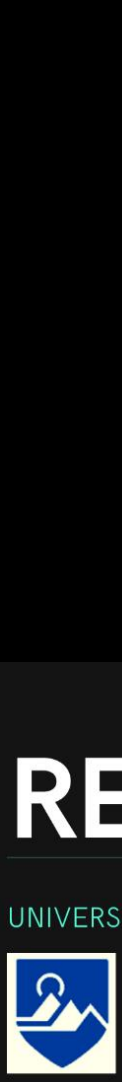
REDES

UNIVERSIDAD DE LA PUNTA



TEMA 2

PROFESORA ING. ASTRI ANDRADA

2021

Tabla de contenido

Arquitectura de protocolos

Las primeras redes de computadoras se diseñaron teniendo en cuenta al hardware como punto principal y al software como secundario. Pero esta estrategia ya no funciona. Ahora el software de red está muy estructurado.

En el intercambio de datos entre computadoras, terminales y/u otros dispositivos de procesamiento, los procedimientos involucrados pueden llegar a ser bastante complejos. Considérese, por ejemplo, la transferencia de un archivo entre dos computadoras. En este caso, debe haber un camino entre las dos computadoras, directo o a través de una red de comunicación, pero además, normalmente se requiere la realización de las siguientes tareas adicionales:

1. El sistema fuente de información debe activar un camino directo de datos o bien debe proporcionar a la red (subred) de comunicación la identificación del sistema destino deseado.
2. El sistema fuente debe asegurarse de que el destino está preparado para recibir datos.
3. La aplicación de transferencia de archivos en el origen debe asegurarse de que el programa gestor en el destino está preparado para aceptar y almacenar el archivo para el usuario determinado.
4. Si los formatos de los dos archivos son incompatibles en ambos sistemas, uno de los dos deberá realizar una operación de traducción.

Para reducir la complejidad de su diseño, en lugar de implementar toda la lógica para llevar a cabo la comunicación en un único nivel, la mayoría de las redes se organizan como una pila de capas o niveles, cada una construida a partir de la que está abajo. El problema se divide en subtareas, cada una de las cuales se realiza por separado. En una arquitectura de protocolos, los distintos módulos se disponen formando una pila vertical. Cada capa de la pila realiza el subconjunto de tareas relacionadas entre sí que son necesarias para comunicar con el otro sistema. Por lo general, las funciones más básicas se dejan a la capa inmediatamente inferior, olvidándose en la capa actual de los detalles de estas funciones. Además, cada capa proporciona un conjunto de servicios a la capa inmediatamente superior. Idealmente, las capas deberían estar definidas de forma tal que los cambios en una capa no deberían necesitar cambios en las otras.
Evidentemente, para que haya comunicación se necesitan dos entidades, por lo que debe existir el mismo conjunto de funciones en capas en los dos sistemas. La comunicación se consigue haciendo que las capas correspondientes, o pares, intercambien información. Las capas pares se comunican intercambiando bloques de datos que verifican una serie de reglas o convenciones denominadas **protocolo**. Los aspectos clave que definen o caracterizan a un protocolo son:

La sintaxis: establece cuestiones relacionadas con el formato de los bloques de datos.

La semántica: incluye información de control para la coordinación y la gestión de errores.

La temporización: considera aspectos relativos a la sintonización de velocidades y secuenciación.

Una arquitectura de protocolos simple

Habiendo definido el concepto de protocolo, podemos definir el concepto de **arquitectura de protocolos**. A modo de ejemplo, la Ilustración 5 muestra cómo se podría implementar una aplicación de transferencia de archivos. Para ello se usan tres módulos. Las tareas 3 y 4 de la lista anterior se podrían realizar por el módulo de transferencia de archivos. Los dos módulos de

ambos sistemas intercambian archivos y órdenes. Sin embargo, en vez de exigir que el módulo de transferencia se encargue de los detalles con los que se realiza el envío de datos y órdenes, dichos módulos delegan en otros módulos que ofrecen el servicio de transmisión. Cada uno de estos se encargará de asegurar que el intercambio de órdenes y datos se realice fiablemente. Entre otras cosas, estos módulos realizarán la tarea 2, por lo que, a partir de este momento, la naturaleza del intercambio entre los sistemas será independiente de la naturaleza de la red que los interconecta. Por tanto, en vez de implementar la interfaz de red en el módulo de servicio de transmisión, tiene sentido prever un módulo adicional de acceso a la red que lleve a cabo la tarea 1, interaccionando con la red.

Resumiendo, el módulo de transferencia de archivos contiene toda la lógica y funcionalidades que son exclusivas de la aplicación, como por ejemplo la transmisión de palabras de paso clave, de órdenes de archivo o de los registros del archivo. Es necesario que esta información (archivos y órdenes) se transmita de una forma fiable. No obstante, estos mismos requisitos de fiabilidad son compartidos por otro tipo de aplicaciones (como por ejemplo, el correo electrónico y la transferencia de documentos). Por tanto, estas funcionalidades se localizan en el módulo separado del servicio de comunicaciones de tal forma que puedan ser utilizadas por otras aplicaciones. El módulo del servicio de comunicaciones trata de asegurar que las dos computadoras estén activas y preparadas para la transferencia de datos, así como de seguir la pista de los datos que se intercambian, garantizando su envío. No obstante, estas tareas son independientes del tipo de red que se esté usando. Por tanto, la lógica encargada de tratar con la red se considera en un módulo separado de acceso a la misma. De esta forma, si se modifica la red que se esté usando, sólo se verá afectado el módulo de acceso a la red.

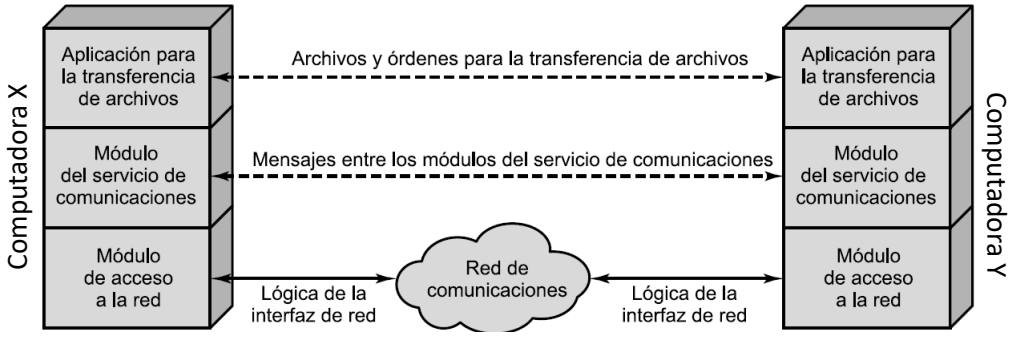


Ilustración 1 - Arquitectura simplificada para la transferencia de archivos.

