Fundamentos de comunicaciones

Los fundamentos de comunicaciones establecen las bases para entender cómo los dispositivos se conectan e interactúan en una red. Desde los modelos de programación cliente/servidor y microservicios hasta las capas físicas, de enlace y de transporte, este capítulo abarca protocolos esenciales como TCP/IP, esquemas de direccionamiento y servicios básicos. También se introducen tecnologías clave como Ethernet y las direcciones físicas.

1. Modelos de programación en red.

Los modelos de programación en red establecen las bases para el diseño de aplicaciones distribuidas que interactúan a través de la red. Este apartado explora el modelo cliente/servidor, los enfoques modernos basados en microservicios y APIs, así como los modelos orientados a mensajes y servicios web, que permiten crear soluciones escalables y flexibles.

A continuación, se presenta una tabla comparativa que detalla características, ejemplos y contextos más adecuados para la implementación de diferentes modelos de programación en red:

Modelo	Características	Ejemplo	Contexto más adecuado basado en su implementación práctica
Cliente/Servidor	Puede incorporar servidores redundantes para mayor disponibilidad y sistemas de caché para acelerar las respuestas.	Una plataforma de aprendizaje online con servidores dedicados para gestionar recursos educativos y usuarios concurrentes.	Sistemas donde los datos deben gestionarse de manera centralizada, como sistemas bancarios, aplicaciones educativas o servicios de streaming básicos.
Microservicios y APIs	Facilita la implementación de despliegues continuos, permitiendo la actualización de componentes individuales sin afectar al sistema.	Una app de delivery donde los microservicios gestionan catálogos, pedidos, pagos y localización en tiempo real.	Escenarios donde se requiere una alta disponibilidad y actualización constante, como marketplaces globales o sistemas IoT de gran escala.
Basados en Mensajes	Ofrecen flexibilidad en sistemas distribuidos gracias al uso de colas persistentes, que permiten procesar tareas sin pérdida de datos.	Un sistema de monitoreo de sensores industriales que recopila datos y los envía para análisis y mantenimiento predictivo.	Procesos que dependen de acciones consecutivas pero descentralizadas, como cadenas de logística o entornos de monitoreo automatizado.

Servicios Web (SOAP y REST)	REST soporta JSON, lo que lo hace más ligero y compatible con aplicaciones móviles, mientras SOAP incluye WS-Security para firmas y cifrado.	Un sistema de salud donde SOAP se usa para gestionar historiales médicos con transacciones seguras y REST para consultas rápidas.	REST es ideal para desarrollos ágiles y flexibles; SOAP se utiliza en sectores con alta regulación, como banca o sistemas médicos críticos.

1.1. El modelo cliente/servidor.

El modelo cliente/servidor es, Probablemente, el enfoque más conocido y utilizado en el desarrollo de aplicaciones en red. En este modelo, un servidor centralizado ofrece servicios o recursos, mientras que los clientes se conectan a él para consumir esos servicios. Es un diseño sencillo y directo que se adapta bien a una amplia variedad de aplicaciones, desde bases de datos hasta páginas web.

Por ejemplo, una tienda online funciona bajo este modelo: el servidor almacena la información sobre los Productos, gestiona las órdenes de compra y mantiene los datos de los usuarios, mientras que el cliente (el navegador del usuario) se conecta al servidor para acceder a esta información.

Una de las ventajas del modelo cliente/servidor es su simplicidad en términos de diseño y desarrollo. Sin embargo, también tiene limitaciones, como la dependencia de un único punto central (el servidor). Si este falla o se satura, toda la aplicación podría verse afectada. Para mitigar este riesgo, suelen implementarse servidores redundantes y balanceadores de carga que distribuyen las solicitudes entre múltiples servidores.

1.2. Modelos de microservicios y APIs modernas.

El modelo de microservicios representa una evolución del modelo cliente/servidor, diseñada para abordar las limitaciones de las aplicaciones monolíticas. En lugar de depender de un único servidor que gestione todas las funciones de la aplicación, los microservicios dividen las tareas en servicios pequeños e independientes, cada uno de los cuales se encarga de una función específica.

