad. Para brindar seguridad a la transferencia de los datos la capa de presentación indica el encriptado de mensajes para dotar de privacidad y autenticación a la comunicación.

- **Compresión.** A los efectos de disminuir la cantidad de bits a transmitirse, la capa de presentación se ocupa también de la compresión de datos. En esta capa operan los algoritmos encargados de esa tarea.

Capa de Aplicación

Esta capa tiene que ver básicamente con la interfaz aplicación-usuario y la interfaz aplicación-servicios de red. Los servicios de transferencias de archivos (FTP), el correo electrónico, los servicios de páginas web, la resolución de nombres de dominio, entre otros, son ejemplos de aplicaciones que se despliegan sobre esta capa.

2.5.Arquitectura de Protocolos TCP/IP

En términos generales, se puede decir que las comunicaciones por computadora involucran a tres protagonistas: aplicaciones, computadoras y redes.

Ejemplos de aplicaciones correo electrónico, servicios web, transferencia de archivos, entre otros. Las aplicaciones que interesan aquí son aplicaciones distribuidas que implican el intercambio de datos entre dos sistemas de computadoras. Estas aplicaciones, y otras, se ejecutan en computadoras que a menudo pueden soportar múltiples aplicaciones simultáneas. Las computadoras están conectadas a redes, y la red transfiere los datos que deben intercambiar desde dispositivo a otro. De esta forma, la transferencia de datos de una aplicación a otra implica primero llevar los datos a la computadora en la que reside la aplicación y luego obtener los datos a la aplicación prevista dentro de la computadora. Teniendo en cuenta estas premisas, es posible organizar la comunicación haciendo uso de cinco capas relativamente independientes (Figura 2.5.1):

- Capa física
- Acceso a la red / capa de enlace de datos
- capa de Internet
- Capa de host a host o de transporte
- Capa de aplicación

La capa física cubre la interfaz física entre un dispositivo de transmisión de datos (por ejemplo, estación de trabajo, notebook, Smartphone, entre otros) y un medio de transmisión o red. Esta capa se ocupa de especificar las características del medio de transmisión, la naturaleza de las señales, la velocidad de datos y asuntos relacionados.

La capa de acceso a la red / enlace de datos está preocupada del acceso y enrutamiento de datos a través de una red para dos sistemas finales conectados a la misma red. En aquellos casos en los que dos dispositivos están conectados a diferentes redes, se necesitan procedimientos para permitir que los datos atraviesen múltiples redes interconectadas. Ésta es la función de la capa de internet. Se usa el Protocolo de Internet

(IP) en esta capa para proporcionar la función de enrutamiento a través de múltiples redes. Este protocolo se implementa no solo en los sistemas finales sino también en los enrutadores. Un enrutador es un procesador que conecta dos redes y cuya función principal es transmitir datos de una red al otro en su ruta desde el sistema final de origen al destino.

La capa de host a host, o capa de transporte, puede proporcionar un extremo a otro confiable servicio, como se discutió en la sección anterior, o simplemente un servicio de entrega de extremo a extremo sin mecanismos de fiabilidad. El Protocolo de Control de Transmisión (TCP) es el protocolo más común utilizado para proporcionar esta funcionalidad.

Finalmente, la capa de aplicación contiene la lógica necesaria para soportar las distintas aplicaciones de usuario. Para cada tipo diferente de aplicación, como la transferencia de archivos, se necesita un módulo que sea peculiar de esa aplicación.

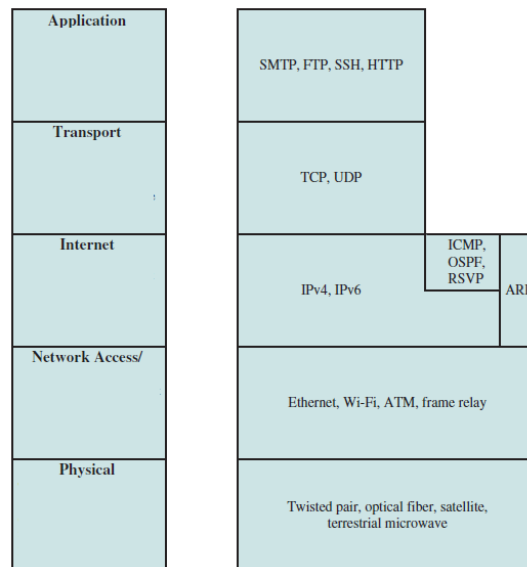


Figura 2.5.1: Arquitectura de Protocolos TCP/IP

La Figura 2.5.2 indica cómo se configuran estos protocolos para las comunicaciones. Es muy claro que las necesidades de comunicación pueden implicar la presencia de muchas redes (en este caso se denominan subredes).