Así, en vez de disponer de un solo módulo que realice todas las tareas involucradas en la comunicación, se considera una estructura consistente en un conjunto de módulos que realizarán todas las funciones. Esta estructura se denomina **arquitectura de protocolos**. Llegados a este punto, la siguiente analogía puede ser esclarecedora. Supóngase que un ejecutivo en una oficina, digamos X, necesita enviar un documento a una oficina Y. El ejecutivo en X prepara el documento y quizá le añada una nota. Esto es análogo a las tareas que realiza la aplicación de transferencia de archivos de la Ilustración 5. A continuación, el ejecutivo le pasa el documento a un secretario o administrativo (A). El A de X mete el documento en un sobre y escribe en él la dirección postal de Y, así como el remite correspondiente a la dirección de X. Puede que en el sobre se escriba igualmente «confidencial». Lo realizado por A corresponde con el módulo del servicio de comunicaciones de la Ilustración. Llegados aquí, A pasa el sobre al departamento de envíos. Alguien aquí decide cómo enviar el paquete: mediante correo o mensajería. Se añaden los documentos necesarios al paquete y se realiza el envío. El departamento de envíos corresponde al módulo de acceso a la red de Ilustración 5. Cuando el paquete llega a Y, se desencadena una serie de operaciones similares en capas. El departamento de envíos en Y recibe el paquete y lo pasa al administrativo correspondiente, dependiendo del

destino que figure en el paquete. El A abre el paquete, extrae el documento y se lo pasa al ejecutivo correspondiente.

En términos muy generales, se puede afirmar que las comunicaciones involucran a tres agentes: aplicaciones, computadoras y redes. Las aplicaciones se ejecutan en computadoras que, generalmente, permiten múltiples aplicaciones simultáneas. Las computadoras se conectan a redes y los datos a intercambiar se transfieren por la red de una computadora a otra. Por tanto, la transferencia de datos desde una aplicación a otra implica, en primer lugar, la obtención de los mismos y, posteriormente, hacerlos llegar a la aplicación destino en la computadora remota.

Teniendo esto presente, parece natural estructurar las tareas de las comunicaciones en tres capas relativamente independientes: la capa de acceso a la red, la capa de transporte y la capa de aplicación.

La **capa de acceso a la red** está relacionada con el intercambio de datos entre la computadora y la red a la que está conectado. La computadora emisora debe proporcionar a la red la dirección del destino, de tal forma que la red pueda encaminar los datos al destino apropiado. La computadora emisora necesitará hacer uso de algunos de los servicios proporcionados por la red, como por ejemplo la gestión de prioridades. Las características del software de esta capa dependerán del tipo de red que se use. Se pretende separar las funciones que tienen que ver con el acceso a la red en una capa independiente. Haciendo esto, el resto del software de comunicaciones 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. El mismo software de las capas superiores debería funcionar correctamente con independencia del tipo de red concreta a la que se esté conectado.

Independientemente de la naturaleza de las aplicaciones que estén intercambiando datos, es un requisito habitual que los datos se intercambien de una manera fiable. Esto es, sería deseable estar seguros de que todos los datos llegan a la aplicación destino y, además, llegan en el mismo orden en que fueron enviados. Como se verá, los mecanismos que proporcionan dicha fiabilidad son independientes de la naturaleza de las aplicaciones. Por tanto, tiene sentido concentrar todos estos procedimientos en una capa común que se comparta por todas las aplicaciones, denominada **capa de transporte**.

Finalmente, la **capa de aplicación** contiene la lógica necesaria para admitir varias aplicaciones de usuario. Para cada tipo distinto de aplicación, como por ejemplo la transferencia de archivos, se necesita un módulo independiente y con características bien diferenciadas.

En la práctica, el número de capas, su nombre, el contenido de cada una y su función difieren de una red a otra. El propósito de cada capa es ofrecer ciertos servicios a las capas superiores, mientras les oculta los detalles relacionados con la forma en que se implementan los servicios ofrecidos.

En realidad este concepto es familiar y se utiliza en muchas áreas de las ciencias computacionales, en donde se le conoce de muchas formas: ocultamiento de información, tipos de datos abstractos, encapsulamiento de datos y programación orientada a objetos. La idea fundamental es que una pieza particular de software (o hardware) provee un servicio a sus usuarios pero mantiene ocultos los detalles de su estado interno y los algoritmos que utiliza.

Cuando la capa n en una máquina lleva a cabo una conversación con la capa n en otra máquina, a las reglas y convenciones utilizadas en esta conversación se les conoce como el **protocolo de la capa n**. Cómo ya se vio, en esencia, un protocolo es un acuerdo entre las partes que se comunican para establecer la forma en que se llevará a cabo esa comunicación.

Realmente no se transfieren datos de manera directa desde la capa n de una máquina a la capa n de otra máquina, sino que cada capa pasa los datos y la información de control a la capa inmediatamente inferior, hasta que se alcanza a la capa más baja. Debajo de la capa 1 se encuentra el medio físico a través del cual ocurre la comunicación real.

Entre cada par de capas adyacentes hay una **interfaz**. Ésta define las operaciones y servicios primitivos que pone la capa más baja a disposición de la capa superior inmediata. Cuando los diseñadores de redes deciden cuántas capas incluir en una red y qué debe hacer cada una, la consideración más importante es definir interfaces limpias entre las capas. Al hacer esto es necesario que la capa desempeñe un conjunto específico de funciones bien entendidas. Además de minimizar la cantidad de información que se debe pasar entre las capas, las interfaces bien definidas también simplifican el reemplazo de una capa con un protocolo o implementación totalmente diferente (por ejemplo, reemplazar todas las líneas telefónicas por canales de satélite), ya que todo lo que se requiere del nuevo protocolo o implementación es que ofrezca exactamente el mismo conjunto de servicios a su vecino de arriba, como lo hacía el protocolo o la implementación anterior. Es común que distintos hosts utilicen diferentes implementaciones del mismo protocolo (a menudo escrito por otras compañías). De hecho, el protocolo en sí puede cambiar en cierta capa sin que las capas superior e inferior lo noten.

Como ya se mencionó, a un conjunto de capas y protocolos se le conoce como **arquitectura de red** o **arquitectura de protocolos**. La especificación de una arquitectura debe contener suficiente información como para permitir que un programador escriba el programa o construya el hardware para cada capa, de manera que se cumpla correctamente el protocolo apropiado. Ni los detalles de la implementación ni la especificación de las interfaces forman parte de la arquitectura, ya que están ocultas dentro de las máquinas y no se pueden ver desde el exterior. Ni siquiera es necesario que las interfaces en todas las máquinas de una red sean iguales, siempre y cuando cada máquina pueda utilizar todos los protocolos correctamente. La lista de los protocolos utilizados por cierto sistema, un protocolo por capa, se le conoce como **pila de protocolos** (*stack* de protocolos).

Aspectos de diseño para las capas

Algunos de los aspectos clave de diseño que ocurren en las redes de computadoras están presentes en las diversas capas. A continuación mencionaremos brevemente los más importantes.

La confiabilidad es el aspecto de diseño enfocado en verificar que una red opere correctamente, aun cuando esté formada por una colección de componentes que sean, por sí mismos, poco confiables. Piense en los paquetes que viajan a través de la red. Existe la posibilidad de que algunos segmentos de estos se reciban dañados (invertidos) debido al ruido eléctrico, a las señales aleatorias inalámbricas, a fallas en el hardware, a errores del software, etc. ¿Cómo es posible detectar y corregir estos errores?