Por ejemplo, en una plataforma de streaming como "Cine Lannister", un microservicio podría encargarse exclusivamente de la búsqueda de películas, otro de gestionar las cuentas de los usuarios, y otro de procesar los pagos. Todos estos servicios se comunican entre sí a través de APIs, que actúan como intermediarios para transmitir datos de forma estructurada.

Este modelo ofrece varias ventajas, como una mayor escalabilidad y flexibilidad. Si un microservicio necesita Procesar más solicitudes, puede escalarse de forma independiente sin afectar al resto de la aplicación. Además, facilita el mantenimiento, ya que cada microservicio puede actualizarse o reemplazarse sin necesidad de rediseñar toda la aplicación.

Por supuesto, este enfoque también tiene desafíos, como la complejidad adicional en la gestión de la comunicación entre microservicios. Para ello, se utilizan herramientas específicas, como contenedores (Docker) y plataformas de orquestación (Kubernetes), que simplifican la implementación y el mantenimiento de aplicaciones basadas en microservicios.

1.3. Modelos basados en mensajes. Introducción a los servicios web.

Los modelos basados en mensajes son especialmente útiles en entornos donde los componentes de una aplicación deben comunicarse de manera asíncrona y descentralizada. En este enfoque, los mensajes son paquetes de datos estructurados que se envían de un componente a otro a través de una cola o intermediario.

Por ejemplo, en un sistema de pedidos en línea, cuando un cliente realiza una compra, el servicio de pagos podría enviar un mensaje al servicio de logística para iniciar el envío. Mientras tanto, el servicio de pagos puede continuar Procesando otras transacciones sin esperar a que el servicio de logística responda.

Este modelo es ideal para aplicaciones que necesitan alta disponibilidad y tolerancia a fallos, ya que los mensajes pueden almacenarse en la cola hasta que el receptor esté listo para Procesarlos. Herramientas como RabbitMQ, Kafka o Azure Service Bus son populares para implementar este enfoque.

En cuanto a los servicios web, estos son una implementación práctica de los modelos basados en mensajes. Un servicio web es una interfaz que permite a diferentes aplicaciones comunicarse entre sí, independientemente de las plataformas o lenguajes de programación que utilicen.

- **SOAP (Simple Object Access Protocol)**: un enfoque más estructurado y formal, que utiliza XML para definir mensajes.
- **REST (Representational State Transfer)**: un enfoque más ligero y flexible, basado en estándares web como HTTP y JSON.

REST, Por ejemplo, es ampliamente utilizado para desarrollar APIs que conectan aplicaciones móviles con servicios en la nube.

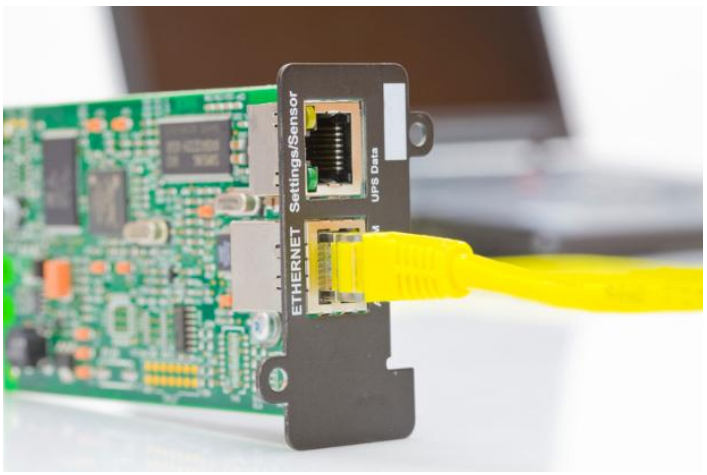
2. El nivel físico.

El nivel físico de una red define cómo se transmiten los datos a través del medio físico. Este apartado aborda los dispositivos físicos que permiten la conectividad y los protocolos asociados, destacando su relevancia en la construcción de infraestructuras de red confiables y eficientes.

