Transmisión de datos digitales

Contenido

Técnica de comunicación de datos Digitales.

La transmisión de una cadena de bits desde un dispositivo a otro, a través de una línea de transmisión, implica un alto grado de cooperación entre ambos extremos. Uno de los requisitos esenciales es la **sincronización.** El receptor debe saber la velocidad a la que se están recibiendo los datos, de tal manera que pueda muestrear la línea a intervalos constantes de tiempo para así determinar cada uno de los bits recibidos. Para este propósito, se utilizan habitualmente dos técnicas.

En la **transmisión asíncrona,** cada carácter se trata independientemente. El primer bit de cada carácter es un bit de comienzo que alerta al receptor sobre la llegada del carácter. El receptor muestrea cada bit del carácter y busca el comienzo del siguiente. Esta técnica puede que no funcione correctamente para bloques de datos excesivamente largos debido a que el reloj del receptor podría perder el sincronismo respecto del emisor. No obstante, la transmisión de datos en bloques grandes es más eficaz que la transmisión carácter a carácter.

Para el envío de bloques grandes se utiliza **la transmisión síncrona.** Cada bloque de datos forma una trama la cual incluirá, entre otros campos, los delimitadores de principio y de fin. Al transmitir la trama se empleará alguna técnica de sincronización, por ejemplo, la obtenida con el código Manchester.

La **detección de errores** se lleva a cabo calculando un código en función de los bits de entrada. El código se añade a los bits a transmitir. Para comprobar si ha habido errores, el receptor calcula el código y lo compara con el código recibido.

La **corrección de errores** opera de forma similar a la detección de errores, pero en este caso será posible corregir ciertos errores en la secuencia de bits recibida.

ARQUITECTURAS DE PROTOCOLOS NORMALIZADAS

Para establecer una comunicación entre computadores 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 propietarios, siendo esto muy costoso, siendo la única alternativa la normalización.

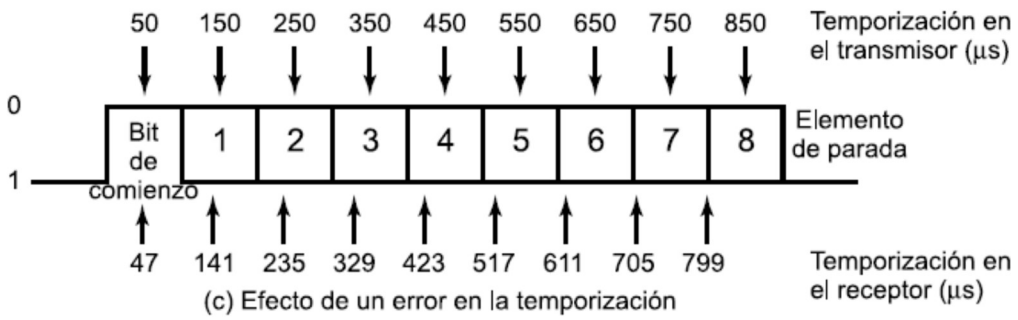
Los estándares tienen las siguientes ventajas: a) Los fabricantes pueden bajar costos y por ende aumentar su mercado y b) el cliente pude exigir una interoperabilidad entre equipos de diferentes fabricantes.

De esta estandarización se destacan los protocolos OSI y TCP/IP

Sincronización

Supóngase que el emisor emite una cadena de bits. Esto se hará de acuerdo con el reloj del transmisor. Generalmente, el receptor deberá muestrear en la parte central de cada bit (mitad del intervalo), obteniendo una muestra por cada intervalo de duración de un bit.

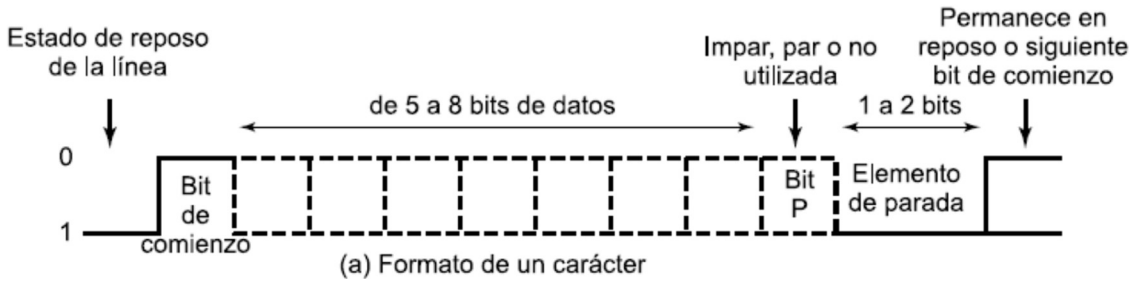
Supongamos que transmitimos a una velocidad de 10 kbps; por tanto, se transmite un bit cada 0,1 milisegundos (ms), es decir, 1 bit tiene una duración de 100 µs. Si el reloj receptor es un 6 por ciento más rápido, el receptor muestrea el carácter de entrada cada 94 µs (medidos con el reloj del transmisor). Como se puede observar, la última muestra será errónea.



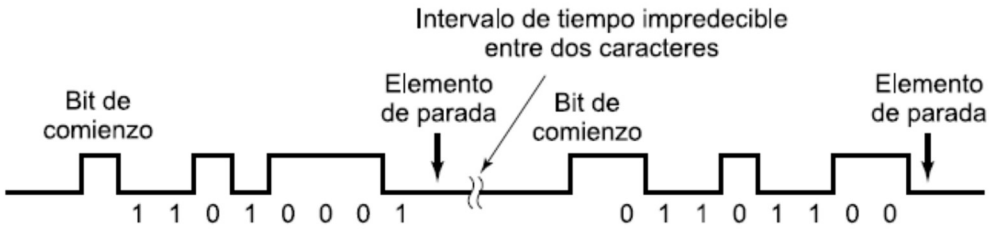
(c) Efecto de un error en la temporización

Transmisión Asíncrona

En esta solución, el inconveniente de sincronismo se evita al enviar los datos carácter a carácter. Normalmente cada carácter no supera los 8 bits, lo que permite sincronizar al Rx al inicio de cada secuencia de datos. Es útil para bajas velocidades de información.
En el estado inicial, la línea de Tx está en reposo (Equivalente al símbolo “1” o si utilizamos codificación NRZ su equivalente “-V”)
El primer bit de cada carácter es un bit de comienzo (Símbolo “0” o en NRZ +V) que alerta al receptor sobre la llegada del carácter.
El receptor muestrea cada bit del carácter y busca el comienzo del siguiente. Esta técnica no funciona correctamente para bloques de datos excesivamente largos debido a que el reloj del receptor podría perder el sincronismo respecto del emisor.