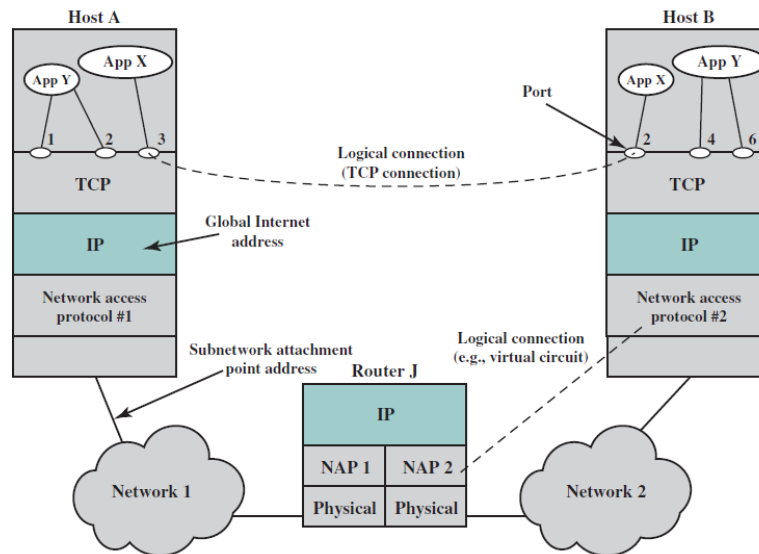


Figura 2.5.2: Métodos de comunicación utilizando la arquitectura TCP/IP

Se utiliza algún tipo de protocolo de acceso a la red, como la lógica de Ethernet o Wi-Fi, para conectar una computadora a una subred. Este protocolo permite que el host envíe datos a través de subred a otro host o, si el host de destino está en otra subred, a un enrutador que reenviará los datos. IP se implementa en todos los sistemas finales y los enrutadores. El enrutador tiene la capacidad de encaminar bloques de datos desde un host, a través de uno o más enrutadores, hacia otro host. TCP se implementa solo en los sistemas finales; mantiene el seguimiento de los bloques de datos para asegurar que todos se entreguen de manera adecuada y confiable.

Como se mencionó anteriormente en este capítulo, cada entidad en el sistema general debe tener una dirección única. Cada host en una subred debe tener una dirección de internet global única. Esto permite que los datos se entreguen al host adecuado. Cada proceso dentro de un dispositivo debe tener una dirección de identificación única (puerto), esto permite al protocolo TCP entregar los datos host a host.

Se desarrolla una operación simple. Se supone que un proceso, asociado con el puerto 3 en el host A, desea enviar un mensaje a otro proceso, asociado con el puerto 2 en host B. El proceso en A entrega el mensaje a TCP con instrucciones para enviar al host B, puerto 2. TCP entrega el mensaje a IP con instrucciones para enviarlo al host B. Tenga en cuenta que no es necesario que se le diga a la IP la identidad del puerto de destino. Todo lo que necesita saber es que los datos están destinados al host B. Luego, IP entrega el mensaje hasta la capa de acceso a la red (por ejemplo, lógica de Ethernet) con instrucciones para enviarla a enrutador J (el primer salto en el camino a B).

Para controlar esta operación, la información de control, así como los datos del usuario, deben ser transmitidos, como se sugiere en la Figura 2.5.3. Digamos que el proceso de envío genera un bloque de datos y pasa esto a TCP. TCP puede romper este bloque en pedazos más pequeños para hacerlo más manejable. Para cada una de estas piezas, TCP agrega información de control conocido como el encabezado TCP, formando un segmento TCP. La información de control debe ser utilizado por la entidad TCP igual en el host B. Ejemplos de elementos en este encabezado incluir:

Puerto de destino: cuando la entidad TCP en B recibe el segmento, debe saber a quién se entregarán los datos.

Número de secuencia: TCP numera los segmentos que envía a un determinado puerto de destino secuencialmente, de modo que, si llegan fuera de servicio, la entidad TCP en B puede reordenarlos.