Un mecanismo para detectar errores en la información recibida utiliza códigos de **detección de errores**. Así, la información que se recibe de manera incorrecta puede retransmitirse hasta que se reciba de manera correcta. Los códigos más poderosos cuentan con **corrección de errores**, en donde el mensaje correcto se recupera a partir de los segmento posiblemente incorrectos que se recibieron originalmente. Ambos mecanismos funcionan añadiendo información redundante. Se utilizan en capas bajas para proteger los paquetes que se envían a través de enlaces individuales, y en capas altas para verificar que el contenido correcto fue recibido.

Otro aspecto de la confiabilidad consiste en encontrar una ruta funcional a través de una red. A menudo hay múltiples rutas entre origen y destino, y en una red extensa puede haber algunos enlaces o enrutadores descompuestos. Suponga que la red está caída en Alemania. Los paquetes que se envían de Londres a Roma a través de Alemania no podrán pasar, pero para evitar esto,

podríamos enviar los paquetes de Londres a Roma vía París. La red debería tomar esta decisión de manera automática. A este tema se le conoce como **enrutamiento**.

Un segundo aspecto de diseño se refiere a la evolución de la red. Con el tiempo, las redes aumentan su tamaño y emergen nuevos diseños que necesitan conectarse a la red existente. Recientemente vimos el mecanismo de estructuración clave que se utiliza para soportar el cambio dividiendo el problema general y ocultando los detalles de la implementación: distribución de **protocolos en capas**. También existen muchas otras estrategias.

Como hay muchas computadoras en la red, cada capa necesita un mecanismo para identificar los emisores y receptores involucrados en un mensaje específico. Este mecanismo se conoce como **direccionamiento** o **nombramiento** en las capas altas y bajas, respectivamente.

Un aspecto del crecimiento es que las distintas tecnologías de red a menudo tienen diferentes limitaciones. Por ejemplo, no todos los canales de comunicación preservan el orden de los mensajes que se envían en ellos, por lo cual es necesario idear soluciones para enumerar los mensajes. Otro ejemplo es el de las diferencias en el tamaño máximo de un mensaje que las redes pueden transmitir. Esto provoca la creación de mecanismos para desensamblar, transmitir y después volver a ensamblar los mensajes. A este tema en general se le conoce como **interconexión de redes** (internetworking).

Cuando las redes crecen, surgen nuevos problemas. Las ciudades pueden tener problemas de tráfico, escasez de números telefónicos y es fácil perderse. No muchas personas tienen estos problemas en su propio vecindario, pero en toda la ciudad pueden representar un gran problema. Se dice que los diseños que siguen funcionando bien cuando la red aumenta su tamaño son **escalables**.

Un tercer aspecto de diseño radica en la asignación de recursos. Las redes proveen un servicio a los hosts desde sus recursos subyacentes, como la capacidad de las líneas de transmisión. Para hacer bien su trabajo necesitan mecanismos que dividan sus recursos de manera que un host no interfiera demasiado con otro host.

Un problema de asignación que ocurre en todas las capas es cómo evitar que un emisor rápido inunde de datos a un receptor lento. Con frecuencia se utiliza retroalimentación del receptor al emisor. A este tema se le denomina **control de flujo**. Algunas veces el problema es que la red sufre un exceso de solicitudes debido a que hay demasiadas computadoras que desean enviar una gran cantidad de información y la red no lo puede entregar todo. A esta sobrecarga de la red se le conoce como **congestión**. Una estrategia es que cada computadora reduzca su demanda cuando experimenta congestión. Esto también se puede usar en todas las capas.

Es interesante observar que la red puede ofrecer más recursos que simplemente el ancho de banda. Para usos como transmitir video en vivo, la puntualidad de la entrega es en extremo importante. La mayoría de las redes deben proveer servicio a las aplicaciones que desean esta entrega en tiempo real al mismo tiempo que proveen servicio a las aplicaciones que desean un alto rendimiento. La **calidad del servicio** es el nombre que se da a los mecanismos que reconcilian estas demandas competitivas.

El último aspecto de diseño importante es asegurar la red y defenderla contra distintos tipos de amenazas. Una de las amenazas que mencionamos antes es la de espiar las comunicaciones. Los mecanismos que proveen **confidencialidad** nos defienden contra esta amenaza y se utilizan en múltiples capas. Los mecanismos de **autenticación** evitan que alguien se haga pasar por otra persona. Se pueden usar para diferenciar los sitios web bancarios falsos de los verdaderos, o para permitir que la red celular verifique que una llamada realmente provenga de nuestro teléfono para pagar la cuenta.

Comparación entre servicio orientado a conexión y servicio sin conexión

Las capas pueden ofrecer dos tipos distintos de servicio a las capas superiores: orientado a conexión y sin conexión. En esta sección analizaremos estos dos tipos y examinaremos las diferencias entre ellos.

El servicio **orientado a conexión** está modelado a partir del sistema telefónico. Para hablar con alguien levantamos el auricular, marcamos el número, hablamos y después colgamos. De manera similar, para usar un servicio de red orientado a conexión, el usuario del servicio establece primero una conexión, la utiliza y después la libera. El aspecto esencial de una conexión es que funciona como un tubo: el emisor mete objetos (bits) en un extremo y el receptor los toma en el otro extremo. En la mayoría de los casos se conserva el orden de manera que los bits llegan en el orden en el que se enviaron.

En algunos casos al establecer una conexión, el emisor, el receptor y la subred llevan a cabo una negociación en cuanto a los parámetros que se van a usar, como el tamaño máximo del mensaje, la calidad requerida del servicio y demás cuestiones relacionadas. Por lo general, uno de los lados hace una propuesta y el otro puede aceptarla, rechazarla o elaborar una contrapropuesta. Un circuito es otro nombre para una conexión con recursos asociados, como un ancho de banda fijo. Esto se remonta a la red telefónica, en la cual un circuito era una ruta sobre alambre que transmitía una conversación telefónica.

En contraste al servicio orientado a la conexión, el servicio **sin conexión** está modelado a partir del sistema postal. Cada mensaje (carta) lleva la dirección de destino completa, y cada uno es enrutado hacia los nodos intermedios dentro del sistema, en forma independiente a todos los mensajes subsecuentes. Hay distintos nombres para los mensajes en diferentes contextos: un **paquete** es un mensaje en la capa de red. Cuando los nodos intermedios reciben un mensaje completo antes de enviarlo al siguiente nodo, se le llama **conmutación de almacenamiento y envío**. La alternativa en donde la transmisión subsiguiente de un mensaje en un nodo empieza antes de que éste la reciba por completo, se conoce como “conmutación al vuelo”. Por lo general, cuando se envían dos mensajes al mismo destino, el primero que se envíe será el primero en llegar. Sin embargo, es posible que el primero que se envié se retrase de manera que el segundo llegue primero.

Cada tipo de servicio se puede caracterizar con base en su confiabilidad. Algunos servicios son confiables en cuanto a que nunca pierden datos. Por lo general, para implementar un servicio confiable, el receptor tiene que confirmar la recepción de cada mensaje, de manera que el emisor esté seguro de que hayan llegado. El proceso de confirmación de recepción introduce sobrecarga y retardos, que a menudo valen la pena pero algunas veces no son deseables.

Una situación común en la que es apropiado un servicio confiable orientado a la conexión es la transferencia de archivos. El propietario del archivo desea estar seguro de que todos los bits lleguen correctamente y en el mismo orden en el que se enviaron. Muy pocos clientes que transfieren archivos preferirían un servicio que ocasionalmente revuelva o pierda unos cuantos bits, incluso aunque fuera mucho más rápido.