2.1. Dispositivos físicos.

Los dispositivos físicos son los elementos que permiten la conexión entre los diferentes nodos de una red. Estos dispositivos varían según el tipo de red y el medio de transmisión, pero todos tienen el mismo objetivo: facilitar la transferencia de datos. Entre los más comunes se encuentran:

- **Tarjetas de red (NIC)**: son componentes esenciales en cualquier dispositivo conectado a una red. Su función es traducir los datos digitales generados Por el software en señales que puedan ser transmitidas a través del medio físico. Las NIC pueden ser para redes cableadas (Ethernet) o inalámbricas (Wi-Fi).
    - Por ejemplo, en un ordenador personal, la tarjeta de red Ethernet permite la conexión a redes locales mediante cables de par trenzado:



- **Concentradores, conmutadores y routers**: aunque estos dispositivos también operan en otros niveles de la pila de red, su función básica incluye la gestión y direccionamiento de señales físicas.
    - Un **conmutador (switch)**, Por ejemplo, utiliza las señales eléctricas para determinar qué puerto debe recibir los datos:

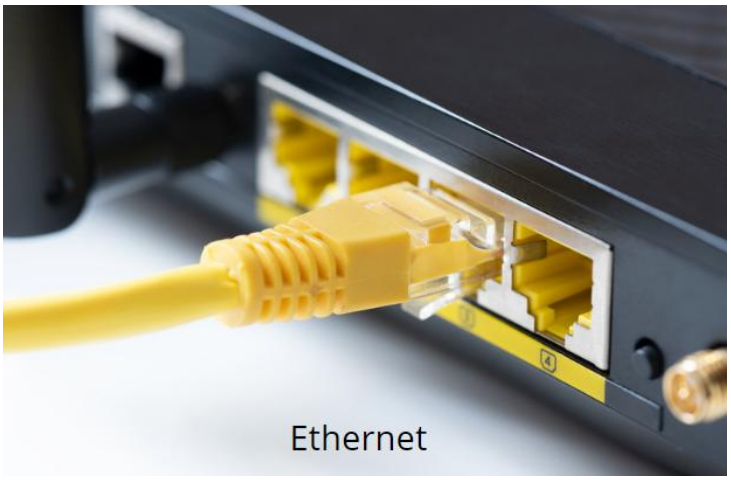
- **Medios de transmisión**: los cables de cobre (como el par trenzado o el coaxial) y la fibra óptica son ejemplos de medios físicos para transportar señales. En redes inalámbricas, las ondas electromagnéticas cumplen esta función.
    - Por ejemplo, un cable de fibra óptica puede transmitir datos a gran velocidad y largas distancias sin apenas pérdida de señal, siendo ideal para redes de alta capacidad:

El diseño y la calidad de los dispositivos físicos tienen un impacto directo en el rendimiento de la red. Factores como la atenuación (pérdida de intensidad de la señal) y las interferencias externas deben ser considerados al seleccionar los componentes.

2.2. Protocolos de nivel físico.

Los protocolos de nivel físico definen cómo se transmiten y reciben las señales en el medio físico. Establecen las reglas para aspectos como la modulación de la señal, la sincronización y la codificación de los datos. Algunos de los protocolos más destacados son:

- **Ethernet**: es uno de los estándares más utilizados en redes locales (LAN). Define cómo deben ser transmitidos los datos en medios cableados, como cables de par trenzado o fibra óptica.

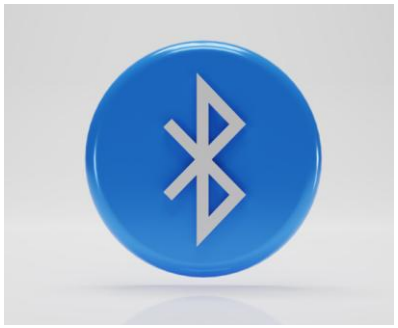


    - Por ejemplo, el estándar Ethernet Gigabit (1000BASE-T) permite transmitir datos a una velocidad de 1 Gbps utilizando cables de categoría 5e o superior.
- **Wi-Fi (IEEE 802.11)**: este protocolo define las normas para la transmisión de datos en redes inalámbricas. Incluye aspectos como las frecuencias de transmisión (2,4 GHz y 5 GHz) y las técnicas de modulación utilizadas para maximizar la velocidad y la fiabilidad.

