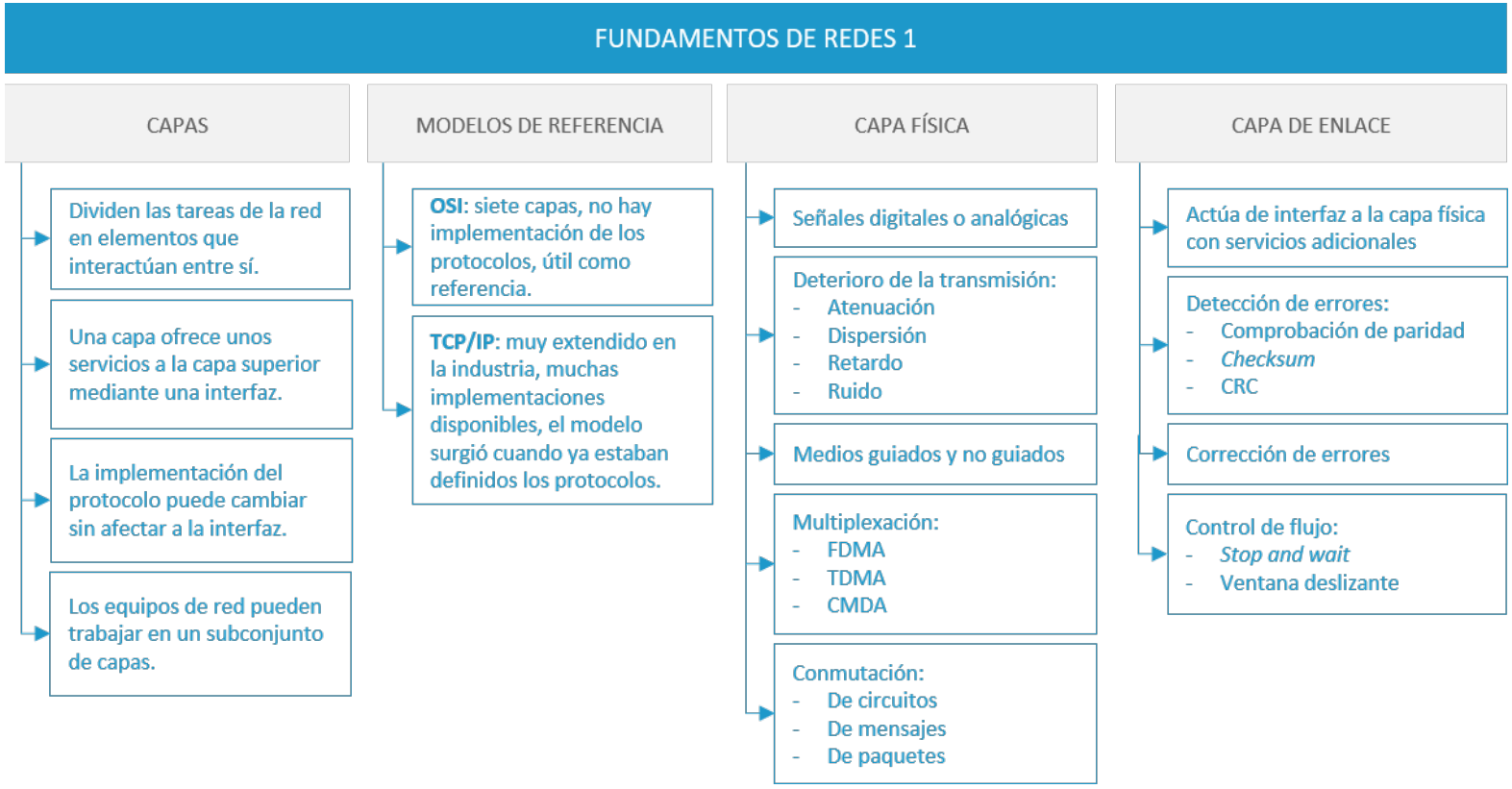
SecDevOps y Administración de Redes para Cloud

Fundamentos de redes 1

Índice

Esquema



Ideas clave

2.1. Introducción y objetivos

Las **redes de ordenadores,** tanto en entornos tradicionales, como de nube, son un tema muy complejo para atacarlo como un elemento monolítico. A tal efecto, los **modelos de red** son una herramienta imprescindible en el estudio, diseño, análisis e implementación de cualquier red. En este tema se presentan estos modelos.

Los objetivos que se pretenden conseguir en este tema son:

- Conocer los dos principales modelos de red en uso.
- Identificar las capas de cada modelo.
- Conocer a alto nivel las capas físicas y de enlace.
- Poner en contexto los protocolos existentes en las capas física y de enlace.

2.2. Modelos de red

La transmisión de datos entre dos equipos, que es el objetivo final de la ingeniería de red, es una tarea que debe solucionar múltiples problemas físicos y lógicos a base de elementos hardware y software. Para facilitar el diseño, implementación, solución de errores y, en general, el razonamiento sobre estos problemas, el concepto de red se divide en **múltiples capas.**

Cada capa resuelve una serie de problemas, liberando a las otras capas de estos problemas.

En el siguiente vídeo, titulado **Modelo de capas,** se explican los principales conceptos de la arquitectura de capas.

 Accede al vídeo

Tareas en capas

En la **arquitectura en capas** todo el proceso de intercambio de tráfico de red se divide en **pequeñas tareas.** Cada tarea se asigna a una capa determinada, que funciona de manera dedicada para procesar esa tarea específica. De esta forma, cada capa se encarga de **todos los detalles de un trabajo concreto.** Cada capa suele solucionar varios problemas para las capas superiores.

En el sistema de comunicación por capas, una capa de un *host* se ocupa de la tarea realizada por su capa homónima en el mismo nivel, en el *host* remoto.