(a) Formato de un carácter



(b) Cadena de caracteres de 8 bits asíncrona

Cada carácter tiene un bit de comienzo, 5 a 8 bits de datos, un bit de paridad y uno o dos bits de parada (Mismo estado de reposo, ej NRZ)

El **bit de paridad** indica la paridad de “unos” incluido el bit de paridad, puede ser Par, Impar o no especificarse (No se utiliza este bit).

Dado que, al menos, 2 de cada 8 bits, en este ejemplo, no tienen información la tasa de efectividad de información por bits transmitidos es 80%, para mejorarla se pude aumentar los bits (Aumenta la probabilidad de error de sincronismo) o bien se utiliza método de transmisión sincrónica.
Resumen Tx asíncrona:

- En estado de reposo, el receptor buscará una transición de 1 a 0 como inicio de la trama
- Muestrea los intervalos siguientes (típicamente 7).
- Se vuelve a sincronizar con la siguiente transición de caracteres de 1 a 0.
- Sencilla de implementar.
- De bajo coste.
- Ineficiente, baja tasa de envio de datos vs datos totales enviados. 80% aprox. o peor si se usan menos bits de datos.
- Adecuado para transmisiones con pocos datos y baja velocidad de transmisión (teclado, sensores).

Transmisión Síncrona

En esta solución, cada bloque de datos de bits se transmite como una cadena estacionaria sin utilizar códigos de comienzo o parada.
Se puede sincronizar Tx y Rx mediante uso de línea/cable externo (Útil en distancias cortas) o enviar el sincronismo en la propia señal de datos, por ejemplo, la obtenida con el código **Manchester**.
Esto es útil para el envío de bloques con grandes cadenas de bits.

Para que el receptor pueda determinar dónde está el comienzo y el final de cada bloque de datos, cada bloque comienza con un patrón de bits denominado ***preámbulo*** y termina con un patrón de bits denominado ***final***.

Al conjunto de bits de inicio, bits de control, datos y final, se lo llama **TRAMA.**

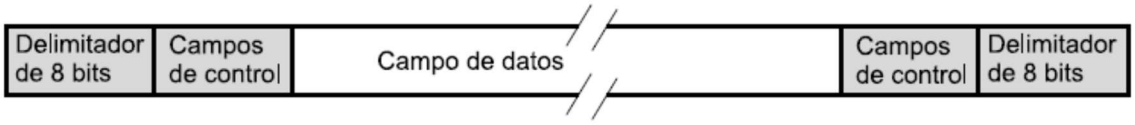


Figura 6.2. Formato de una trama síncrona.

La transmisión síncrona es mucho más eficiente que la asíncrona. La información de control, el preámbulo y el final son normalmente menos de 100 bits.
Por ejemplo, si se definen 48 bits de control (preámbulo y final) y el bloque de datos es de 1.000 caracteres de 8 bits, la eficiencia será Ef= Datos/(Preámbulo + datos), (1000*8)/(48+1000*8) = 99,4%

Tipos de errores

Se dice que ha habido un error cuando se altera un bit. Es decir, cuando se transmite un 1 binario y se recibe un 0, o cuando se transmite un 0 binario y se recibe un 1.
Existen dos tipos de errores: errores aislados o errores a ráfagas. Los primeros corresponden con eventualidades que alteran a un solo bit, sin llegar a afectar a los vecinos. Por el contrario, se dice que ha habido una ráfaga de longitud B cuando se recibe

una secuencia de B bits en la que el primero, el último y cualquier número de bits intermedios son erróneos.
Téngase en cuenta que los efectos de una ráfaga serán siempre mayores cuanto mayor sea la velocidad de transmisión.

Detección de Errores

En todo sistema de transmisión habrá ruido. El ruido dará lugar a errores que modificarán uno o varios bits de la trama.
Se definen las siguientes probabilidades para los posibles errores en las tramas transmitidas:

Pb: Probabilidad de que un bit recibido sea erróneo, también se denomina tasa de error por bit (BER, Bit Error Rate).

P1: Probabilidad de que una trama llegue sin errores.

P2: Probabilidad de que, utilizando un algoritmo para la detección de errores, una trama llegue con uno o más errores no detectados.

F es el número de bits por trama

Para calcular las probabilidades se supondrá que todos los bits tienen una probabilidad de error (Pb) constante e independiente. Entonces se tiene que:

$$P1=(1 - P_b)^F$$
$$P2=1 - P1$$

Ejemplo: Pb (BER) = 10^{-6}
F=1024 bits
$P1= (1-10^{-6})^{1024} = (0,999999)^{1024} = 0,9989 => 99,89\%$

La **detección de errores** se lleva a cabo calculando un código en función de los bits de entrada. El código se añade a los bits a transmitir. Para comprobar si ha habido errores, el receptor calcula el código en función de los bits recibidos y lo compara con el código recibido.

Todas se basan en el siguiente diagrama:

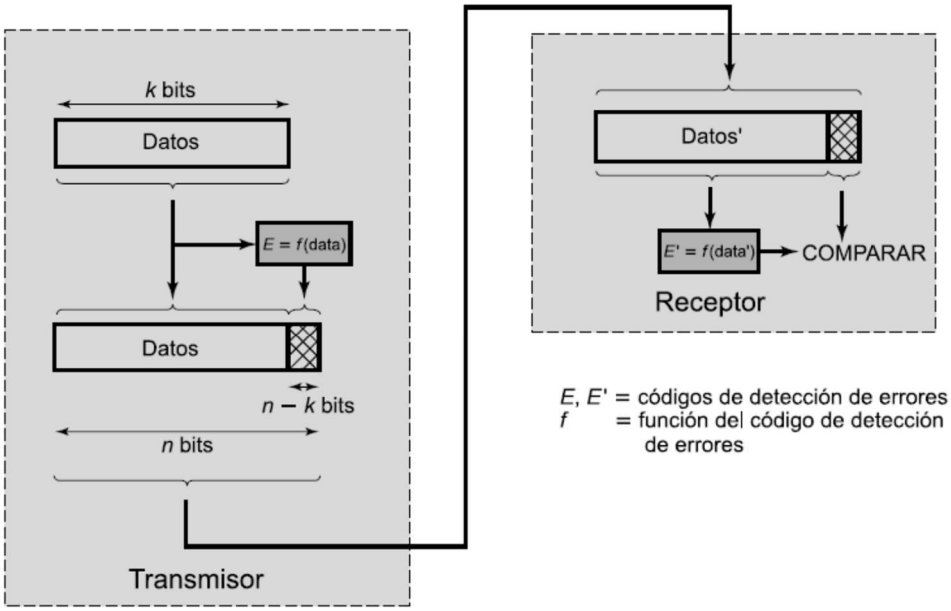
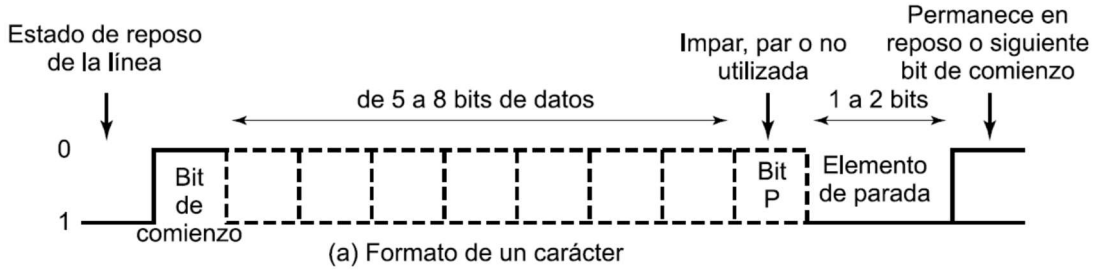