- Por ejemplo, el estándar Wi-Fi 6 (802.11ax) ofrece velocidades más altas y una mejor gestión de dispositivos conectados simultáneamente.

- **Protocolo de señalización óptica (SONET/SDH)**: utilizado en redes de fibra óptica, este protocolo define cómo se transmiten los datos en forma de pulsos de luz a través de largas distancias.

- **Bluetooth (IEEE 802.15)**: diseñado para comunicaciones a corta distancia, este protocolo utiliza ondas de radio para transmitir datos entre dispositivos cercanos, como móviles o auriculares.



Estos protocolos establecen cómo se transmiten los datos y garantizan que las señales sean interpretadas correctamente por los dispositivos receptores, independientemente del fabricante o modelo.

3. El nivel de enlace.

El nivel de enlace se encarga de garantizar una comunicación confiable entre dispositivos conectados a la misma red. Este apartado detalla tecnologías como Ethernet y el uso de direcciones físicas, elementos esenciales para asegurar la correcta transmisión y recepción de datos en redes locales.

3.1. Redes Ethernet.

Ethernet es, sin duda, una de las tecnologías más utilizadas en redes locales (LAN). Introducida en la década de 1970, se ha convertido en un estándar gracias a su capacidad para ofrecer conexiones rápidas, estables y escalables. Este protocolo opera Principalmente en el nivel de enlace, aunque también interactúa con el nivel físico para gestionar la transmisión de datos.

En Ethernet, los datos se dividen en tramas que incluyen tanto la información a transmitir como detalles adicionales necesarios para garantizar su correcta entrega. Cada trama contiene un encabezado con información clave, como las direcciones físicas de origen y destino, y un campo de verificación (CRC) para detectar posibles errores durante la transmisión.

Por ejemplo, si un ordenador envía un archivo a una impresora conectada a la misma red, Ethernet se encarga de fragmentar los datos en tramas, etiquetarlas correctamente y transmitirlas a través del medio físico. La impresora, al recibir estas tramas, las reensambla para reconstruir el archivo original.

Uno de los aspectos que ha impulsado la popularidad de Ethernet es su capacidad para adaptarse a diferentes medios de transmisión y velocidades. Desde los Primeros estándares que ofrecían 10 Mbps hasta las actuales redes Gigabit Ethernet y Ethernet de 10 Gbps, esta tecnología sigue siendo la base de la mayoría de las redes empresariales y domésticas.

Además, en entornos modernos, el uso de switches en lugar de hubs ha mejorado significativamente el rendimiento de las redes Ethernet. Los switches permiten una comunicación directa entre los dispositivos involucrados, reduciendo colisiones y optimizando el ancho de banda disponible.

3.2. Direcciones físicas.

Las direcciones físicas, también conocidas como direcciones MAC (Media Access Control), son identificadores únicos asignados a cada dispositivo que se conecta a una red. Estas direcciones son esenciales para que el nivel de enlace pueda dirigir las tramas al destinatario correcto dentro de una red local.

Una dirección MAC está compuesta Por48 bits y se representa en formato hexadecimal (Por ejemplo, `00:1A:2B:3C:4D:5E`). Los Primeros 24 bits suelen identificar al fabricante del dispositivo, mientras que los últimos 24 son asignados de manera única por el fabricante para evitar duplicados.

Por ejemplo, en una red doméstica con varios dispositivos conectados, como un ordenador, un televisor inteligente y un móvil, cada uno tiene una dirección MAC distinta. Cuando el router recibe datos destinados al televisor, utiliza esta dirección para asegurarse de que la información llegue al dispositivo correcto.