En algunas aplicaciones, los retardos de tránsito ocasionados por las confirmaciones de recepción son inaceptables. Una de estas aplicaciones es el tráfico de *voz digitalizada* o *voz sobre IP*. Es preferible para los usuarios del teléfono escuchar un poco de ruido en la línea de vez en cuando que experimentar un retardo al esperar las confirmaciones de recepción. De manera similar, al transmitir una conferencia de video no hay problema si unos cuantos píxeles están mal, pero es molesto cuando la imagen se sacude mientras el flujo se detiene y avanza para corregir errores.

No todas las aplicaciones requieren conexiones. Por ejemplo, los emisores de correo electrónico basura (spammers) envían su correo a muchos destinatarios. Es probable que el emisor no quiera tener que pasar por el problema de establecer y desmantelar una conexión con un destinatario sólo para enviarle un mensaje. Tampoco es esencial una entrega cien por ciento confiable, sobre todo si eso es más costoso. Todo lo que se requiere es una forma de enviar un solo mensaje que tenga una muy alta probabilidad de llegar, aunque sin garantías. Al servicio sin conexión no confiable (que significa sin confirmación de recepción) se le denomina **servicio de datagramas**, en analogía al servicio de telegramas que tampoco devuelve una confirmación de recepción al emisor. A pesar de ser poco confiable, es la forma más dominante en la mayoría de las redes por motivos que veremos más adelante.

En otros casos es conveniente no tener que establecer una conexión para enviar un mensaje, pero la confiabilidad es esencial. En estas aplicaciones se puede utilizar el servicio de **datagramas con confirmación de recepción**. Es como enviar una carta certificada y solicitar una confirmación de recepción. Al regresar la confirmación de recepción el emisor tiene la absoluta certeza de que la carta se entregó al destinatario correcto y que no se perdió en el camino. La mensajería de texto en los teléfonos móviles es un ejemplo.

Hay otro servicio conocido como servicio de **solicitud-respuesta**. En este servicio el emisor transmite un solo datagrama que contiene una solicitud; al receptor envía la respuesta. El servicio de solicitud-respuesta se utiliza mucho para implementar la comunicación en el modelo cliente-servidor; el cliente emite una petición y el servidor le responde. Por ejemplo, el cliente de un teléfono móvil podría enviar una consulta a un servidor de mapas para recuperar los datos del mapa de la ubicación actual. En la Ilustración 6 se sintetizan los tipos de servicios antes descritos.

	Servicio	Ejemplo
Orientado a conexión	Flujo de mensajes confiable.	Secuencia de páginas.
	Flujo de bytes confiable.	Descarga de películas.
	Conexión no confiable.	Voz sobre IP.
Sin conexión	Datagrama no confiable.	Correo electrónico basura.
	Datagrama confirmación de recepción.	Mensajería de texto.
	Solicitud-respuesta.	Consulta en una base de datos.

Ilustración 2 - Seis tipos distintos de servicios

Tal vez el concepto de usar una comunicación poco confiable le parezca confuso en un principio. Después de todo, ¿por qué preferiría alguien una comunicación poco confiable en vez de una comunicación confiable? Primero que nada, tal vez la comunicación confiable (en nuestro contexto significa que es con confirmación de recepción) no esté disponible en cierta capa. Por ejemplo, Ethernet no provee una comunicación confiable. Los paquetes se pueden dañar ocasionalmente durante el tránsito. Las capas de protocolos más altas deben tener la capacidad de recuperarse de este problema. En particular, muchos servicios confiables se basan en un servicio de datagramas no confiables. En segundo lugar, los retardos inherentes al proveer un servicio confiable tal vez sean inaceptables, en especial en las aplicaciones de tiempo real como multimedia. Éstas son las razones por las que coexisten la comunicación confiable y la comunicación poco confiable.

Primitivas de servicios

Un servicio se puede especificar de manera formal como un conjunto de **primitivas** (operaciones) disponibles a los procesos de usuario para que accedan al servicio. Estas primitivas le indican al servicio que desarrollen alguna acción o que informen sobre la acción que haya tomado una entidad par. Si la pila de protocolos se encuentra en el sistema operativo, como se da en la mayoría de los casos, por lo general las primitivas son llamadas al sistema. Estas llamadas provocan un salto al modo de kernel, que a su vez devuelve el control de la máquina al sistema operativo para que envíe los paquetes necesarios.

El conjunto de primitivas disponibles depende de la naturaleza del servicio que se va a ofrecer. Las primitivas para el servicio orientado a conexión son distintas de las primitivas para el servicio sin conexión.

La relación entre servicios y protocolos

Los servicios y los protocolos son conceptos distintos. Esta distinción es tan importante que la enfatizaremos una vez más. Un *servicio* es un conjunto de primitivas (operaciones) que una capa proporciona a la capa que está encima de ella. El servicio define qué operaciones puede realizar la capa en beneficio de sus usuarios, pero no dice nada sobre cómo se implementan estas operaciones. Un servicio se relaciona con una interfaz entre dos capas, en donde la capa inferior es el proveedor del servicio y la capa superior es el usuario.

En contraste, un *protocolo* es un conjunto de reglas que rigen el formato y el significado de los paquetes o mensajes que intercambian las entidades iguales en una capa. Las entidades utilizan protocolos para implementar sus definiciones de servicios. Pueden cambiar sus protocolos a voluntad, siempre y cuando no cambien el servicio visible para sus usuarios. De esta manera, el servicio y el protocolo no dependen uno del otro. Éste es un concepto clave que cualquier diseñador de red debe comprender bien.

Vale la pena mencionar una analogía con los lenguajes de programación. Un servicio es como un tipo de datos abstracto o un objeto en un lenguaje orientado a objetos. Define las operaciones que se pueden realizar en un objeto, pero no especifica cómo se implementan estas operaciones. En contraste, un protocolo se relaciona con la implementación del servicio y como tal, no es visible al usuario del mismo.

Muchos protocolos antiguos no diferenciaban el servicio del protocolo. En efecto, una capa típica podría tener una primitiva de servicio *SEND PACKET* en donde el usuario proporcionaba un apuntador hacia un paquete completamente ensamblado. Este arreglo significaba que los usuarios podían ver de inmediato todos los cambios en el protocolo. Ahora, la mayoría de los diseñadores de redes consideran dicho diseño como un error garrafal.

Arquitectura de protocolos normalizadas

Cuando se desea establecer una comunicación entre computadoras de diferentes fabricantes, el desarrollo del software puede convertirse en una pesadilla. Los distintos fabricantes pueden hacer uso de distintos formatos y protocolos de intercambio de datos. Incluso dentro de una misma línea de productos de un fabricante dado, los diferentes modelos pueden comunicarse de forma diferente.

Los estándares son necesarios para promover la interoperatividad entre los equipos de distintos fabricantes, así como para facilitar economías de gran escala. Debido a la complejidad que implican las comunicaciones, un solo estándar no es suficiente. En su lugar, las distintas funcionalidades deberían dividirse en partes más manejables, estructurándose en una arquitectura de protocolos. La arquitectura constituirá, por tanto, el marco de trabajo para el proceso de normalización.