Suma de verificación: el TCP de envío incluye un código que es función del contenido del resto del segmento. El TCP receptor realiza el mismo cálculo y compara el resultado con el código entrante. Una discrepancia resulta si ha habido algún error en la transmisión.

Luego, TCP entrega cada segmento a IP, con instrucciones para transmitirlo a B. Estos segmentos deben transmitirse a través de una o más subredes y retransmitirse a través de uno o más enrutadores intermedios. Esta operación también requiere el uso de información de control

Por lo tanto, IP agrega un encabezado de información de control a cada segmento para formar un datagrama de IP. Un ejemplo de un elemento almacenado en el encabezado IP es la dirección de host de destino (en este ejemplo, B).

Finalmente, cada datagrama IP se presenta a la capa de acceso a la red para su transmisión a través de la primera subred en su viaje hacia el destino. El acceso a la red capa agrega su propio encabezado, creando un paquete o marco. El paquete se transmite a través de la subred al enrutador J. El encabezado del paquete contiene la información que la subred necesita transferir los datos a través de la subred.

En el enrutador J, el encabezado del paquete se elimina y se examina el encabezado IP. Sobre la base de la información de la dirección de destino en el encabezado IP, el módulo IP en el enrutador dirige el datagrama a través de la subred 2 a B. Para hacer esto, el datagrama se aumenta nuevamente con un encabezado de acceso a la red.

Cuando los datos se reciben en B, se produce el proceso inverso. En cada capa, el encabezado correspondiente se elimina y el resto se pasa al siguiente más alto capa, hasta que los datos originales del usuario se entreguen al proceso de destino.

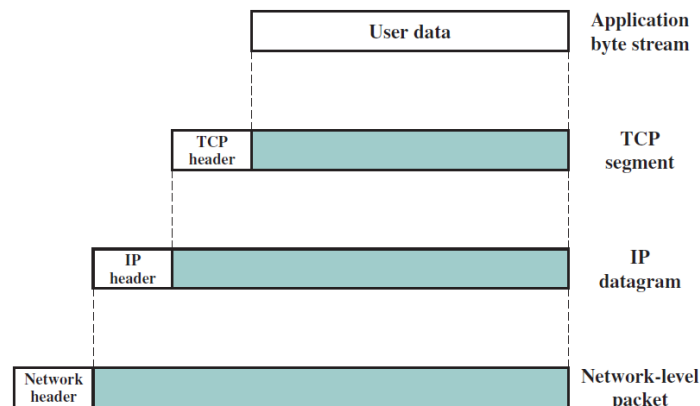


Figura 2.5.3: Encabezados en el modelo TCP/IP

2.6.Arquitectura de protocolos y los estándares

Una arquitectura de protocolo, como la arquitectura TCP / IP u OSI, proporciona un marco para la estandarización. Dentro del modelo, uno o más estándares de protocolo pueden ser desarrollado en cada capa. El modelo define en términos generales las funciones a ser implementadas en cada capa y facilita el proceso de elaboración de normas de dos maneras:

- Debido a que las funciones de cada capa están bien definidas, se pueden desarrollar estándares independientemente y simultáneamente para cada capa. Esto acelera el proceso de elaboración de normas.
- Debido a que los límites entre las capas están bien definidos, los cambios en los estándares en una capa no afectan el software ya existente en otra capa. Esto facilita la introducción de nuevos estándares.

La figura 2.6.1 ilustra el uso de una arquitectura de protocolo como marco para la producción de estándares:

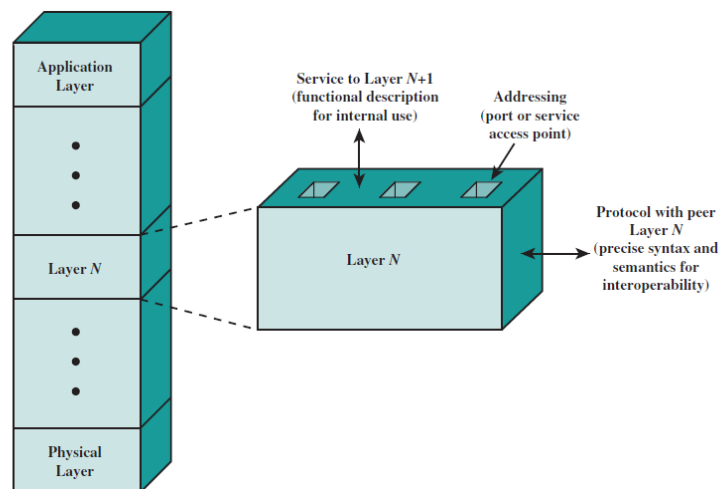


Figura 2.6.1: Arquitectura de protocolos y estándares

Se utiliza el principio de diseño de la ocultación de información: las capas inferiores se preocupan por mayores niveles de detalle; las capas superiores son independientes de estos detalles. Cada capa proporciona servicios a la siguiente capa superior e implementa un protocolo para la capa par en el sistema destino.