Las direcciones MAC son permanentes y están integradas en el hardware de los dispositivos, como las tarjetas de red. Sin embargo, en algunos casos, pueden ser "enmascaradas" o sustituidas mediante técnicas como la suplantación de direcciones MAC (MAC spoofing), que se utiliza tanto para fines legítimos como para actividades malintencionadas.

Un aspecto interesante de las direcciones MAC es su papel en protocolos como ARP (Address Resolution Protocol), que permite a los dispositivos de una red local asociar direcciones IP con direcciones MAC. Esto facilita la comunicación entre los niveles de red y de enlace.

4. El nivel de transporte.

El nivel de transporte gestiona la entrega confiable de datos entre dispositivos en una red. Este apartado analiza protocolos fundamentales como TCP y UDP, los esquemas de direccionamiento, el uso de puertos y los servicios básicos de red, permitiendo comprender cómo se garantiza la comunicación efectiva entre sistemas distribuidos.

4.1. El protocolo TCP/IP.

El conjunto de protocolos TCP/IP es la base de la mayoría de las redes modernas, incluida Internet. Este modelo, desarrollado en la década de 1970, divide las comunicaciones en varias capas, cada una con funciones específicas. El nivel de transporte, dentro de este modelo, se apoya Principalmente en dos protocolos: **TCP** (Transmission Control Protocol) y **UDP** (User Datagram Protocol).

TCP proporciona una comunicación confiable, orientada a la conexión. Esto significa que antes de transmitir datos, establece una conexión entre los dispositivos emisores y receptores, asegurándose de que los datos lleguen de forma completa y en el orden correcto.

- Por ejemplo, cuando envías un correo electrónico, TCP asegura que el mensaje completo llegue al servidor de destino sin errores.

Por otro lado, UDP es un protocolo más simple y rápido, pero no garantiza la entrega de los datos ni el orden. Es ideal para aplicaciones donde la velocidad es más importante que la fiabilidad, como las transmisiones en vivo o los videojuegos en línea.

4.2. Esquemas de direccionamiento.

Para que los datos lleguen a su destino, es fundamental identificar de manera única cada dispositivo en la red. Esto se logra mediante **esquemas de direccionamiento**, como las direcciones IP.

Las direcciones IP pueden ser de dos tipos:

- **IPv4**: utiliza un esquema de 32 bits, lo que permite un máximo de aproximadamente 4.300 millones de direcciones únicas (Por ejemplo, 192.168.1.1).
- **IPv6**: utiliza un esquema de 128 bits, ofreciendo una cantidad Prácticamente ilimitada de direcciones (Por ejemplo, 2001:0db8:85a3:0000:0000:8a2e:0370:7334).

Por ejemplo, en una red doméstica, cada dispositivo tiene una dirección IP única asignada Por el router, lo que permite que los datos enviados desde un ordenador lleguen a una impresora o a otro dispositivo conectado.

El nivel de transporte utiliza estas direcciones en conjunto con los puertos para identificar tanto el dispositivo como la aplicación destino.

4.3. El nivel de transporte. protocolos TCP y UDP. Otros protocolos de uso común.

Los dos protocolos más conocidos en esta capa son TCP y UDP, pero no son los únicos. Otros protocolos también desempeñan un papel importante en diferentes aplicaciones:

- **SCTP (Stream Control Transmission Protocol)**: combina características de TCP y UDP, ofreciendo fiabilidad y Soporte para múltiples flujos de datos.
- **DCCP (Datagram Congestion Control Protocol)**: diseñado para aplicaciones sensibles a la latencia, como las de streaming, que requieren control de congestión, pero no la fiabilidad de TCP.

Por ejemplo, un sistema de videoconferencia puede utilizar UDP para el audio y el vídeo, pero TCP para garantizar la entrega de mensajes de texto en el chat integrado.