Hay dos arquitecturas que han sido determinantes y básicas en el desarrollo de los estándares de comunicación: **el conjunto de protocolos TCP/IP** y el **modelo de referencia de OSI**. TCP/IP es, con diferencia, la arquitectura más usada. OSI, aun siendo bien conocida, nunca ha llegado a alcanzar las promesas iniciales.

Por su parte la Organización Internacional de Estandarización (ISO, Internacional *Organization for Standarization*) estableció en 1977 un subcomité para el desarrollo de una arquitectura de protocolos. El resultado fue el modelo de referencia OSI.

Por otro lado la arquitectura de protocolos TCP/IP es resultado de la investigación y desarrollo llevados a cabo en la red experimental de conmutación de paquetes *ARPANET*, financiada por la Agencia de Proyectos de Investigación Avanzada para la Defensa (DARPA, *Defense Advanced Research Projects Agency*), y se denomina globalmente como la familia de protocolos TCP/IP. Esta familia consiste en una extensa colección de protocolos que se han especificado como estándares de Internet por parte de IAB (*Internet Architecture Board*).

El modelo de referencia OSI

El modelo OSI se muestra en la Ilustración 7. Este modelo se basa en una propuesta desarrollada por la ISO como el primer paso hacia la estandarización internacional de los protocolos utilizados en las diversas capas (*Day y Zimmerman*, 1983). Este modelo se revisó en 1995 (*Day*, 1995) y se le llama Modelo de referencia OSI (Interconexión de Sistemas Abiertos, del inglés *Open Systems Interconnection*) de la ISO puesto que se ocupa de la conexión de sistemas abiertos; esto es, sistemas que están abiertos a la comunicación con otros sistemas. Para abreviar, lo llamaremos modelo OSI.

El modelo OSI tiene siete capas. Los principios que se aplicaron para llegar a las siete capas se pueden resumir de la siguiente manera:

1. Se debe crear una capa en donde se requiera un nivel diferente de abstracción.
2. Cada capa debe realizar una función bien definida.
3. La función de cada capa se debe elegir teniendo en cuenta la definición de protocolos estandarizados internacionalmente.
4. Es necesario elegir los límites de las capas de modo que se minimice el flujo de información a través de las interfaces.
5. La cantidad de capas debe ser suficiente como para no tener que agrupar funciones distintas en la misma capa; además, debe ser lo bastante pequeña como para que la arquitectura no se vuelva inmanejable.

Tenga en cuenta que el modelo OSI en sí no es una arquitectura de protocolos, ya que no especifica los servicios y protocolos exactos que se van a utilizar en cada capa. Sólo indica lo que una debe hacer. Sin embargo, la ISO también ha elaborado estándares para todas las capas, aunque no son parte del modelo de referencia en sí. Cada uno se publicó como un estándar internacional separado. Aunque el modelo (en parte) es muy usado, los protocolos asociados han estado en el olvido desde hace tiempo.

La capa física

La capa física se relaciona con la transmisión de bits puros a través de un canal de transmisión. Los aspectos de diseño tienen que ver con la acción de asegurarse que cuando uno de los lados envíe un bit 1 el otro lado lo reciba como un bit 1, no como un bit 0. En este caso las preguntas típicas son: ¿qué señales eléctricas se deben usar para representar un 1 y un 0?, ¿cuántos nanosegundos dura un bit?, ¿la transmisión puede proceder de manera simultánea en ambas direcciones?, ¿cómo se establece la conexión inicial y cómo se interrumpe cuando ambos lados han terminado?, ¿cuántos pines tiene el conector de red y para qué sirve cada uno? Los aspectos

de diseño tienen que ver con las interfaces mecánica, eléctrica y de temporización, así como con el medio de transmisión físico que se encuentra bajo la capa física.

La capa de enlace de datos

La principal tarea de la capa de enlace de datos es transformar un medio de transmisión puro en una línea que esté libre de errores de transmisión. Enmascara los errores reales, de manera que la capa de red no los vea. Para lograr esta tarea, el emisor divide los datos de entrada en tramas de datos (por lo general, de algunos cientos o miles de bytes) y transmite las tramas en forma secuencial. Si el servicio es confiable, para confirmar la recepción correcta de cada trama, el receptor devuelve una trama de confirmación de recepción.

Otra cuestión que surge en la capa de enlace de datos (y en la mayoría de las capas superiores) es cómo evitar que un transmisor rápido inunde de datos a un receptor lento. Tal vez sea necesario algún mecanismo de regulación de tráfico para notificar al transmisor cuando el receptor puede aceptar más datos.

Las redes de difusión tienen una consideración adicional en la capa de enlace de datos: cómo controlar el acceso al canal compartido. Una subcapa especial de la capa de enlace de datos, conocida como subcapa de control de acceso al medio, es la que se encarga de este problema.

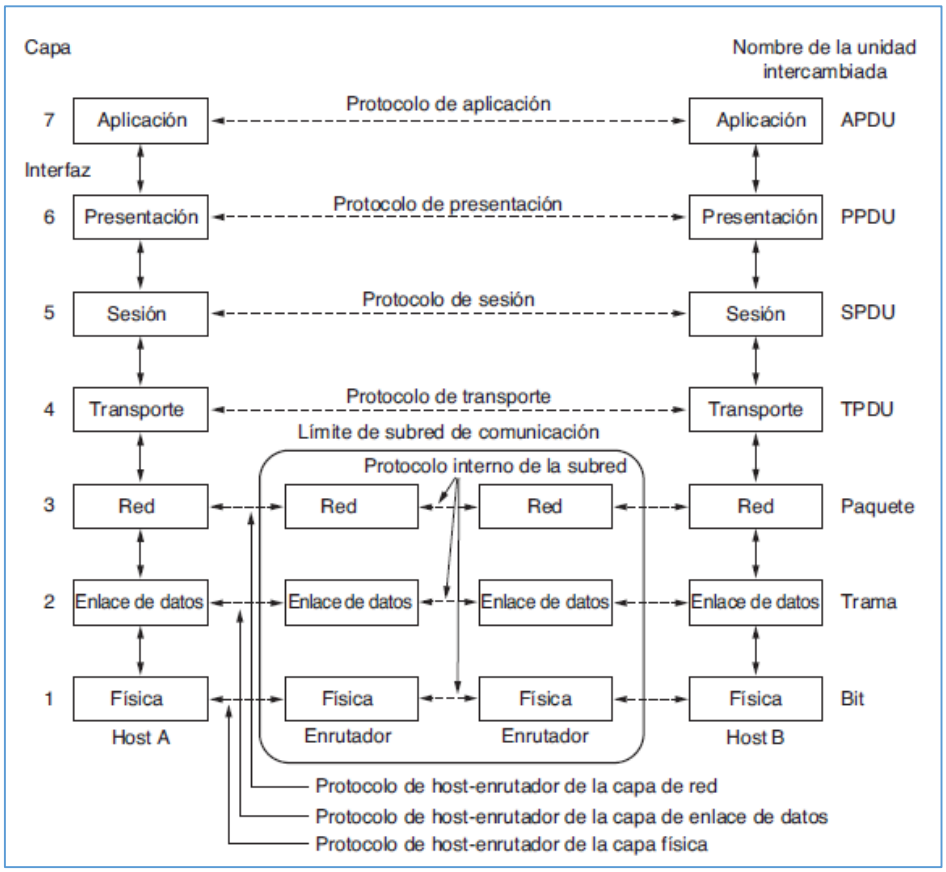


Ilustración 3 - Modelo de referencia OSI

La capa de red

La capa de red controla la operación de la subred. Una cuestión clave de diseño es determinar cómo se encaminan los paquetes desde el origen hasta el destino. Las rutas se pueden basar en tablas estáticas que se “codifican” en la red y rara vez cambian, aunque es más común que se actualicen de manera automática para evitar las fallas en los componentes. También se pueden determinar el inicio de cada conversación; por ejemplo, en una sesión de terminal al iniciar

sesión en una máquina remota. Por último, pueden ser muy dinámicas y determinarse de nuevo para cada paquete, de manera que se pueda reflejar la carga actual en la red.