Un *host* se puede referir a un elemento de la red: un *switch,* un *router,* un amplificador de señal, una interfaz de red, etc. Básicamente, cualquier equipo que actúa en la red, es decir, que recibe y envía tráfico en cualquiera de las capas, puede ser considerado un *host.*

En el envío, la **capa del nivel superior** compone un **paquete de datos** y le delega el envío a la capa inmediatamente por debajo de ella. Esta segunda capa hará lo propio, añadiendo **cabeceras** que servirán para resolver los problemas de los que se encarga esta capa. Durante la recepción, se sigue el camino inverso: los datos llegarán a la **capa inferior,** que desempaquetará los datos y los entregará a la capa inmediatamente por encima.

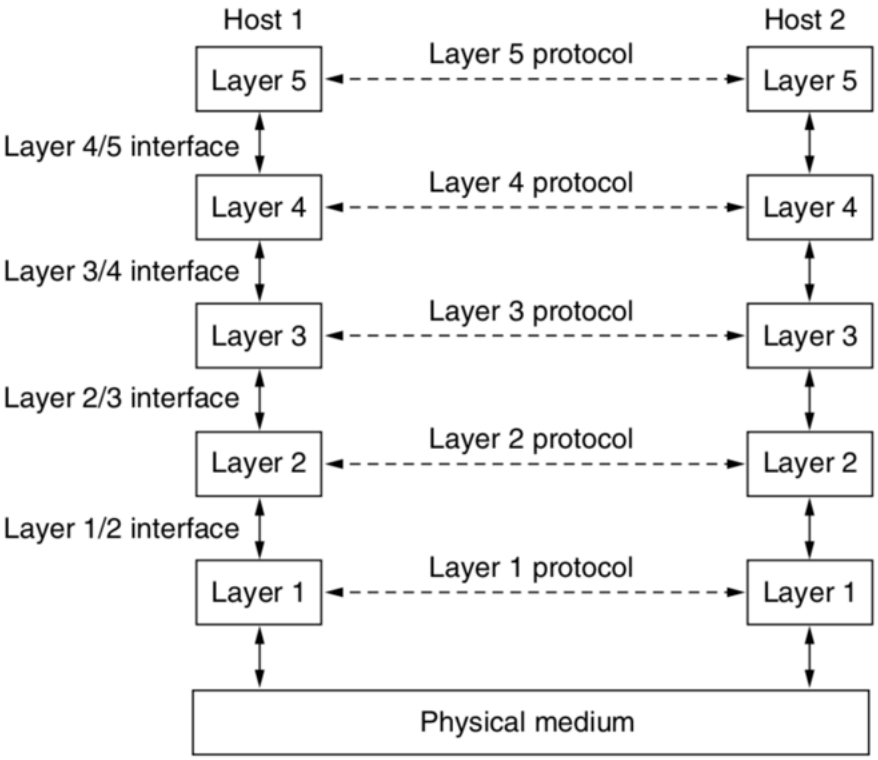


Figura 1. Capas de red. Fuente: Tanenbaum y Wetherall, 2011.

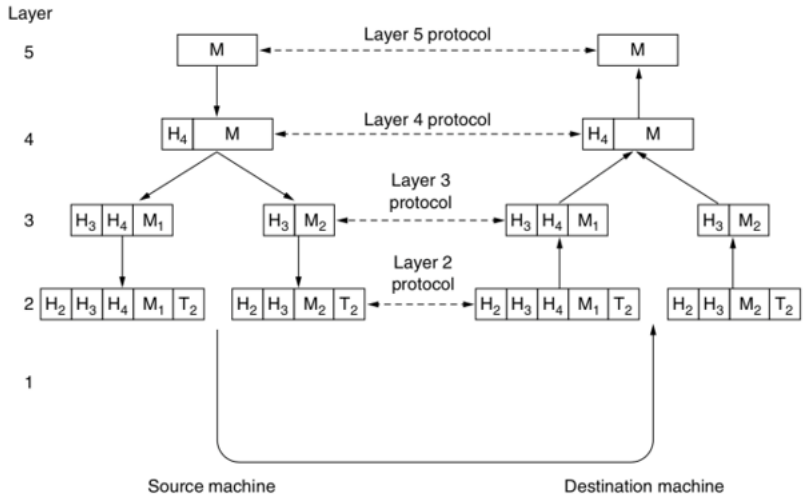


Figura 2. Cabeceras añadidas en cada capa de red. Fuente: Tanenbaum y Wetherall, 2011.

La **interacción entra las capas de varios equipos de red** se representa en la Figura 3. Aunque, simplificado, el diagrama contiene la mayoría de los elementos presentes en una **conexión real.** El escenario es una conexión HTTPS iniciada por un **navegador web** desde el portátil Windows en una red doméstica, hasta el **servidor Linux** alojado en el centro de datos del ISP.

Accede a la «Figura 3. Ejemplo de flujo de red a través de cinco capas» a través del aula virtual.

De la capa superior a la inferior se pueden diferenciar los siguientes **elementos:**

- **Aplicación:** el tráfico en la capa de aplicación es extremo a extremo. El contenido transferido por el protocolo HTTPS de la capa de aplicación a la capa de transporte debe permanecer inalterado hasta que llegue al servidor. Los **nodos intermedios** no tienen por qué interpretar siquiera este tráfico: podría ser una transferencia de archivos por FTP y las capas inferiores no cambiarían.

- **Transporte:** el flujo en esta capa es también extremo a extremo. La pila TCP/IP de Windows abrirá un *socket* en un puerto aleatorio en el cliente y añadirá cabeceras a las tramas HTTPS, en las que indicará el puerto del equipo de destino, 443 por ser HTTPS. Además, **dividirá las tramas** en unidades de menor tamaño e iniciará el envío poco a poco, aumentando la velocidad para evitar congestionar la línea. Si algún paquete se pierde, TCP lo reenviará sin que el servidor web, ni el navegador, hagan nada al respecto. Además, el servidor recibirá los paquetes en orden. Este tipo de tráfico se denomina **orientado a conexión.**

- **Red:** el tráfico IP no es extremo a extremo. Cada equipo enviará los paquetes a su *gateway* por defecto o, en el caso de los *routers*, al siguiente salto en función de su tabla de rutas. Cada paquete del flujo TCP superior puede seguir un camino diferente sin que por ello IP incumpla sus garantías de servicio. Aunque el diagrama solo muestra un ***router* del ISP,** es habitual que haya más de uno. En caso de caída de uno, el ***router* de fibra doméstico** seguiría enviando los paquetes al ***router* activo.** Los equipos sin IP, como el punto de acceso Wifi y el sw*itch ethernet,* solo reenvían los paquetes, sin considerar la ruta IP.