Figura 6.3. Procedimiento para detectar errores.

El código de detección de errores, también llamado bits de comprobación, se añade al bloque de datos para generar la trama de n bits de longitud, la cual será posteriormente transmitida. El receptor separará la trama recibida en los k bits de datos y los (n.k) bits correspondientes al código de detección de errores. El receptor realizará el mismo cálculo sobre los bits de datos recibidos y comparará el resultado con los bits recibidos en el código de detección de errores. Se detectará un error si, y solamente si, los dos resultados mencionados no coinciden

Comprobación de Paridad

El método más sencillo es la comprobación de paridad, el Bit de paridad indica la cantidad de “unos” incluido el bit de paridad, puede ser Par, Impar o no especificarse.



(a) Formato de un carácter

Nótese, no obstante, que si dos (o cualquier número par) de bits se invierten debido a un error, aparecerá un error no detectado. La utilización de bits de paridad no es infalible, ya que los impulsos de ruido son, a menudo, lo suficientemente largos como para destruir más de un bit, especialmente a velocidades de transmisión altas.
Ejemplo: Para una secuencia de datos 1100101, si se define el BP como par el BP=0 transmitiéndose 1100101**0** y si se define como impar el BP=1 transmitiéndose 1100101**1**

Comprobación de errores cíclica (CRC)

Cyclic Redundancy Check, Dado un bloque o mensaje de *k*-bits, el transmisor genera una secuencia de (*n-k*) bits, denominada secuencia de comprobación de la trama (FCS, *Frame Check Sequence*), de tal manera que la trama resultante, con *n* bits, sea divisible por algún número predeterminado.
El receptor dividirá la trama recibida entre ese número y si el resto en la división es cero, supondrá que no ha habido errores.

Corrección de errores

La **corrección de errores** opera de forma similar a la detección de errores, pero en este caso será posible corregir ciertos errores en la secuencia de bits recibida.
¿Cómo es posible que el decodificador corrija los bits erróneos? Esencialmente, la corrección de errores funciona añadiendo redundancia al mensaje transmitido. La redundancia hace posible que el receptor deduzca cuál fue el mensaje original, incluso para ciertos niveles de la tasa de bits erróneos.

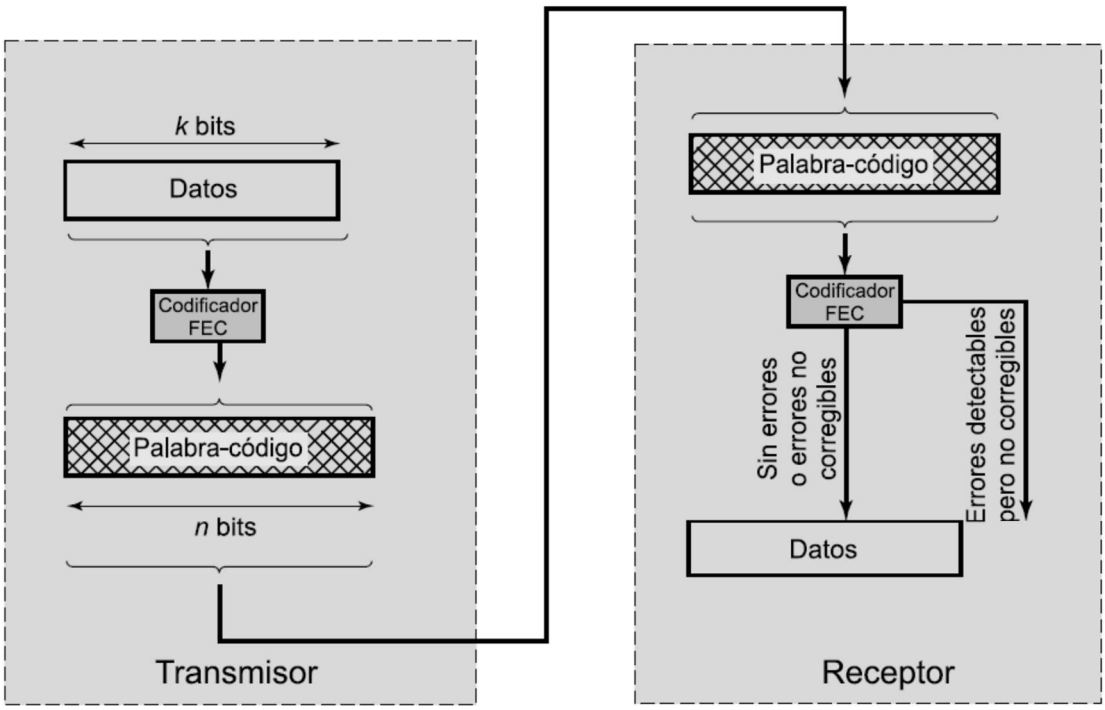


Figura 6.7. Procedimiento para corregir errores.

En el emisor, se utiliza un codificador con corrección de errores hacia delante **FEC (Forward Error Correction).**
Para cada bloque de datos de k bits se genera uno de n bits (n>k) denominado palabra-código, básicamente se agrega redundancia a los datos transmitido.