Si hay demasiados paquetes en la subred al mismo tiempo, se interpondrán en el camino unos con otros y formarán cuellos de botella. El manejo de la congestión también es responsabilidad de la capa de red, en conjunto con las capas superiores que adaptan la carga que colocan en la red. Otra cuestión más general de la capa de red es la calidad del servicio proporcionado (retardo, tiempo de tránsito, variaciones, etcétera).

Cuando un paquete tiene que viajar de una red a otra para llegar a su destino, pueden surgir muchos problemas. El direccionamiento utilizado por la segunda red puede ser distinto del que utiliza la primera. La segunda red tal vez no acepte el paquete debido a que es demasiado grande. Los protocolos pueden ser diferentes, etc. Es responsabilidad de la capa de red solucionar todos estos problemas para permitir la interconexión de redes heterogéneas.

En las redes de difusión, el problema de encaminamiento es simple, por lo que con frecuencia la capa de red es delgada o incluso inexistente.

La capa de transporte

La función básica de la capa de transporte es aceptar datos de la capa superior, dividirlos en unidades más pequeñas si es necesario, pasar estos datos a la capa de red y asegurar que todas las piezas lleguen correctamente al otro extremo. Además, todo esto se debe realizar con eficiencia y de una manera que aísle las capas superiores de los inevitables cambios en la tecnología de hardware que se dan con el transcurso del tiempo.

La capa de transporte también determina el tipo de servicio que debe proveer a la capa de sesión y, en última instancia, a los usuarios de la red. El tipo más popular de conexión de transporte es un canal punto a punto libre de errores que entrega los mensajes o bytes en el orden en el que se enviaron. Sin embargo existen otros posibles tipos de servicio de transporte, como el de mensajes aislados sin garantía sobre el orden de la entrega y la difusión de mensajes a múltiples destinos. El tipo de servicio se determina al establecer la conexión (cabe mencionar que es imposible lograr un canal libre de errores; lo que se quiere decir en realidad con este término es que la tasa de errores es lo bastante baja como para ignorarla en la práctica).

La capa de transporte es una verdadera capa de extremo a extremo; lleva los datos por toda la ruta desde el origen hasta el destino. En otras palabras, un programa en la máquina de origen lleva a cabo una conversación con un programa similar en la máquina de destino mediante el uso de los encabezados en los mensajes y los mensajes de control. En las capas inferiores cada uno de los protocolos está entre una máquina y sus vecinos inmediatos, no entre las verdaderas máquinas de origen y de destino, que pueden estar separadas por muchos enrutadores. En la Ilustración 7 se muestra la diferencia entre las capas de la 1 a la 3, que están encadenadas, y entre las capas de la 4 a la 7, que son de extremo a extremo.

La capa de sesión

La capa de sesión permite a los usuarios en distintas máquinas establecer sesiones entre ellos. Las sesiones ofrecen varios servicios, incluyendo el control del diálogo (llevar el control de quién va a transmitir), el manejo de tokens (evitar que dos partes intenten la misma operación crítica al mismo tiempo) y la sincronización (usar puntos de referencia en las transmisiones extensas para reanudar desde el último punto de referencia en caso de una interrupción).

La capa de presentación

A diferencia de las capas inferiores, que se enfocan principalmente en mover los bits de un lado a otro, la capa de presentación se enfoca en la sintaxis y la semántica de la información transmitida. Para hacer posible la comunicación entre computadoras con distintas

representaciones internas de datos, podemos definir de una manera abstracta las estructuras de datos que se van a intercambiar, junto con una codificación estándar que se use "en el cable". La capa de presentación maneja estas estructuras de datos abstractas y permite definir e intercambiar estructuras de datos de mayor nivel (por ejemplo, registros bancarios).

La capa de aplicación

La capa de aplicación contiene una variedad de protocolos que los usuarios necesitan con frecuencia. Un protocolo de aplicación muy utilizado es HTTP (Protocolo de Transferencia de Hipertexto, del inglés *HyperText Transfer Protocol*), el cual forma la base para la World Wide Web. Cuando un navegador desea una página web, envía el nombre de la página que quiere al servidor que la hospeda mediante el uso de HTTP. Después el servidor envía la página de vuelta. Hay otros protocolos de aplicación que se utilizan para transferir archivos, enviar y recibir correo electrónico y noticias.

El modelo de referencia TCP/IP

Pasemos ahora del modelo de referencia OSI al modelo de referencia que se utiliza en la más vieja de todas las redes de computadoras de área amplia: ARPANET y su sucesora, Internet. ARPANET era una red de investigación patrocinada por el DoD (Departamento de Defensa de Estados Unidos, del inglés U.S. *Department of the Defense*). En un momento dado llegó a conectar cientos de universidades e instalaciones gubernamentales mediante el uso de líneas telefónicas rentadas. Cuando después se le unieron las redes de satélites y de radio, los protocolos existentes tuvieron problemas para interactuar con ellas, de modo que se necesitaba una nueva arquitectura de referencia. Así, casi desde el principio la habilidad de conectar varias redes sin problemas fue uno de los principales objetivos de diseño. Posteriormente esta arquitectura se dio a conocer como el Modelo de referencia TCP/IP, debido a sus dos protocolos primarios. Este modelo se definió por primera vez en Cerf y Kahn (1974); después se refinó y definió como estándar en la comunidad de Internet (*Braden*, 1989). Clark (1988) describe la filosofía de diseño detrás de este modelo.

Debido a la preocupación del DoD de que alguno de sus valiosos hosts, enrutadores y puertas de enlace de interredes pudieran ser volados en pedazos en cualquier momento por un ataque de la antigua Unión Soviética, otro de los objetivos principales fue que la red pudiera sobrevivir a la pérdida de hardware de la subred sin que se interrumpieran las conversaciones existentes. En otras palabras, el DoD quería que las conexiones permanecieran intactas mientras las máquinas de origen y de destino estuvieran funcionando, incluso aunque algunas de las máquinas o líneas de transmisión en el trayecto dejaran de funcionar en forma repentina. Además, como se tenían en mente aplicaciones con requerimientos divergentes que abarcaban desde la transferencia de archivos hasta la transmisión de voz en tiempo real, se necesitaba una arquitectura flexible.

La capa de enlace o acceso a la red

Todos estos requerimientos condujeron a la elección de una red de conmutación de paquetes basada en una capa sin conexión que opera a través de distintas redes. La capa más baja en este modelo es la capa de enlace; ésta describe qué enlaces (como las líneas seriales y Ethernet clásica) se deben llevar a cabo para cumplir con las necesidades de esta capa de interred sin conexión. En realidad no es una capa en el sentido común del término, sino una interfaz entre los hosts y los enlaces de transmisión. El primer material sobre el modelo TCP/IP tiene poco que decir sobre ello.