- **Enlace:** esta capa se encarga de asegurar que no haya errores de transmisión en cada enlace físico. Por ejemplo, el punto de acceso WiFi recibe los paquetes por la antena WiFi y los retransmite, sin errores, por su tarjeta *ethernet.* Todos los equipos, salvo los que solo se involucran en la capa física, reciben y reenvían los paquetes en esta capa.

- **Física:** el medio físico puede ser tan diverso como sea necesario. En el ejemplo hay un **medio inalámbrico** en la conexión entre el portátil y el punto de acceso, varios cables *ethernet* que pueden ser de diferentes categorías y un **enlace de fibra** entre el *router* doméstico y el *router* del ISP. El amplificador de fibra se encarga de amplificar la señal óptica analógica, sin siquiera interpretar la señal digital (los amplificadores de señal son más habituales en medios electromagnéticos, pero el ejemplo a nivel de capas aplica igualmente).

De manera formal, **cada capa ofrece un servicio a la capa superior.** Por ejemplo, la capa N puede ofrecer garantías de entregas sin errores a la capa N+1. Las reglas para el **funcionamiento interno** de una capa definen un protocolo. Una capa ofrece un servicio mediante una interfaz, es decir, una serie de primitivas de software a modo de funciones con sus parámetros.

Si se hace necesario cambiar la **implementación de un protocolo** (por ejemplo, para eliminar un *bug*) pero se mantiene la interfaz, solo será necesario modificar los componentes de una capa, de manera que se reduce la complejidad y el alcance del cambio.

Modelos de referencia

Hay **dos modelos** de red principales en uso: **el modelo OSI** y el **modelo TCP/IP.** El modelo OSI no ha llegado a ver una implementación en la industria y, por tanto, sus protocolos no se usan. No obstante, el diseño se usa extensivamente. El modelo TCP/IP, por el contrario, tiene como punto fuerte sus protocolos, ya que es el estándar de facto en las redes de ordenadores e Internet.

Modelo OSI

La International Organization for Standardization (ISO), en español, Organización Internacional de Estandarización, publicó en 1980 el estándar OSI, de *Open System Interconnect.* OSI es un estándar abierto para cualquier sistema de comunicación formado por siete capas detalladas en la Tabla 1.

Capas OSI		
Nivel 7	Aplicación	Es responsable de proporcionar interfaz al usuario de la aplicación. Incorpora las funcionalidades que interactúan con el usuario final.
Nivel 6	Presentación	Define cómo traducir el contenido al formato nativo del host destino.
Nivel 5	Sesión	Establece y mantiene un concepto de sesión durante la comunicación a partir de un establecimiento inicial, por ejemplo, a partir de una autenticación a base de credenciales.
Nivel 4	Transporte	Es responsable de la entrega de extremo a extremo entre hosts, es decir, los nodos intermedios entre origen y destino no deberían afectar a este contenido. También se encarga de dividir los paquetes del tráfico de la aplicación en unidades más pequeñas y de asegurar la entrega en orden.
Nivel 3	Red	Es responsable de la asignación de direcciones de hosts en una red y el enrutamiento.
Nivel 2	Enlace de datos	Interactúa directamente con la capa física para convertir los datos en el elemento físico de la transmisión. También se encarga de detectar y corregir errores de transmisión.
Nivel 1	Física	Define el hardware, cableado, potencia de salida, pulso, frecuencias, longitud de onda, ancho de banda, etc.

Tabla 1. Capas del modelo OSI. Fuente: elaboración propia.

Internet utiliza la **suite de protocolos TCP/IP,** por lo que este modelo se puede denominar también **modelo de Internet.** El modelo OSI es un modelo de comunicación general, pero el modelo TCP/IP es una colección de protocolos concreta, utilizada en Internet y en multitud de redes privadas. Mientras que el modelo OSI precedió a los protocolos OSI, la implementación de TCP/IP vino primero y el modelo se definió después.

Sin entrar en detalles sobre su historia, los **criterios** que dieron pie a TCP/IP fueron:

- Resistencia a la pérdida de elementos de red, sin afectar al tráfico existente.
- Arquitectura flexible capaz de soportar tanto transferencia de ficheros, como tráfico en tiempo real.

El resultado fue una **red de conmutación de paquetes** basada en una capa no orientada a conexión, capaz de interconectar múltiples subredes.

El modelo TCP/IP está compuesto de las **capas** ampliadas en la Tabla 2.

Capas TCP/IP		
Nivel 5	Aplicación	Define el protocolo que permite al usuario interactuar con los servidores de la red. Por ejemplo, FTP, HTTP, etc.
Nivel 4	Transporte	Define cómo deben fluir los datos en el camino extremo a extremo entre los hosts. Hay dos protocolos principales: *Transmission Control Protocol* (TCP) y *User Datagram Protocol* (UDP). TCP es orientado a conexión, y garantiza la entrega en orden de todos los paquetes. UDP, por el contrario, es no orientado a conexión y no asegura, ni la entrega, ni el orden.
Nivel 3	Red	El protocolo de internet (IP) es el estándar de facto. Facilita el direccionamiento, el reconocimiento del host y el enrutamiento. Define también un protocolo de control, ICMP.
Nivel 2	Enlace	Define los requisitos que deben cumplir los enlaces y las líneas de datos respecto a la capa superior. Ofrece control de flujo y de errores.

Tabla 2. Capas del modelo TCP/IP. Fuente: elaboración propia.

Como modelos que son, ambos son útiles para **razonar sobre los servicios y protocolos de la red.** Sin embargo, cada modelo tiene sus **ventajas** y sus **inconvenientes:**

- En la **capa de transporte,** TCP/IP define un protocolo orientado a conexión y uno no orientado a conexión, mientras que OSI solo define un protocolo orientado a conexión.