En el receptor, la señal de entrada se obtiene una cadena de bits similar a la palabra-código original, pero posiblemente con errores. Este bloque se pasa al decodificador FEC, el cual generará una de las siguientes cuatro salidas:

1. Si no ha habido errores, la entrada al decodificador FEC es idéntica a la palabra-código original, por lo que el decodificador generará el bloque de datos original.
2. Para ciertos patrones de error, es posible que el decodificador detecte y corrija esos errores.
   Por tanto, aunque los bloques de datos recibidos difieran de la palabra-código transmitida, el decodificador FEC será capaz de asociar el bloque recibido al bloque de datos original.
3. Para ciertos patrones de error, el decodificador podrá detectarlos, pero no corregirlos. En este caso, el decodificador simplemente informará sobre la detección de un error irrecuperable.
4. Para ciertos, aunque raros, patrones de error, el decodificador no detectará la ocurrencia de dichos errores y asignará el bloque de datos recibido, de n bits, a un bloque de k bits que será distinto al bloque original de k bits.

En algunas situaciones de alta tasa de errores o altos retardos de transmisión, como en la transmisión inalámbrica, suele ser más apropiado no utilizar la corrección de errores y retransmitir la trama.

PROTOCOLO DE CONTROL DEL ENLACE DE DATOS

Las técnicas de sincronización y gestión de la interfaz resultan insuficientes para detectar y corregir errores en una transmisión. Es necesario, por tanto, incluir en una capa de control que regule el flujo de información, además de detectar y controlar los errores. Esta capa se denomina protocolo de **control del enlace de datos**.
El control de flujo permite al receptor regular el flujo de los datos enviados por el emisor, de manera que la memoria temporal del receptor no se desborde.
En un protocolo de control del enlace de datos, el **control de errores** se lleva a cabo mediante la **retransmisión** de las tramas dañadas que no hayan sido confirmadas o de aquellas para las que el otro extremo solicite su retransmisión.

En resumen, la necesidad del control del enlace de datos tiene los siguientes objetivos:

- Sincronización de trama: los datos se envían en bloques denominados tramas, cuyo principio y fin deben ser identificables.

- **Control de flujo**: la estación emisora no debe enviar tramas a una velocidad superior a la que la estación receptora pueda recibirlas.

- **Control de errores**: se debe corregir cualquier error en los bits provocado por el sistema de transmisión.

- Direccionamiento: en una línea multipunto, como por ejemplo una red de área local (LAN), se debe identificar a las dos estaciones involucradas en una transmisión.

- Datos y control sobre el mismo enlace: por lo general, no se desea tener un canal de comunicaciones independiente para la información de control. En consecuencia, el receptor deberá ser capaz de diferenciar entre la información de control y los datos.

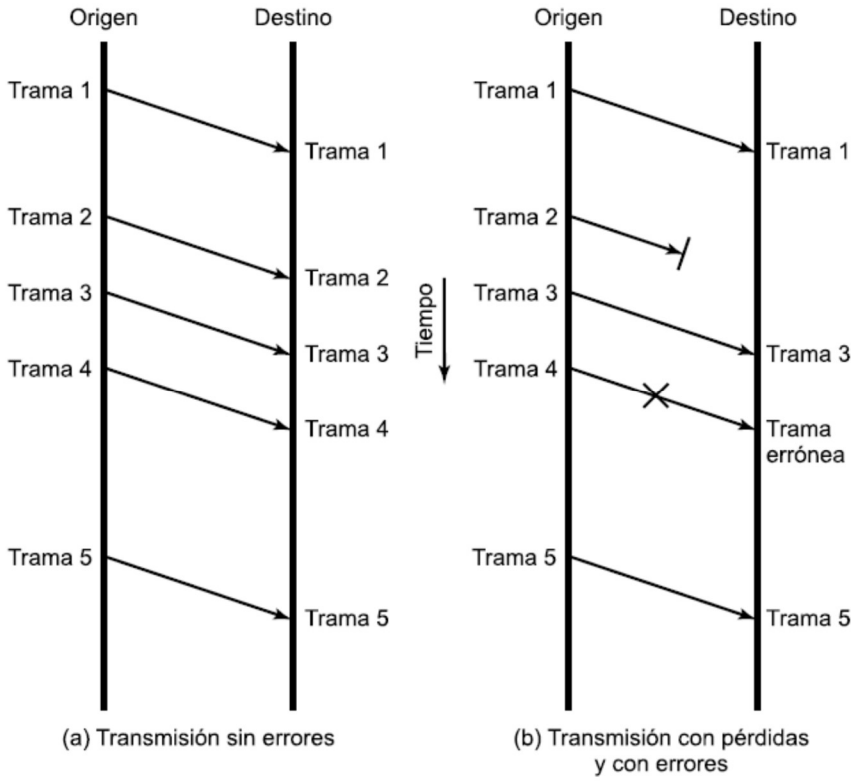
- Gestión del enlace: el inicio, mantenimiento y finalización de un intercambio de datos precisa un alto grado de coordinación y cooperación entre las estaciones. Se necesitan, pues, una serie de procedimientos para llevar a cabo la gestión de este intercambio.

Control de flujo

El control de flujo es una técnica utilizada para asegurar que una entidad de transmisión no sobrecargue a la entidad receptora con una excesiva cantidad de datos.
En ausencia de procedimientos para el control de flujo, la memoria temporal del receptor se podría llenar y **desbordarse** mientras éste se encuentra procesando datos previos.

Tiempo de transmisión: Tiempo para transmitir todos los bits en el medio de transmisión.
Tiempo de Propagación: Tiempo que demora un bit en atravesar el medio, es decir llegar a destino.

Ejemplo de transmisión en ausencia de errores



Control de flujo mediante parada y espera

Una entidad origen transmite una trama. Tras la recepción,la entidad destino indica su deseo de aceptar otra trama mediante el envío de una confirmación de la trama que acaba de recibir (Acknowledgement ACK). El origen debe esperar a recibir la confirmación antes de proceder a la transmisión de la trama siguiente. De este modo, el destino puede parar el flujo de los datos sin más que retener las confirmaciones.
Este control es adecuado para pocas tramas y de corta longitud, que es limitada por el buffer del receptor y la velocidad de propagación de los bits.

Tanto para tramas muy largas o velocidades de transmisión elevadas, aumenta el % de error por lo que se retransmiten las tramas y aumenta la ineficiencia del canal.

Ventana deslizante

Para mejorar la eficiencia de transmisión, se establece un protocolo que permite transmitir varias tramas simultáneamente sin esperar un ACK.
La memoria temporal del receptor (Rx) determina la cantidad de tramas (N) simultáneas en tránsito a la espera del ACK. Por lo que el Tx puede enviar hasta (N) tramas sin recibir la confirmación.
Cada trama es etiquetada con un numero de secuencia, esto permite confirmar varias tramas.
El ACK que envía el Rx contiene el número de la Trama que espera recibir RRn (Receive Ready), dando el OK a la trama n-1.