Sabías que…

El auge de ipv6 está redefiniendo el esquema de direccionamiento en internet, permitiendo una cantidad prácticamente ilimitada de dispositivos conectados. Esta evolución es necesaria en un mundo dominado por el iot, donde cada sensor y electrodoméstico requiere una dirección única para interactuar en la red.

4.4. Puertos.

En el nivel de transporte, los puertos son números que identifican de manera única a las aplicaciones dentro de un dispositivo. Esto permite que varios Programas utilicen la red simultáneamente sin interferir entre ellos.

Los puertos están divididos en varias categorías:

- **Puertos bien conocidos** (0-1023): reservados para servicios comunes, como el puerto 80 para HTTP o el puerto 443 para HTTPS.
- **Puertos registrados** (1024-49151): utilizados por aplicaciones específicas.
- **Puertos dinámicos o Privados** (49152-65535): asignados de forma temporal para conexiones cliente.

Por ejemplo, al abrir una página web, tu navegador utiliza un puerto dinámico para conectarse al servidor, que responde a través del puerto 80 o 443, dependiendo del tipo de conexión.

4.5. Servicios de red básicos.

El nivel de transporte también gestiona los **servicios de red básicos**, que son esenciales para la conectividad y la funcionalidad de las aplicaciones:

- **HTTP/HTTPS**: permiten la transferencia de información en la web.
- **SMTP/IMAP/POP3**: gestionan el envío y recepción de correos electrónicos.
- **FTP/SFTP**: facilitan la transferencia de archivos.
- **DNS (Domain Name System)**: traduce nombres de dominio como www.ejemplo.com en direcciones IP que los dispositivos pueden entender.

Por ejemplo, cuando introduces una URL en tu navegador, el servicio DNS convierte el nombre del dominio en una dirección IP, permitiendo que el navegador establezca la conexión adecuada para cargar la página web.

Actividad 4

Relaciona los modelos con sus características:

Modelo Cliente/Servidor

Microservicios

Modelos basados en mensajes

a) Usa un servidor central que gestiona las solicitudes de los clientes.

b) Divide funciones en pequeñas unidades independientes conectadas a través de APIs.

c) Utiliza colas para permitir comunicación asincrónica entre componentes.

Actividad 5

¿Cómo influye la elección del modelo de programación en la escalabilidad de un sistema de streaming?

Actividad 6

Si diseñaras un servicio para gestionar millones de mensajes por segundo, ¿optarías por un modelo basado en mensajes o microservicios? Justifica.

5. Prueba de autoevaluación.

¿Qué caracteriza al modelo cliente-servidor?

a) La descentralización de tareas entre servicios pequeños

b) La existencia de un servidor central que ofrece servicios

c) El uso exclusivo de microservicios

¿Qué protocolo se utiliza comúnmente para transmitir datos en redes locales?

a) Wi-Fi

b) Ethernet

c) TCP

¿Qué se define en el nivel físico de una red?

a) Las direcciones IP de los dispositivos

b) La forma en que los datos se transmiten físicamente

c) Los servicios web disponibles

¿Qué protocolo se utiliza en las direcciones de redes IPv6?

a) ARP

b) SONET

c) 2001:db8: :/32

¿Qué dispositivo traduce nombres de dominio en direcciones IP?

a) Router

b) DNS

c) NIC

En el modelo _____, un servidor ofrece recursos que son consumidos por los clientes.

El nivel físico incluye dispositivos como _____ y cables de fibra óptica.

El protocolo _____ permite la conexión entre dispositivos en redes locales cableadas.

Las direcciones IPv4 utilizan un esquema de _____ bits.

El sistema _____ traduce nombres de dominio a direcciones IP.

La programación de servicios de comunicaciones abarca desde el uso de librerías hasta la implementación de componentes avanzados para garantizar conexiones fiables. Este capítulo explora estándares y protocolos utilizados en sistemas operativos, el uso de sockets para programación cliente/servidor, y técnicas de depuración para asegurar la estabilidad y el rendimiento de los servicios. También se incluyen herramientas modernas de monitorización y control del ancho de banda.

Programación de servicios de comunicaciones