- Los **protocolos de OSI** no han llegado a implementarse, mientras que el **modelo TCP/IP** se definió como modelo después de la implementación de los protocolos. La industria no llegó a implementar los protocolos de OSI porque TCP/IP ya estaba muy extendido y ninguna compañía quería soportar dos pilas de protocolos diferentes, aunque el modelo OSI tuviera una definición formal más correcta que la de TCP/IP.

- Aunque el **modelo TCP/IP** no hace distinciones claras entre servicios, interfaces y protocolos, y no es tan general como OSI. Sus protocolos tienen implementaciones para **múltiples sistemas.** La implementación original de TCP/IP para UNIX fue rápidamente aceptada por su calidad, extendiendo su uso en la comunidad.

- La elección de **siete capas** en el modelo OSI fue más política que técnica y las interacciones entre capas son tan complejas, que las primeras implementaciones fueron de muy mala calidad.

La **capa física** no está propiamente definida en el modelo TCP/IP, así que se seguirá el detalle de la capa equivalente del modelo OSI con referencias a los protocolos usados en entornos reales.

2.3. Introducción a la capa física

La *capa física* es la responsable de definir **cómo trabaja el hardware de red,** las propiedades de los cables, las frecuencias y modulaciones, etc. En el modelo OSI es la única capa que hace referencia a la **conectividad física entre dos equipos,** ya que el resto de las capas trabajan con abstracciones de software.

En el siguiente vídeo, titulado **Capa física y capa de enlace,** se resumen las características de las capas físicas y de las capas de enlace.

 Accede al vídeo

La capa de enlace de datos usa la interfaz de la capa física para **pasarle tramas, que esta convierte en pulsos eléctricos** en un cable de cobre, en señales electromagnéticas en un protocolo inalámbrico y en pulsos de luz en un cable de fibra óptica.

Deterioro de la transmisión

Las **señales tienden a deteriorarse** cuando viajan a través del medio por diversas razones, por ejemplo:

- **Atenuación:** la potencia de la señal (o más concretamente, la densidad de potencia de la señal) se reduce a medida que ésta se transmite por el medio. La señal debe tener suficiente potencia en la recepción para que el receptor pueda interpretarla correctamente.

- **Dispersión:** a medida que la señal viaja a través del medio, la banda de frecuencia tiende a extenderse y solaparse.

- **Retardo:** aunque el protocolo defina la velocidad y frecuencia de transmisión, los equipos pueden tener defectos en el hardware, que hagan que los parámetros varíen. Si la velocidad de la señal y la frecuencia no coinciden en ambos extremos, hay posibilidades de que la señal llegue al destino de manera arbitraria.

- **Ruido:** se llama ruido en la señal a la perturbación aleatoria que tiene la capacidad de distorsionar la información real que se está transmitiendo. Se pueden identificar, entre otros:

  - **Ruido térmico:** producido por la agitación de los conductores electrónicos del medio.
  - **Intermodulación:** producido en transmisiones en banda cuando la frecuencia usada por un canal no se limita adecuadamente y solapa con otros canales.
  - ***Crosstalk:*** producido por señales ajenas a la transmisión, pero que comparten el mismo medio.
  - **Impulso:** producido por perturbaciones irregulares e instantáneas.

Medios de transmisión

A grandes rasgos, los medios de transmisión se pueden clasificar en **guiados** y **no guiados:**

Medios guiados	Cualquier cable, ya sea de coaxial, de cobre o de fibra óptica, es un medio guiado. El cable puede conectar punto a punto dos dispositivos o conectar varios en una arquitectura de bus.
Medios no guiados	Son aquellos en los que no hay un material físico que dirija la señal a su destino. Cualquier dispositivo podría recibir la transmisión por el mero hecho de estar dentro del alcance de la transmisión.

Tabla 3. Clasificación de los medios de transmisión. Fuente: elaboración propia.

La multiplexación es la **técnica** que permite aprovechar un medio para enviar más de un **flujo continuo de información.** En la fuente, un sistema multiplexor combina los **canales** (así se denominan los flujos) y los transmite en un único medio. En el destino, un **demultiplexor** extrae cada canal y lo entrega de manera individual. Las tres técnicas principales de multiplexación principales a nivel físico son las siguientes:

- **Multiplexación por división de frecuencia o FDMA:** aprovecha la transmisión en bandas de frecuencia para compartir un canal. La transmisión de radio y televisión analógica sirve de ejemplo familiar: hay múltiples canales y cada canal ocupa un cierto ancho de banda (kHz en el caso de la radio, MHz en el caso de la TV). El diagrama de la Figura 4 muestra tres canales en banda base (que podrían ser señales analógicas de telefonía de 4 KHz, por ejemplo) que son mezclados en tres bandas diferentes. Cada canal mantiene su ancho de banda original, pero se transmite con una portadora, o frecuencia central, superior.

- **Multiplexación por división de tiempo o TDMA:** divide el flujo de tiempo en un número fijo de intervalos. Por ejemplo, en una división en tres intervalos se pueden multiplexar tres canales, tal como muestra la Figura 5. Cada canal aprovecha todo el ancho de banda posible en su intervalo asignado. En esta técnica, el medio multiplexado debe usar una velocidad de transmisión más alta que la suma de velocidades de todos los canales.

- **Multiplexación por división de código o CDMA:** convierte una señal de banda estrecha (el equivalente a un canal de FDMA, por ejemplo) en una señal de banda ancha multiplicándola por un código digital mucho más rápido que la variación de la señal de datos. La multiplexación se consigue usando códigos diferentes para cada canal.