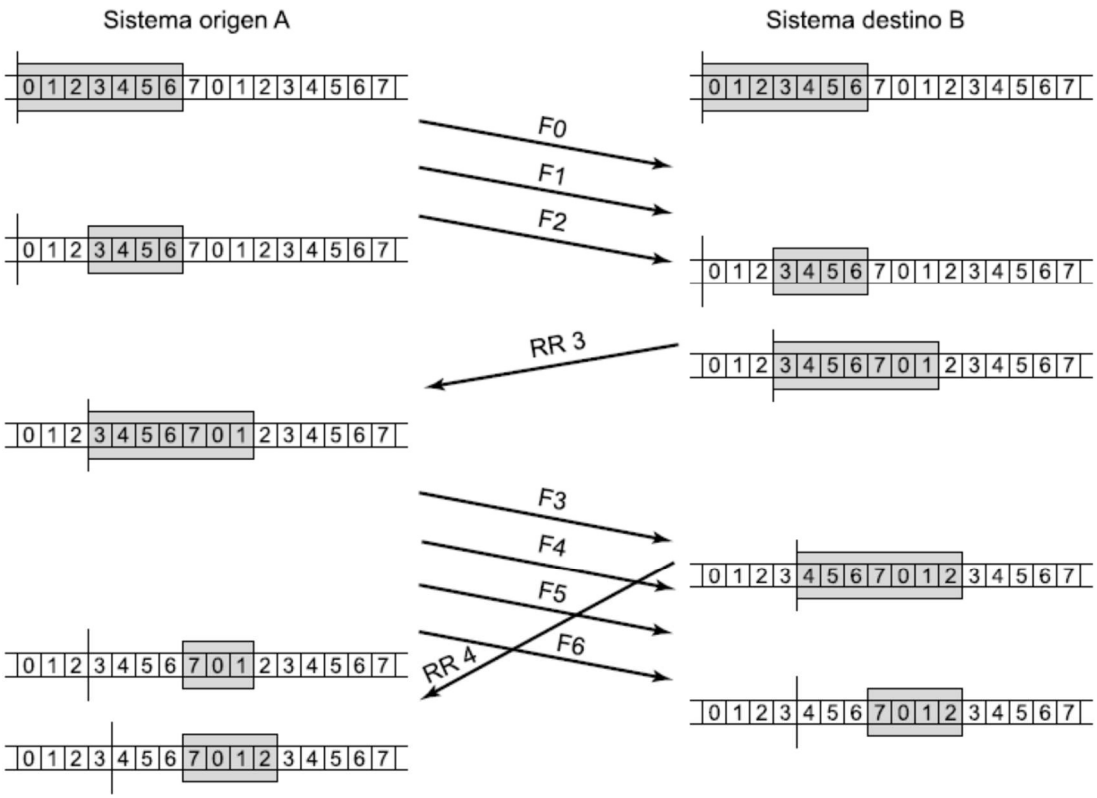


Figura 7.4. Ejemplo de transmisión mediante el protocolo de ventana deslizante.

Control de errores

El control de errores hace referencia a los mecanismos necesarios para la detección y la corrección de errores que aparecen en una transmisión de tramas.
Estos mecanismos se denominan genéricamente solicitud de repetición automática (**ARQ, Automatic Repeat reQuest**); el objetivo de un esquema ARQ es convertir un enlace de datos no fiable en fiable. Hay tres variantes ARQ estandarizadas:

- ARQ con parada y espera.
- ARQ con vuelta atrás N.
- ARQ con rechazo selectivo.

ARQ con parada y espera.

El esquema ARQ con parada y espera se basa en la técnica para el control de flujo mediante parada y espera estudiada para el control de flujo.
La estación origen transmite una única trama y debe esperar la recepción de una confirmación (ACK). No se podrá enviar ninguna otra trama hasta que la respuesta de la estación destino llegue al emisor.

Si la trama llega con errores al Rx, no se envía ACK, por lo tanto, el Tx no recibirá una confirmación para envio de una nueva trama. El Tx utiliza un temporizador, y si no recibe el ACK antes de que expire, reenviara la última trama.
Pueden suceder tres variantes para que el Tx reenvíe una trama por expiración del timer:

1. La trama llegue dañada al RX y la descarta. No hay envio de ACK
2. La trama no llega al RX, por lo tanto, no hay envio de ACK
3. El ACK no llega al TX o llega con errores

Para la situación 3, (se daña el ACK), TX retransmitiría nuevamente la última trama y el RX aceptaría dos tramas iguales como si fueran distintas. Para solucionarlo el RX utiliza ACK0 y ACK1 alternadamente al igual que el TX identifica alternadamente con 0 y 1 las tramas que envía.

Este es muy simple pero ineficiente.