La figura 2.6.1 también muestra más específicamente la naturaleza de la estandarización requerida. Tres elementos son clave:

- Especificación de protocolo: dos entidades en la misma capa en diferentes sistemas cooperan e interactúan por medio de un protocolo. Debido a que dos sistemas abiertos están involucrados, el protocolo debe especificarse con precisión. Esto incluye el formato de las unidades de datos de protocolo

intercambiadas, la semántica de todos los campos, y la secuencia permitida de PDU.

- Definición del servicio: además del protocolo o protocolos que operan en una capa dada, se necesitan estándares para los servicios que cada capa proporciona a la siguiente capa superior. Típicamente, la definición de servicios es equivalente a una descripción funcional que define qué servicios se proporcionan, pero no cómo los servicios deben ser proporcionados.
- Direccionamiento: cada capa proporciona servicios a entidades en la siguiente capa superior. Se hace referencia a estas entidades por medio de un puerto o punto de acceso al servicio (SAP). Por lo tanto, un punto de acceso al servicio de red (NSAP) indica una entidad de transporte que es un usuario del servicio de red.

Primitivas de servicio y parámetros

Los servicios entre capas adyacentes en una arquitectura de protocolo se expresan en términos de primitivas y parámetros. Una primitiva especifica la función a ser realizada, y los parámetros se utilizan para pasar datos e información de control. La forma real de una primitiva depende de la implementación. Un ejemplo es una llamada a procedimiento.

Se utilizan cuatro tipos de primitivas en los estándares para definir la interacción entre capas adyacentes en la arquitectura (Figura 2.6.2). Estas son:

- Solicitar: Una primitiva emitida por una capa usuaria para invocar algún servicio de una capa proveedora y pasar los parámetros necesarios para especificar completamente el servicio solicitado. Ejemplo: “Connect.request”
- Indicar: Una primitiva emitida por un proveedor de servicios para, ya sea:
 - Indicar que el usuario del servicio par ha invocado un procedimiento en la conexión y para proporcionar los parámetros asociados, o
 - Notificar a la capa usuaria del servicio de una acción iniciada por la capa proveedora. Ejemplo: “Connect.indication”
- Responder: Una primitiva emitida por la capa usuaria del servicio para reconocer o completar algún procedimiento invocado previamente. Ejemplo: “Connect.Response”
- Confirmar: Una primitiva emitida por un proveedor de servicios para reconocer o completar algún procedimiento invocado previamente por una solicitud de la capa usuaria del servicio. Ejemplo: “Connect.confirm”

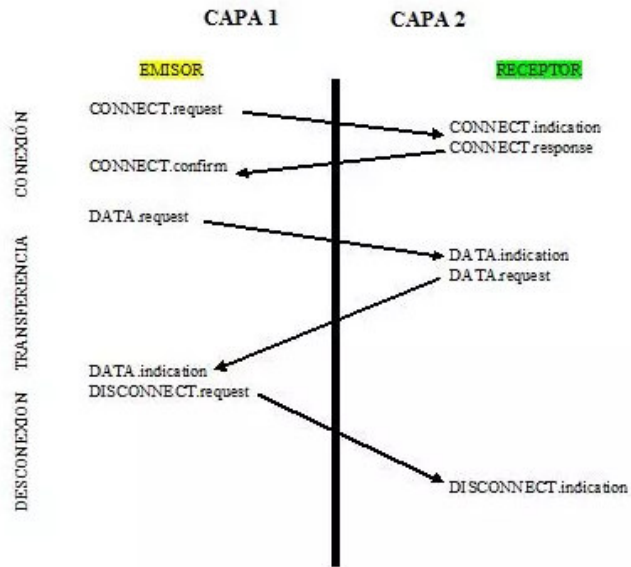


Figura 2.6.2: Comunicación entre dos capas utilizando primitivas