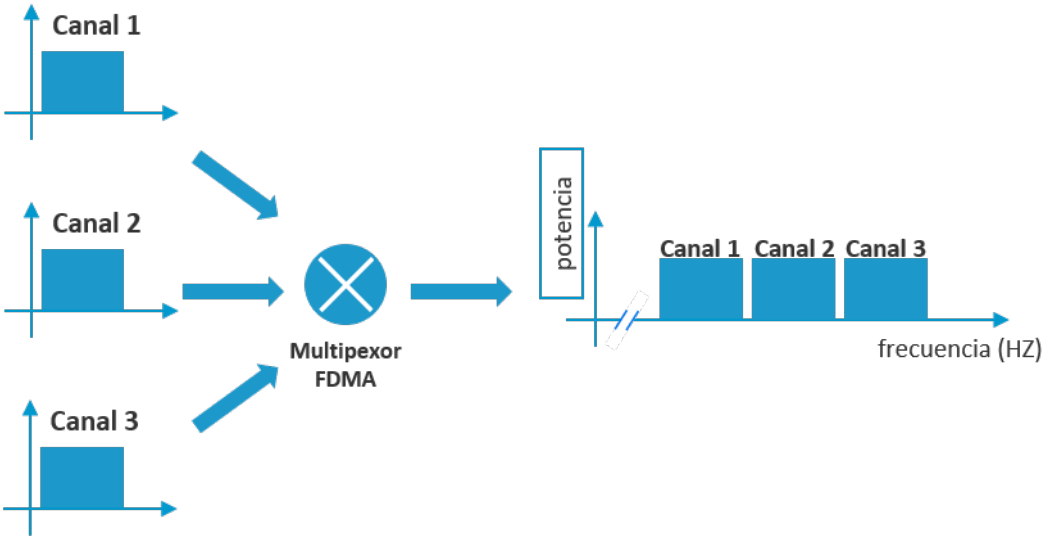


Figura 4. Diagrama de FDMA. Fuente: elaboración propia.

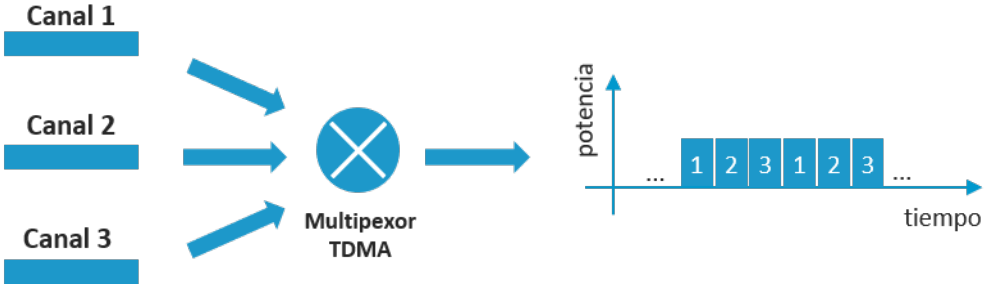


Figura 5. Diagrama de TDMA. Fuente: elaboración propia.

Estas técnicas se pueden combinar. Por ejemplo, la **comunicación digital móvil** de segunda generación (GSM) usa una combinación de FDMA y TDMA (la evolución a 3G cambió a acceso **multiplexación por división de códigos, CDMA).**

Conmutación

La conmutación es el mecanismo que permite **transmitir datos desde una fuente a un destino que no están conectados directamente.**

Los nodos de interconexión de la red reciben datos de fuentes conectadas directamente, los almacenan, analizan y, finalmente, los envían hacia el siguiente **dispositivo de interconexión** más próximo al destino.

A nivel general, la conmutación puede dividirse en dos categorías principales:

- **Sin conexión:** no se requiere ningún enlace previo y la confirmación de recepción es opcional.
- **Orientado a la conexión:** es necesario establecer el circuito a lo largo de la ruta entre ambos extremos antes de poder intercambiar tráfico. Los datos se envían entonces a lo largo del circuito. El circuito puede cerrarse nada más terminar la transferencia o puede mantenerse temporalmente para un uso posterior.

Conmutación de circuitos

En un sistema de conmutación de circuitos, **los datos se transmiten a través de un canal exclusivo.** Es necesario que estén especificados los datos que viajan por esa ruta, ya que no se permiten otros en la misma ruta. Además, el **circuito debe estar establecido** para que la transferencia de datos pueda tener lugar. Los circuitos pueden establecerse antes de la primera comunicación y mantenerse indefinidamente (como en MPLS) o establecerse y cerrarse a demanda para cada transmisión.

El ejemplo más claro de conmutación de circuitos es la **telefonía analógica tradicional,** ya que la señal viaja por un canal exclusivo, no compartido, durante toda la duración de la comunicación.

Conmutación de mensajes

La conmutación de mensajes está a **medio camino entre la de circuitos y la de paquetes:** cada mensaje se trata como una unidad y se transmite a lo largo de un canal exclusivo. Un conmutador de mensajes recibe el mensaje entero y **almacena los datos temporalmente** hasta que haya recursos disponibles para transferirlo al siguiente salto del circuito. El conmutador almacenará los datos y se mantendrá a la espera hasta que el conmutador del siguiente salto tenga suficientes recursos.
Al igual que en la conmutación de circuitos, la **red debe reservar una ruta exclusiva para la transmisión.** La conmutación de mensajes tiene dos inconvenientes principales:

- Cada dispositivo en la trayectoria del tránsito necesita capacidad de almacenaje suficiente para gestionar el mensaje entero.
- La técnica de almacenamiento y reenvío, junto a la latencia debido a las esperas intermedias, hacen que esta técnica sea muy lenta en comparación a la conmutación de paquetes.

La conmutación de mensajes se sustituyó por **conmutación** de paquetes al no ser una buena solución para los medios de transmisión en tiempo real.

Conmutación de paquetes

En la conmutación, cada mensaje se fragmenta en **paquetes de menor tamaño.** Cada uno lleva asociado una serie de cabeceras con información sobre el destino, control de errores, etc. Cada paquete, con sus cabeceras, se transmite de manera independiente al resto de paquetes.

La ventaja de la conmutación de paquetes, respecto con la conmutación de circuitos, es que se puede aprovechar mejor la **capacidad de las líneas:** un circuito reservado no permite la transmisión de más datos, incluso, aunque en un momento dado, ni emisor ni receptor estén enviando datos activamente. Los paquetes, sin embargo, pueden ***multiplexarse*** **a la velocidad que permita la línea,** por lo que los silencios de transmisión de un nodo pueden aprovecharse por otros.