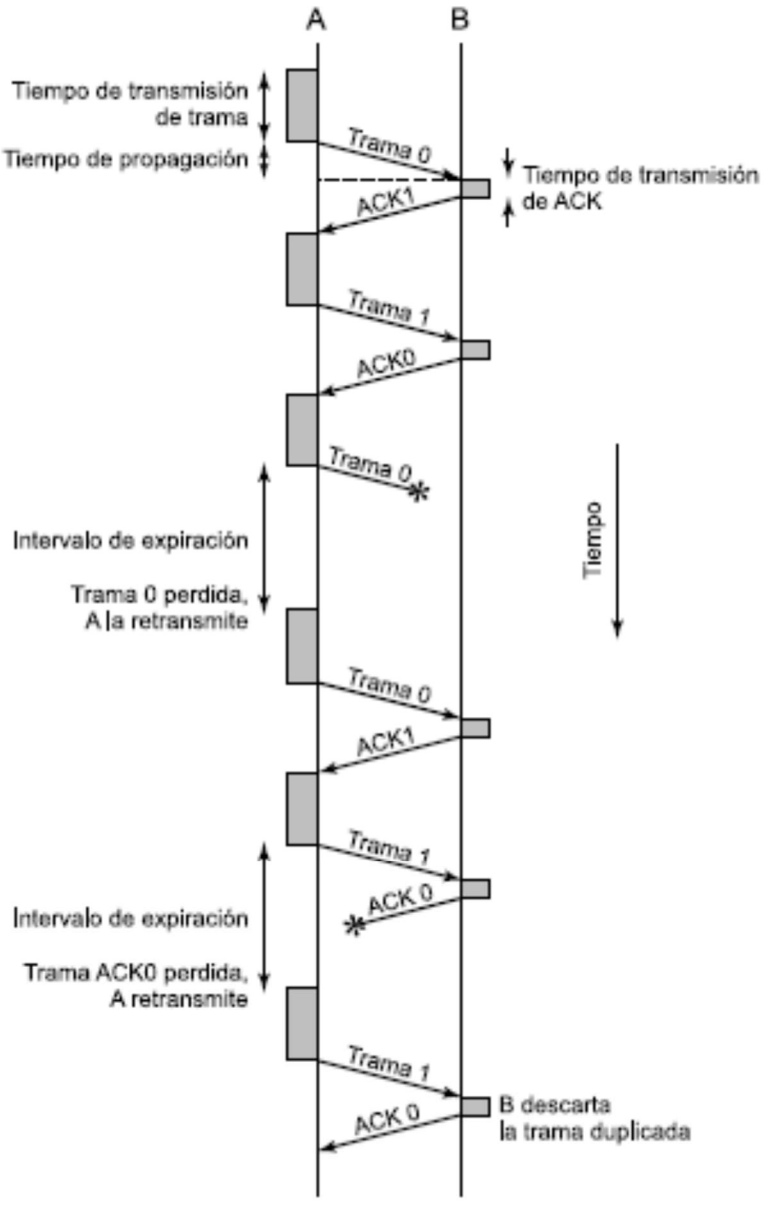


Figura 7.5. ARQ con parada y espera.

ARQ con vuelta atrás N.

Está basado en el control de flujo por ventana deslizante. Mientras no aparecen errores, el destino confirmará las tramas recibidas. Utiliza envio de RRn (Receiver Ready) mientras no reciba tramas con errores.
Si detecta una trama con error, el Rx envía una confirmación negativa REJ (Reject) y descarta todas las tramas que reciba con posterioridad hasta que la trama errónea se reciba correctamente.
Por su parte el Tx debe renviar la trama errónea y todas las tramas **posteriores**.

ARQ con rechazo selectivo

En el esquema ARQ con rechazo selectivo, o también denominado de retransmisión selectiva, solo se retransmiten las tramas para las que se recibe una confirmación negativa, denominada SREJ (Selective REJect).
El RX acepta las tramas posteriores a la rechazada y las almacena una su memoria temporal hasta recibir nuevamente la trama rechazada.
Este sistema minimiza las retransmisiones, sin embargo, tanto el Receptor como el Transmisor deberán tener memorias de mayor tamaño y lógicas de comunicación más complejas.

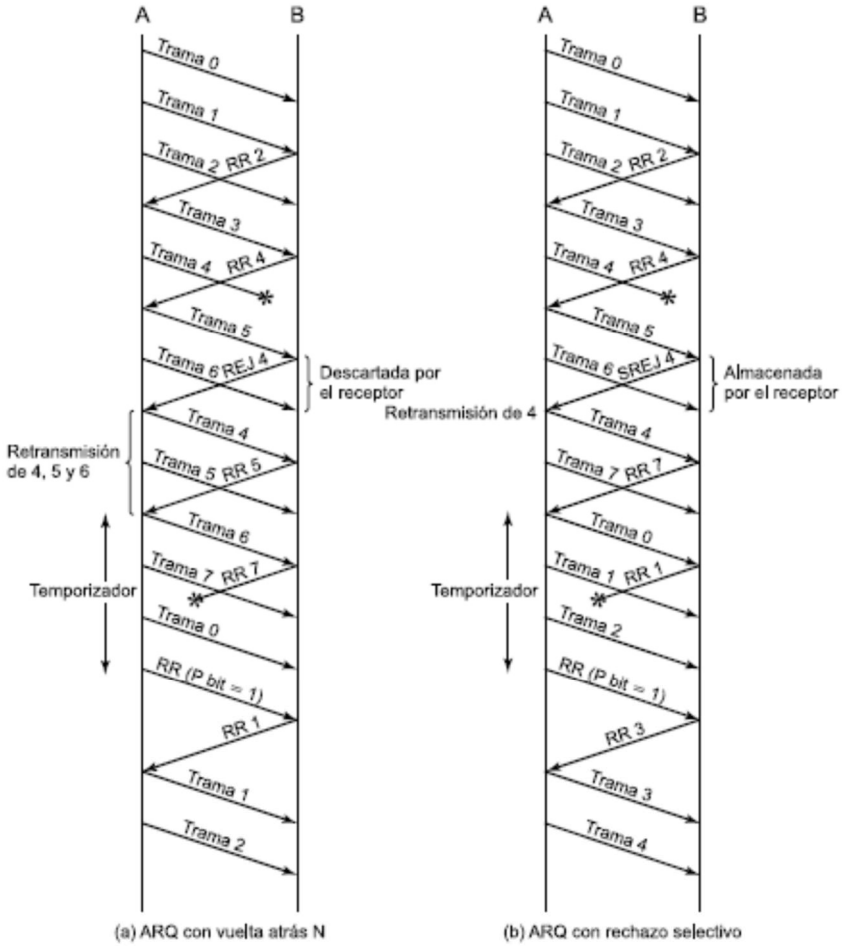


Figura 7.6. Protocolos ARQ mediante ventana deslizante.

Protocolos de comunicación digital

Es evidente que debe haber un alto grado de cooperación entre los extremos involucrados en la transmisión de datos, sin embargo, sería muy complejo implementar esta lógica en un solo modulo, por lo que se subdivide la tarea en capas.
Por lo general, las funciones más básicas se dejan a la capa inmediatamente inferior y 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.
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 intercambien información verificando 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. (Formato de los datos, nivel de señal)
- La semántica: incluye información de control para la coordinación y la gestión de errores. (Información de control, manejo de errores)
- La temporización: considera aspectos relativos a la sintonización de velocidades y secuenciación. (Sintonización de velocidad, secuenciación)

Arquitectura de protocolo

Establecido el concepto de protocolo y su arquitectura de módulos apilados (Stack), analicemos diferentes opciones de implementación de dicha arquitectura.

Arquitectura simple

Las comunicaciones involucran a tres agentes: aplicaciones, computadores 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 hacerlos llegar a la aplicación destino en el computador remoto.

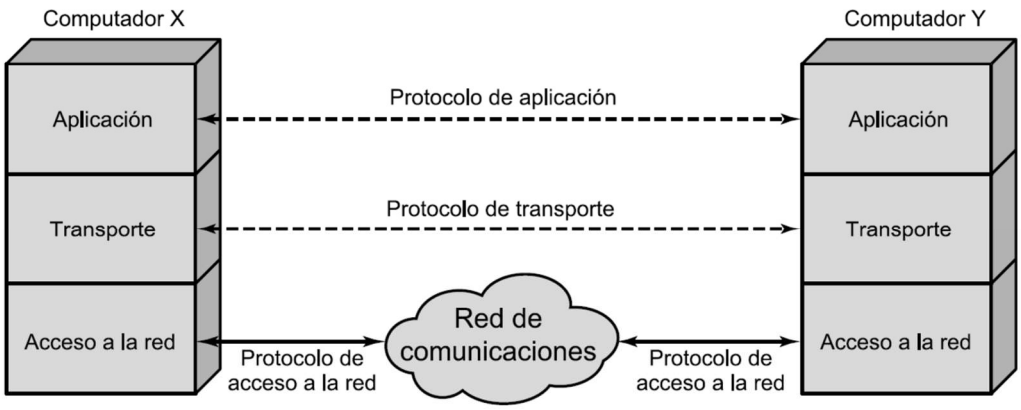


Figura 2.3. Protocolos en una arquitectura simplificada.