La capa de interred o internet

Esta capa es el eje que mantiene unida a toda la arquitectura. Aparece en la Ilustración 8 con una correspondencia aproximada a la capa de red de OSI. Su trabajo es permitir que los hosts

inyecten paquetes en cualquier red y que viajen de manera independiente hacia el destino (que puede estar en una red distinta). Incluso pueden llegar en un orden totalmente diferente al orden en que se enviaron, en cuyo caso es responsabilidad de las capas más altas volver a ordenarlos, si se desea una entrega en orden. Tenga en cuenta que aquí utilizamos "interred" en un sentido genérico, aunque esta capa esté presente en la Internet.

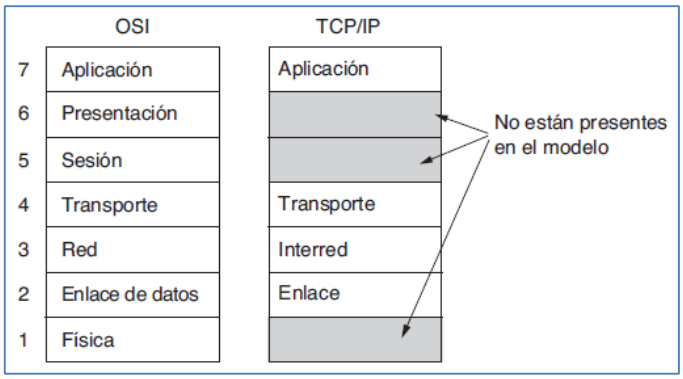


Ilustración 4 - Modelo de referencia TCP/IP

La analogía aquí es con el sistema de correos convencional (lento). Una persona puede dejar una secuencia de cartas internacionales en un buzón en un país y, con un poco de suerte, la mayoría de ellas se entregarán a la dirección correcta en el país de destino. Es probable que las cartas pasen a través de una o más puertas de enlace de correo internacionales en su trayecto, pero esto es transparente a los usuarios. Además, los usuarios no necesitan saber que cada país (es decir, cada red) tiene sus propias estampillas, tamaños de sobre preferidos y reglas de entrega.

La capa de interred define un formato de paquete y un protocolo oficial llamado IP (Protocolo de Internet, del inglés *Internet Protocol*), además de un protocolo complementario llamado ICMP (Protocolo de Mensajes de Control de Internet, del inglés Internet *Control Message Protocol*) que le ayuda a funcionar. La tarea de la capa de interred es entregar los paquetes IP a donde se supone que deben ir. Aquí el ruteo de los paquetes es sin duda el principal aspecto, al igual que la congestión (aunque el IP no ha demostrado ser efectivo para evitar la congestión).

La capa de transporte

Por lo general, a la capa que está arriba de la capa de interred en el modelo TCP/IP se le conoce como capa de transporte; y está diseñada para permitir que las entidades pares, en los nodos de origen y de destino, lleven a cabo una conversación, al igual que en la capa de transporte de OSI. Aquí se definieron dos protocolos de transporte de extremo a extremo. El primero, TCP (Protocolo de Control de la Transmisión, del inglés *Transmission Control Protocol*), es un protocolo confiable orientado a la conexión que permite que un flujo de bytes originado en una máquina se entregue sin errores a cualquier otra máquina en la interred. Este protocolo segmenta el flujo de bytes entrante en mensajes discretos y pasa cada uno a la capa de interred. En el destino, el proceso TCP receptor vuelve a ensamblar los mensajes recibidos para formar el flujo de salida. El TCP también maneja el control de flujo para asegurar que un emisor rápido no pueda inundar a un receptor lento con más mensajes de los que pueda manejar.

El segundo protocolo en esta capa, UDP (Protocolo de Datagrama de Usuario, del inglés *User Datagram Protocol*), es un protocolo sin conexión, no confiable para aplicaciones que no desean la asignación de secuencia o el control de flujo de TCP y prefieren proveerlos por su cuenta. También se utiliza mucho en las consultas de petición-respuesta de una sola ocasión del tipo cliente-servidor, y en las aplicaciones en las que es más importante una entrega oportuna que una entrega precisa, como en la transmisión de voz o video. En la Ilustración 9 se muestra la

relación entre IP, TCP y UDP. Desde que se desarrolló el modelo, el IP se ha implementado en muchas otras redes.

La capa de aplicación

El modelo TCP/IP no tiene capas de sesión o de presentación, ya que no se consideraron necesarias. Las aplicaciones simplemente incluyen cualquier función de sesión y de presentación que requieran. La experiencia con el modelo OSI ha demostrado que esta visión fue correcta: estas capas se utilizan muy poco en la mayoría de las aplicaciones.

Encima de la capa de transporte se encuentra la capa de aplicación. Ésta contiene todos los protocolos de alto nivel. Entre los primeros protocolos están el de terminal virtual (TELNET), transferencia de archivos (FTP) y correo electrónico (SMTP). A través de los años se han agregado muchos otros protocolos. En la Ilustración 9 se muestran algunos de los más importantes que veremos más adelante: el Sistema de nombres de dominio (DNS) para resolución de nombres de hosts a sus direcciones de red; HTTP, el protocolo para recuperar páginas de la World Wide Web; y RTP, el protocolo para transmitir medios en tiempo real, como voz o películas.

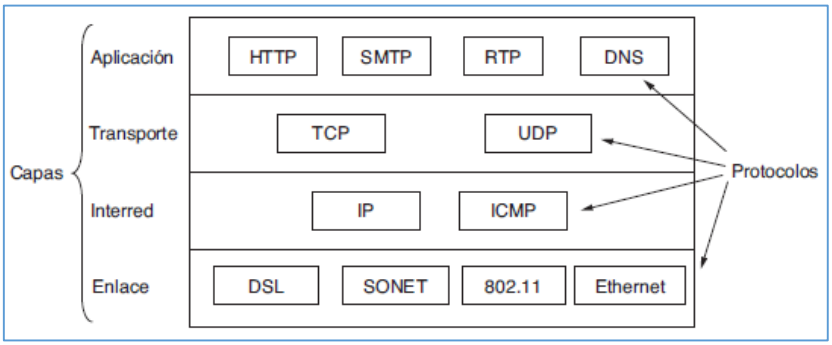


Ilustración 5 - Modelo TCP/IP con algunos de los protocolos

Comparación de los modelos de referencia OSI y TCP/IP

Los modelos de referencia OSI y TCP/IP tienen mucho en común. Ambos se basan en el concepto de una pila de protocolos independientes. Además, la funcionalidad de las capas es muy similar. Por ejemplo, en ambos modelos las capas por encima de la de transporte, incluyendo ésta, se encuentran ahí para proporcionar un servicio de transporte independiente de la red, de extremo a extremo, para los procesos que desean comunicarse. Estas capas forman el proveedor de transporte. También en ambos modelos, las capas que están arriba de la de transporte son usuarios orientadas a la aplicación del servicio de transporte.