Las redes IP son esencialmente **redes de conmutación de paquetes,** aunque dan soporte para calidad de servicio (QoS, Quality of Service), dando más prioridad a unos paquetes frente a otros.

2.4. Introducción a la capa de enlace de datos

La capa de enlace de datos abstrae los conceptos físicos y la implementación hardware a las capas superiores. Actúa entre **dos nodos que están conectados directamente en un mismo medio.** Cuando en un mismo medio hay múltiples nodos, esta capa se encarga de evitar colisiones.

Es, por tanto, la capa encargada de **traducir los datos** en parámetros que el hardware puede convertir en señales físicas. El receptor recoge los **datos del hardware,** que están en forma de señales eléctricas u ópticas, los ensambla en un formato de reconocible, y los entrega a la capa superior.

Características de la capa de enlace de datos	
Encapsulado	La capa de enlace de datos toma los paquetes de la capa de red y los encapsula en tramas o *frames,* que luego envía bit a bit al hardware. La capa de enlace del extremo receptor recoge las señales de hardware y ensambla las tramas en los paquetes que entrega a la capa de red.
Direccionamiento	La capa de enlace de datos proporciona el mecanismo de direccionamiento de hardware. En esta capa, cada dirección es única a nivel de enlace y se codifica en hardware durante la fabricación (en entornos virtuales, las direcciones pueden no ser globalmente únicas, pero los orquestadores son capaces de prevenir estas situaciones). Las direcciones de Medium Access Control (MAC) se encuentran en las tarjetas *ethernet* y Wifi y permiten identificar unívocamente cada interfaz en un medio. Al contrario que las direcciones IP, las direcciones MAC no se usan para enrutamiento y son fijas (los *switches ethernet* pueden usarlas para reducir el número de puertos por los que replica un paquete, pero el concepto es diferente al de enrutamiento IP).
Sincronización	Los equipos involucrados en una transferencia deben sincronizarse antes de la misma, tanto para que usen la misma velocidad de bit como para detectar el principio de una trama.
Control de errores	Las señales pueden encontrar problemas en la transmisión y los bits pueden estar invertidos. Algunos errores se pueden detectar y notificar y otros se pueden corregir en destino.
Control de flujo	Los nodos en el mismo enlace pueden funcionar a diferentes velocidades. La capa de enlace de datos controla estas situaciones para que ambos nodos puedan trabajar a una velocidad común.
Acceso múltiple	En medios compartidos es posible que dos nodos intenten emitir señales a la vez. Estas señales colisionan y se vuelven ilegibles por el receptor. La capa de enlace de datos es capaz de hacer uso de técnicas como FDMA, TDMA o CDMA ofrecidas por la capa física, dar la capacidad de acceder a un medio compartido entre múltiples sistemas.

Tabla 4. Características de la capa de enlace de datos. Fuente: elaboración propia.

Los diferentes **tipos de ruido,** mencionados anteriormente, pueden provocar **errores en la transmisión,** volviendo los datos ilegibles. Las capas superiores trabajan sobre una vista generalizada de **arquitectura de red** y asumen, gracias al servicio ofrecido por la capa de enlace, que estos problemas no existen. La mayoría de las aplicaciones no funcionarán de forma esperada si reciben **datos erróneos** (las aplicaciones de voz y vídeo suelen ser tolerantes a fallos y pueden verse menos afectadas).

Detección de errores

Entre las técnicas de detección se pueden contar la comprobación de paridad, el uso de ***checksums*** y los **algoritmos de verificación de redundancia cíclica** (CRC). Todas estas técnicas implicar enviar **bits** adicionales, calculados a partir de los datos. En la recepción se vuelven a calcular estos bits a partir del mensaje y se comprueba que los bits calculados y los recibidos coincidan. Si los bits no coinciden, se considera que ha existido un fallo en la transmisión.

Comprobación de paridad

Se envía un **bit adicional,** típicamente por cada 8 bits de datos, de forma que el número de bits a 1 en total sea par. Si el número de bits de datos a 1 es par, se añade un bit de paridad en cada trama con el valor 0; si el número es impar, el bit de paridad tiene el valor 1, de manera que **siempre hay un número par de unos** (hay diferentes convenios y el valor del bit de paridad puede ser el contrario al explicado en este ejemplo).

El **receptor** solo necesita contar el número de unos en una trama. La **trama** se considera válida si el recuento de unos es par, incluido el bit de paridad. En caso contrario, **la trama se considera dañada.**

Esta técnica funciona si solo se alterna un **bit de la trama.** Si el número de bit erróneos es mayor, la comprobación de paridad puede dar **falsos positivos.**

Checksums

Los **bits de paridad** son un caso particular del concepto de *checksum.* Cualquier grupo de bits calculados a partir de los datos se puede consideran un *checksum*, una vez añadidos a la cabecera de una trama. Por ejemplo, el *checksum* usado en las **cabeceras IP** se calcula como la suma de los bits del mensaje, una vez dividido en palabras de 16 bits.

Este *checksum* es capaz de descubrir **errores indetectables** por un bit de paridad. No obstante, es vulnerable a errores más típicos de hardware de mala calidad debido a ruido de la línea, como la inserción de ceros o la **reordenación de palabras.**

Comprobación de redundancia cíclica (CRC)

En esta técnica se transmite un **código polinomial.** Los bits de una trama de longitud N se consideran coeficientes de un polinomio de grado N-1. El código polinomial se calcula de manera que la **trama y el código son divisibles** por un polinomio, que emisor y receptor han acordado antes de la transmisión. Si el cálculo de la división en el receptor tiene un resto diferente de cero, se puede asumir que algún bit ha **cambiado de signo** y la trama se ha **corrompido en tránsito.**

Corrección de errores

Hay **dos modelos** de corrección de errores:

- **Corrección de errores hacia atrás:** más que corregir, este modelo solicita la retransmisión de una trama cuando el receptor detecta un error. Este es el caso comentado en la sección anterior; se usa una técnica de detección para solicitar una retransmisión.
- **Corrección de errores hacia adelante:** cuando el receptor detecta algún error en los datos recibidos, ejecuta una técnica de auto recuperación para corregir el error.