Características de la capa de acceso a la red

- El emisor proporciona la dirección de la red de destino.
- Adapta el mensaje según la red que se utilice.
- Dirección única para cada elemento.

Características capa de transporte

- Intercambio de datos de manera segura
- Independiente de la red que se utilice
- Independiente de las aplicaciones.

Características capa de aplicación

- Establecer la lógica para diferentes aplicaciones de usuario

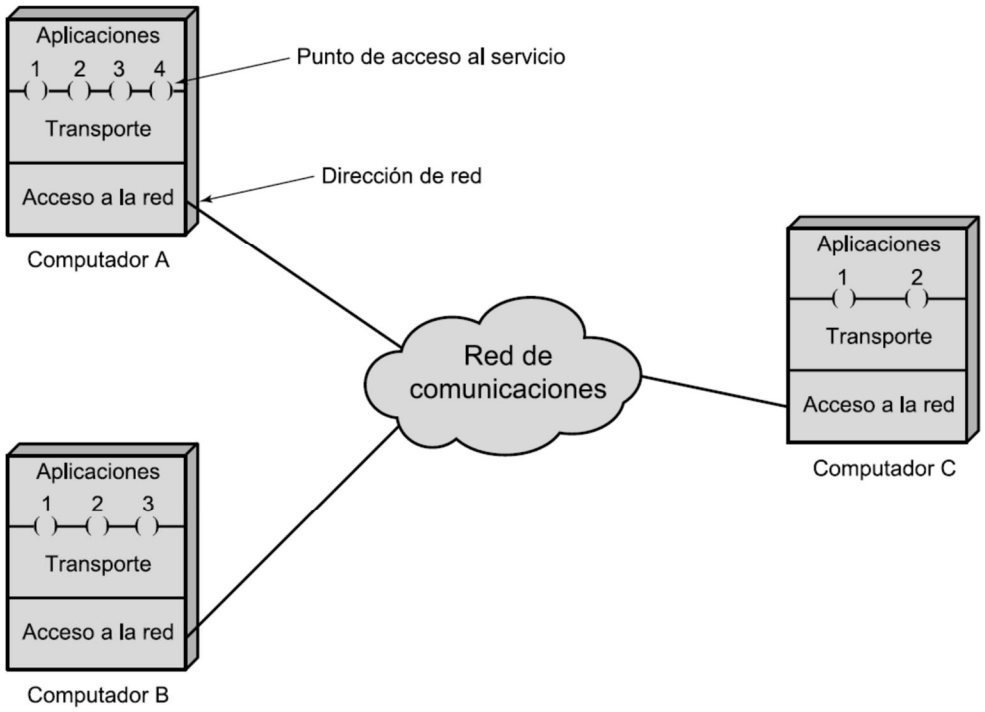


Figura 2.2. Redes y arquitecturas de protocolos.

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

Arquitectura OSI

La principal motivación para el desarrollo del modelo OSI fue proporcionar un modelo de referencia para la normalización. Dentro del modelo, en cada capa se pueden desarrollar uno o más protocolos. El modelo define en términos generales las funciones que se deben realizar en cada capa y simplifica el procedimiento de la normalización.

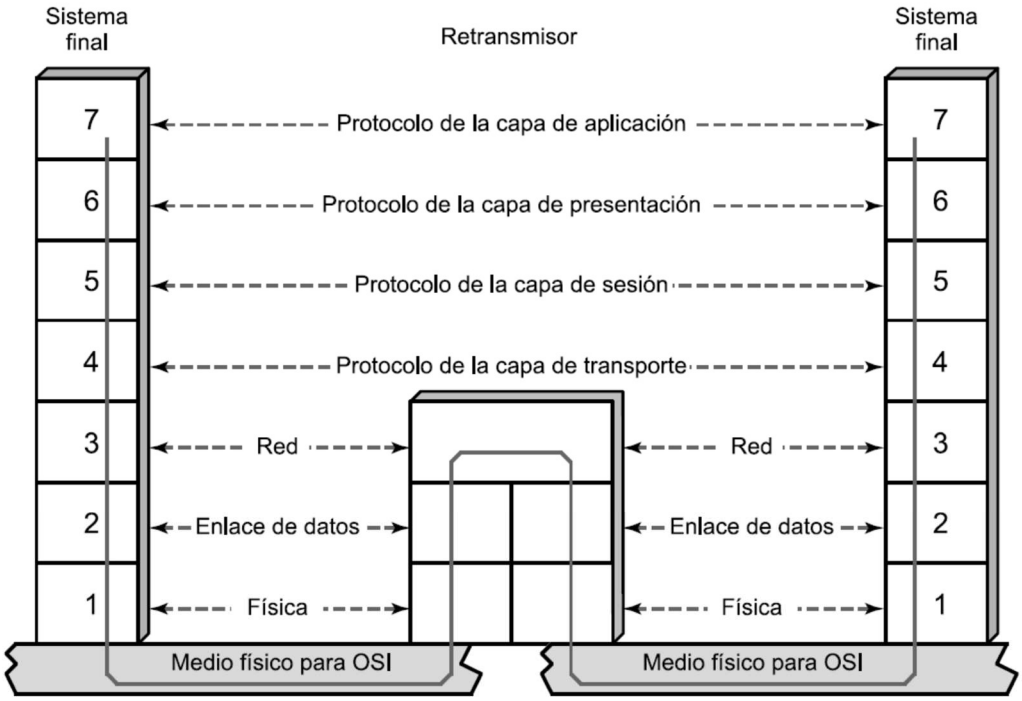


Figura 2.11. Utilización de un retransmisor.

Capa de Física

La capa física se encarga de la interfaz física entre los dispositivos. Además, define las reglas que rigen en la transmisión de los bits.

La capa física tiene cuatro características importantes:

- Mecánicas
- Eléctricas
- Funcionales
- De procedimiento

Capa de enlace de datos

La principal función es que la capa física sea fiable, para ello implementa la detección y corrección de errores.