A pesar de estas similitudes fundamentales, los dos modelos también tienen muchas diferencias. En esta sección nos enfocaremos en las diferencias clave entre los dos modelos de referencia. Es importante tener en cuenta que aquí compararemos los modelos de referencia y no las pilas de protocolos correspondientes. Más adelante estudiaremos los protocolos en sí. Un libro completo dedicado a comparar y contrastar TCP/IP y OSI es el de Piscitello y Chapin (1993).

Hay tres conceptos básicos para el modelo OSI:

1. Servicios.
2. Interfaces.
3. Protocolos.

Quizá, la mayor contribución del modelo OSI es que hace explícita la distinción entre estos tres conceptos. Cada capa desempeña ciertos servicios para la capa que está sobre ella. La definición del servicio indica lo que hace la capa, no cómo acceden a ella las entidades superiores ni cómo funciona. Define la semántica de la capa.

La interfaz de una capa indica a los procesos superiores cómo pueden acceder a ella. Especifica cuáles son los parámetros y qué resultados se pueden esperar. Pero no dice nada sobre su funcionamiento interno.

Por último, la capa es la que debe decidir qué protocolos de iguales utilizar. Puede usar los protocolos que quiera, siempre y cuando realice el trabajo (es decir, que provea los servicios ofrecidos). También los puede cambiar a voluntad sin afectar el software de las capas superiores.

Estas ideas encajan muy bien con las ideas modernas sobre la programación orientada a objetos. Al igual que una capa, un objeto tiene un conjunto de métodos (operaciones) que los procesos fuera del objeto pueden invocar. La semántica de estos métodos define el conjunto de servicios que ofrece el objeto. Los parámetros y resultados de los métodos forman la interfaz del objeto. El código interno del objeto es su protocolo y no se puede ver ni es de la incumbencia de las entidades externas al objeto.

Al principio, el modelo TCP/IP no tenía una distinción clara entre los servicios, las interfaces y los protocolos, aunque las personas han tratado de reajustarlo a fin de hacerlo más parecido al OSI. Por ejemplo, los únicos servicios que realmente ofrece la capa de interred son *send ip packet* y *receive ip packet*. Como consecuencia, los protocolos en el modelo OSI están ocultos de una mejor forma que en el modelo TCP/IP, además se pueden reemplazar con relativa facilidad a medida que la tecnología cambia. La capacidad de realizar dichos cambios con transparencia es uno de los principales propósitos de tener protocolos en capas en primer lugar.

El modelo de referencia OSI se ideó antes de que se inventaran los protocolos correspondientes. Este orden significa que el modelo no estaba orientado hacia un conjunto específico de protocolos, un hecho que lo hizo bastante general. La desventaja de este orden fue que los diseñadores no tenían mucha experiencia con el tema y no supieron bien qué funcionalidad debían colocar en cada una de las capas.

Por ejemplo, en un principio la capa de enlace de datos trabajaba sólo con redes de punto a punto. Cuando surgieron las redes de difusión, fue necesario insertar una nueva subcapa al modelo. Además, cuando las personas empezaron a construir redes reales mediante el modelo OSI y los protocolos existentes, se descubrió que estas redes no coincidían con las especificaciones de los servicios requeridos, de modo que tuvieron que integrar en el modelo subcapas convergentes que permitieran cubrir las diferencias. Finalmente, el comité en un principio esperaba que cada país tuviera una red operada por el gobierno en la que se utilizaran los protocolos OSI, por lo que no se tomó en cuenta la interconexión de redes. Para no hacer el cuento largo, las cosas no salieron como se esperaba.

Con TCP/IP sucedió lo contrario: primero llegaron los protocolos y el modelo era en realidad sólo una descripción de los protocolos existentes. No hubo problema para que los protocolos se ajustaran al modelo. Encajaron a la perfección. El único problema fue que el modelo no encajaba en ninguna otra pila de protocolos. En consecuencia, no era útil para describir otras redes que no fueran TCP/IP.

Pasando de las cuestiones filosóficas a las más específicas, una diferencia obvia entre los dos modelos está en el número de capas: el modelo OSI tiene siete capas, mientras que el modelo TCP/ IP tiene cuatro. Ambos tienen capas de (inter)red, transporte y aplicación, pero las demás capas son distintas.

Hay otra diferencia en el área de la comunicación sin conexión frente a la comunicación orientada a conexión. El modelo OSI soporta ambos tipos de comunicación en la capa de red, pero sólo la comunicación orientada a conexión en la capa de transporte, en donde es más importante (ya que el servicio de transporte es visible a los usuarios). El modelo TCP/IP sólo soporta un modo en la capa de red (sin conexión) pero soporta ambos en la capa de transporte,

de manera que los usuarios tienen una alternativa, que es muy importante para los protocolos simples de petición-respuesta.

El modelo utilizado en la materia

Como dijimos antes, la fortaleza del modelo de referencia OSI es el modelo en sí (excepto las capas de presentación y de sesión), el cual ha demostrado ser excepcionalmente útil para hablar sobre redes de computadoras. En contraste, la fortaleza del modelo de referencia TCP/IP son los protocolos, que se han utilizado mucho durante varios años. Como a los científicos de computadoras les gusta hacer sus propias herramientas, utilizaremos el modelo híbrido de la Ilustración 10 como marco de trabajo.

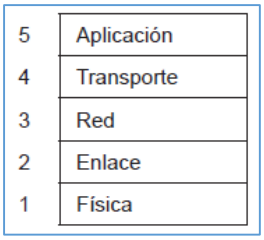


Ilustración 6 - Modelo de referencia híbrido

Este modelo tiene cinco capas, empezando por la capa física, pasando por las capas de enlace, red y transporte hasta llegar a la capa de aplicación. La capa física especifica cómo transmitir bits a través de distintos tipos de medios como señales eléctricas (u otras señales analógicas). La capa de enlace trata sobre cómo enviar mensajes de longitud finita entre computadoras conectadas de manera directa con niveles específicos de confiabilidad. Ethernet y 802.11 son ejemplos de protocolos de capa de enlace.

La capa de red se encarga de combinar varios enlaces múltiples en redes, y redes de redes en interredes, de manera que podamos enviar paquetes entre computadoras distantes. Aquí se incluye la tarea de buscar la ruta por la cual enviarán los paquetes. IP es el principal protocolo de ejemplo que estudiaremos para esta capa. La capa de transporte fortalece las garantías de entrega de la capa de Red, por lo general con una mayor confiabilidad, además provee abstracciones en la entrega, como un flujo de bytes confiable, que coincida con las necesidades de las distintas aplicaciones. TCP es un importante ejemplo de un protocolo de capa de transporte.

Por último, la capa de aplicación contiene programas que hacen uso de la red. Muchas aplicaciones en red tienen interfaces de usuario, como un navegador web. Sin embargo, nuestro interés está en la parte del programa que utiliza la red. En el caso del navegador web se trata del protocolo HTTP. También hay programas de soporte importantes en la capa de aplicación, como el DNS, que muchas aplicaciones utilizan.

El desarrollo de las unidades de esta materia se basa en este modelo. De esta forma, retenemos el valor del modelo OSI para comprender las arquitecturas de red al tiempo que nos concentramos principalmente en los protocolos que son importantes en la práctica, desde TCP/IP y los protocolos relacionados hasta los más recientes como 802.11, SONET y Bluetooth.