Hay **cuatro algoritmos de corrección de errores principales** (Figura 6).

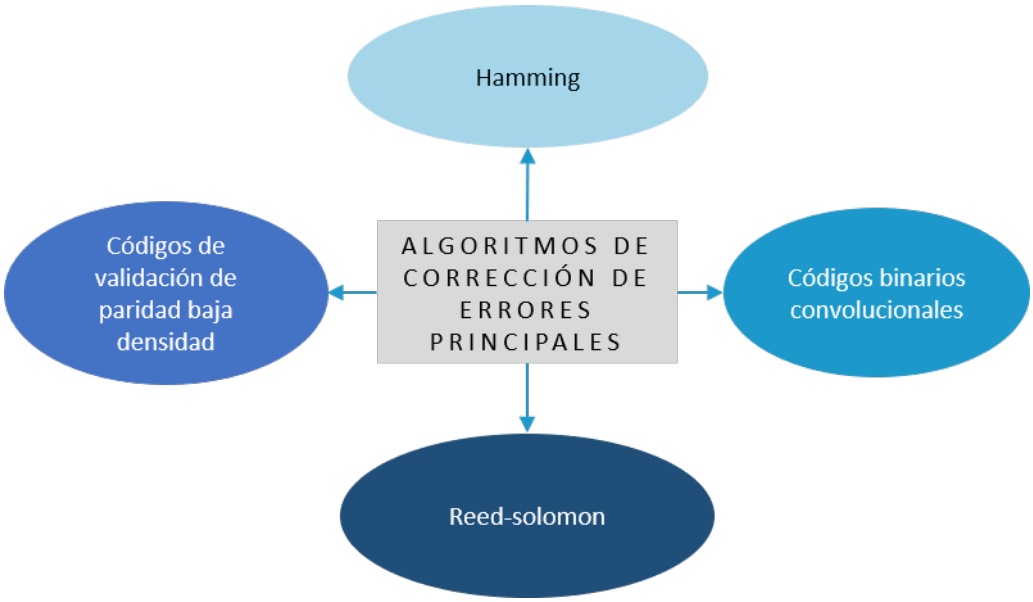


Figura 6. Algoritmos de corrección de errores principales. Fuente: elaboración propia.

Control de flujo

Cuando una trama se envía de un *host* a otro a través de un único medio, es necesario que el remitente y el receptor vayan a la **misma velocidad.**

El **control de flujo** se encarga de asegurar que el emisor no envía datos (es decir, no activa señales eléctricas en el medio) más rápido de lo que el receptor va a poder actuar sobre ellos. Este problema es diferente al de la **sincronización:** el control de flujo asume que emisor y receptor transmiten a la **misma velocidad** (por ejemplo, 9600 bits por segundo), pero que el receptor puede estar haciendo otras tareas que le impidan aceptar una trama en un momento dado.

Flujo

Para entender la situación, se propone un escenario con un *router* conectado a tres equipos. Las tres conexiones son de 10 Mbps. Los nodos A y B envían tráfico al nodo C, y empiezan a transmitir tramas a 10 Mbps. El *router* puede recibir tramas de A y B en paralelo, pero solo puede enviar uno de los flujos a C. El *router* podría almacenar las tramas en un *buffer* interno, pero dado que el *buffer* no va a ser infinito, la solución es controlar el flujo al que A y B envían las tramas. Cada trama se enviará a la velocidad de la línea, 10 Mbps, pero habrá tiempos de espera entre trama y trama que reducirán la velocidad efectiva a, idealmente, 5 Mbps para cada flujo.

Los siguientes apartados explican dos **mecanismos de control de flujo:** detención y espera *(stop and wait)* y ventana deslizante *(sliding window).*

- *Stop and wait*

Cuando un emisor usa este **mecanismo**, se detendrá después del envío de una trama. Solo enviará la siguiente una vez haya recibido un acuse de recibo o *acknowledgement* (ACK) por parte del receptor. El cronograma de la Figura 7 muestra un ejemplo sencillo en el que el **emisor solo puede enviar una trama** al recibir el ACK de la trama anterior.

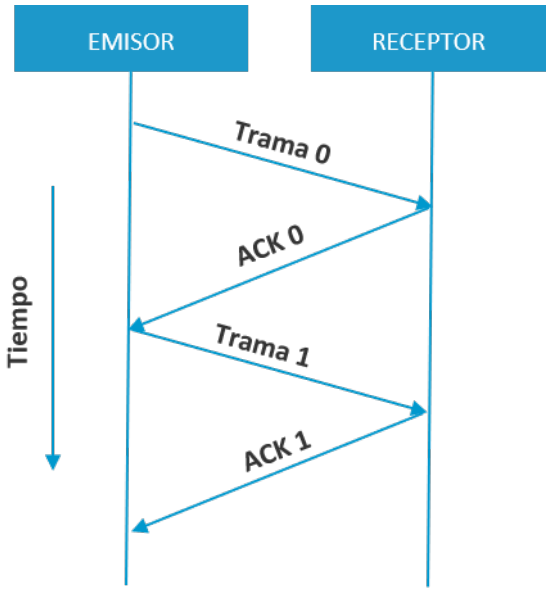


Figura 7. Esquema de control de flujo con *stop and wait.* Fuente: elaboración propia.

- Ventana deslizante

En este mecanismo de control de flujo, el emisor mantiene un **contador de tramas sin ACK.** Tras enviar una trama, el emisor reduce el contador en uno. Cuando recibe un ACK, aumenta el contador en uno. Mientras el contador sea mayor que cero, el emisor es libre de **enviar más tramas.**

Este protocolo aprovecha mejor el uso del canal que la técnica de *stop and wait.* El cronograma de la Figura 8 muestra un **flujo con ventana deslizante de tres tramas.** Se puede observar que el receptor envía el ACK de la primera trama nada más recibirla.