Capa de Red

Esta capa se encarga de realizar la transferencia de información entre los sistemas finales, a través de un sistema de comunicación. Establecerá la dirección de destino y las prioridades de los mensajes.

Capa de transporte

Esta capa proporciona un mecanismo para intercambiar datos entre sistemas finales, asegurando que los datos se entreguen libre de errores, en orden, sin perdidas ni duplicaciones.

Capa de sesión

La capa de sesión proporciona los mecanismos para controlar el diálogo entre las aplicaciones de los sistemas finales. En muchos casos, los servicios de la capa de sesión son parcialmente, o totalmente prescindibles.

La capa de sesión proporciona los siguientes servicios:

- Control del dialogo. Ej Half o full duplex
- Agrupamiento
- Recuperación.

Capa de presentación

La capa de presentación define la sintaxis utilizada entre las entidades de aplicación y proporciona los medios para seleccionar y modificar la representación utilizada. Algunos ejemplos de servicios específicos que se pueden realizar en esta capa son los de compresión y cifrado de datos.

Capa de aplicación

A esta capa pertenecen las funciones de administración y los mecanismos genéricos necesarios para la implementación de aplicaciones distribuidas. Además, en esta capa también residen las aplicaciones de uso general como, por ejemplo, la transferencia de archivos, el correo electrónico y el acceso desde terminales a computadores remotos, entre otras.

Arquitectura TCP/IP

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.

El modelo TCP/IP estructura el problema de la comunicación en cinco capas relativamente independientes entre sí:

- Capa física.
- Capa de acceso a la red.
- Capa internet.
- Capa de transporte.
- Capa de aplicación.

Capa física

Esta capa se encarga de la especificación de las características del medio de transmisión, la naturaleza de las señales, la velocidad de datos.

Capa de acceso a la red

Esta capa es responsable del intercambio de datos entre el sistema final (servidor, estación de trabajo, etc.) y la red a la cual está conectado. El emisor debe proporcionar a la red la dirección del destino, de tal manera que ésta pueda encaminar los datos hasta el destino apropiado.

Capa de internet

El protocolo internet (IP, Internet Protocol) se utiliza para ofrecer el servicio de encaminamiento (ruteo) a través de varias redes. Este protocolo se implementa tanto en los sistemas finales como en los routers intermedios. Su función es interconectar redes diferentes siguiendo la ruta adecuada.

Capa de transporte

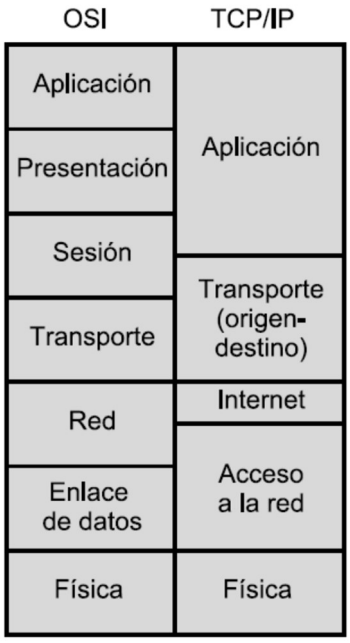
Esta capa proporciona un mecanismo para intercambiar datos entre sistemas finales.
En la arquitectura de protocolos TCP/IP se han especificado dos protocolos para la capa de transporte:

- El orientado a conexión, TCP (Transmission Control Protocol), asegurando que los datos se entreguen libre de errores, en orden, sin perdidas ni duplicaciones.
- El no orientado a conexión UDP (User Datagram Protocol). No asegura la entrega de datos, la complejidad es mínima, solo identificar puertos.

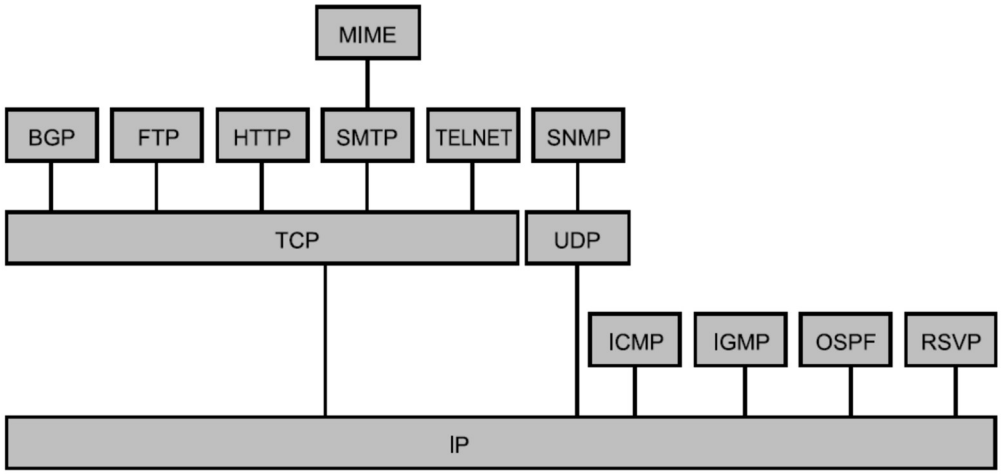
Capa de aplicación

Esta capa contiene toda la lógica necesaria para posibilitar las distintas aplicaciones de usuario. Para cada tipo particular de aplicación, como por ejemplo, la transferencia de archivos, se necesitará un módulo bien diferenciado.

Comparación arquitectura OSI vs TCP/IP



Aplicaciones estandarizadas de TCP/IP



BGP (*Border gateway protocol*) = Protocolo de pasarela fronteriza
FTP (*File transfer protocol*) = Protocolo de transferencia de archivos
HTTP (*Hypertext transfer protocol*) = Protocolo de transferencia de hipertexto
ICMP (*Internet control message protocol*) = Protocolo de mensajes de control de Internet
IGMP (*Internet group management protocol*) = Protocolo de gestión de grupos en Internet
IP (*Internet protocol*) = Protocolo Internet
MIME (*Multipurpose internet mail extension*) = Extensiones multipropósito de correo electrónico
OSPF (*Open shortest path first*) = Protocolo del primer camino más corto disponible
RSVP (*Resource reservation protocol*) = Protocolo de reserva de recursos
SMTP (*Simple mail transfer protocol*) = Protocolo simple de transferencia de correo electrónico
SNMP (*Simple network management protocol*) = Protocolo simple de gestión de red
TCP (*Transmission control protocol*) = Protocolo de control de transmisión
UDP (*User datagram protocol*) = Protocolo de datagrama de usuario

Figura 2.15. Algunos protocolos en la familia de protocolos TCP/IP.