El emisor, al ver que hay un **hueco libre en la ventana,** puede enviar otra trama, pero solo una. El emisor retrasa el ACK de las tramas dos y tres hasta más tarde (por ejemplo, porque no ha podido procesarlas debido a una falta de recursos de CPU). En este caso, puede enviar un **único ACK para las dos tramas.** A la recepción de este ACK, el emisor puede enviar dos tramas más.

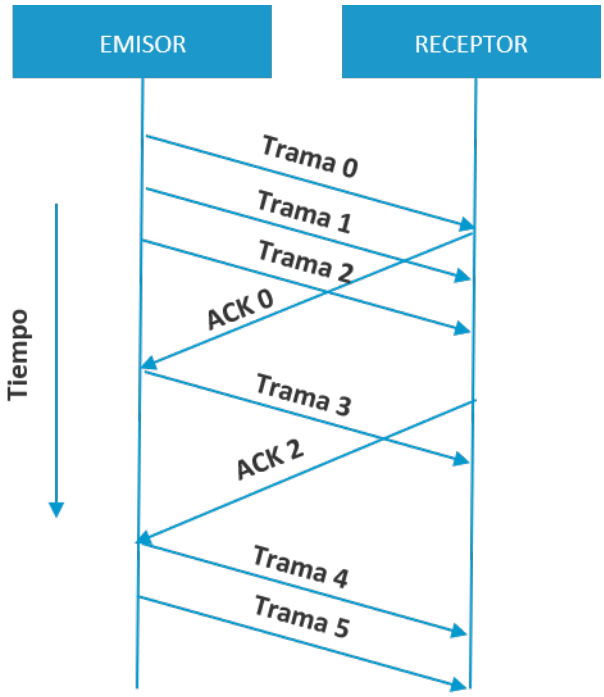


Figura 8. Esquema de control de flujo con ventana deslizante. Fuente: elaboración propia.

2.5. Referencias bibliográficas

Tanenbaum, A. y Wetherall, D. (2011). *Computer Networks.* (5ªed.) Pearson New International.

A fondo

Computer Networks

Tanenbaum, A. y Wetherall, D. (2011). *Computer Networks.* (5ª ed.) Pearson New International.

La referencia básica de las redes de ordenadores entra en mucho más detalle en la arquitectura de capas y en los modelos de referencia, además de presentar muchos más ejemplos de protocolos reales. Te recomendamos sobre este tema los capítulos uno (apartados 1.3 y 1.4) y los capítulos dos, tres y cuatro.

Cloud Computing Networking

Chao, L. (2016). *Cloud Computing Networking.* CRC Press.

El capítulo dos trata muchos de los protocolos. El apartado 2.6, en particular, pone en contexto estos protocolos con la arquitectura de capas.

Test

1. ¿Qué capas componen el modelo TCP/IP?

A. Física, enlace, red, transporte y aplicación.

B. Enlace, red, transporte y aplicación.

C. Física, enlace, red, transporte, sesión, presentación y aplicación.

D. Ninguna de las anteriores.

2. ¿Qué relación hay entre servicios, interfaces y protocolos en el contexto de la arquitectura de capas?

A. Una capa ofrece un servicio a una capa superior cumpliendo una interfaz; la capa trabaja internamente siguiendo un protocolo.

B. Servicio e interfaz son sinónimos y se refieren a las funciones de una capa que pueden usar el resto de las capas. El protocolo se refiere al nombre en la industria de ese servicio.

C. La interfaz se refiere a la funcionalidad que una capa puede usar del protocolo de la capa inferior.

D. Ninguna de las anteriores.

3. ¿Qué servicios ofrece la capa de enlace de TCP/IP?

A. Control de flujo y enrutamiento.

B. Ninguno, la capa de enlace no pertenece a TCP/IP.

C. Garantía de entrega extremo a extremo.

D. Control de flujo y control de errores.

4. ¿Qué servicios ofrece la capa de red de TCP/IP?

A. Direccionamiento.

B. Enrutamiento.

C. Todas las anteriores.

D. Ninguna de las anteriores.

5. ¿Qué diferencia hay entre conmutación de paquetes y conmutación de circuitos?

A. Ninguna.

B. En la conmutación de circuitos, todos los datos de un flujo de tráfico recorren el mismo camino entre origen y destino, que ha de establecerse antes de empezar a transmitir. En la conmutación de paquetes, los datos se dividen en paquetes que pueden recorrer caminos diferentes antes, para llegar a su destino.

C. En la conmutación de circuitos, todos los datos de un flujo de tráfico recorren el mismo camino entre origen y destino, que ha de establecerse antes de empezar a transmitir. En la conmutación de paquetes, los datos se dividen en paquetes, que a su vez siguen un camino preestablecido antes de empezar la transmisión.

D. La conmutación de circuitos se usa para comunicación de audio y la conmutación de paquetes se usa para datos digitales.

6. ¿Cuáles de los siguientes son ejemplos de protocolos de aplicación? Escoge todas las opciones correctas.

A. UDP.

B. HTTP.

C. IP.

D. DNS.

7. ¿En qué se diferencian TCP y UDP?

A. TCP ofrece un servicio no orientado a conexión, mientras que UDP ofrece un servicio orientado a conexión.

B. TCP ofrece un servicio orientado a conexión, mientras que UDP ofrece un servicio no orientado a conexión.

C. TCP funciona en la capa de transporte y UDP en la capa de red.

D. TCP forma parte del modelo TCP/IP y UDP forma parte del modelo OSI.

8. Relaciona la capa con el protocolo correspondiente.

Aplicación	1	A	Ethernet
Transporte	2	B	UDP
Red	3	C	SMTP
Enlace	4	D	IP

9. ¿Por qué el modelo TCP/IP no tiene capas de presentación y sesión?

A. Se asume que cada aplicación se encarga de los servicios de ambas capas.

B. Porque OSI los añadió por razones políticas.

C. TCP/IP sí que tiene ambas capas.

D. Porque están integradas en la capa de transporte.

10. Relaciona cada concepto con el ejemplo correspondiente.

Detección de errores	1	A	FDMA
Control de flujo	2	B	*Checksum*
Tipo de ruido	3	C	Térmico
Multiplexación	4	D	Ventana deslizante