

Capa de enlace

Definiciones



Es la capa 2 de la arquitectura OSI.

Se encarga de conseguir que la comunicación de datos se produzca correctamente a través de un medio físico de transmisión.

Para lograr que dos dispositivos adyacentes se comuniquen, se necesita un control del intercambio de datos: el control del enlace.

La capa de enlace proporciona a la capa de Red un servicio de transporte de bits fiable (asegura que los bits se transmiten correctamente por el medio físico).

El bloque de datos transmitido se denomina TRAMA.

Capa de enlace

Funciones

Sincronización a nivel de trama.

Control de flujo: las estaciones deben ponerse de acuerdo en el ritmo de transmisión de datos.

Control de errores: los enlaces no son perfectos. Hay que controlar que no haya errores en la transmisión.

Direccionamiento: si hay varios posibles destinos, es necesario identificar a quien va dirigida la trama.

Gestión del enlace:

- Inicio de la transmisión
- Mantenimiento de la transmisión
- Finalización de la transmisión



Capa de enlace

Arquitectura IEEE 802

Recuerda que en la arquitectura IEEE 802, el nivel de enlace se divide en dos subcapas:

- LLC: se encarga de las funciones comunes de la capa independientemente del medio físico usado (ej: control de errores). Sus funciones han sido definidas por el subgrupo 802.2.
- MAC: se encarga del acceso al medio.

En esta presentación nos ocuparemos de algunas de las funciones definidas en la subcapa LLC.

Vamos a ello





Capa de enlace

Tipos de servicios

Sin conexión – No fiable

- No se establece conexión.
- Se envía información y no se espera confirmación.
- Nivel físico muy fiable.
- Información en tiempo real.
- Rapidez y sin retrasos.

Sin conexión – Fiable

- Por cada trama enviada el receptor “avisa” al emisor que ha llegado con éxito.
- El emisor se asegura que la trama ha llegado correctamente.

Orientado a conexión – Fiable

- Se establece una conexión antes de iniciar cualquier intercambio de datos.
- Es el más seguro y complejo de implementar.
- Se lleva a cabo en tres fases:
 - Inicio de conexión. Se definen contadores de trama. Se reserva memoria para las retransmisiones.
 - Envío de datos.
 - Terminación de la conexión y liberación de memoria.



Capa de enlace

Tramas



Una trama es un bloque de bits agrupados que son enviados por la línea.

El tamaño de la trama depende del tipo de red.

Agrupar los bits en tramas facilita la detección y corrección de errores y mejora la compartición del medio.

En general, una trama se compone de tres partes

- Información sobre la trama
- Datos.
- Redundancia.



Capa de enlace

Control de flujo

El control de flujo consiste en asegurarse que el emisor no sature o sobrecargue al receptor con una cantidad de datos excesiva

Típicamente el receptor almacena la información recibida en una memoria temporal (buffer).

El receptor puede así comprobar que la información es correcta y gestionar el orden de las tramas.

Si la información es correcta se transfiere a la capa de red.

El control de flujo impide que el buffer en el receptor se llene (porque el receptor no consigue procesar la información de forma suficientemente rápida).



Capa de enlace

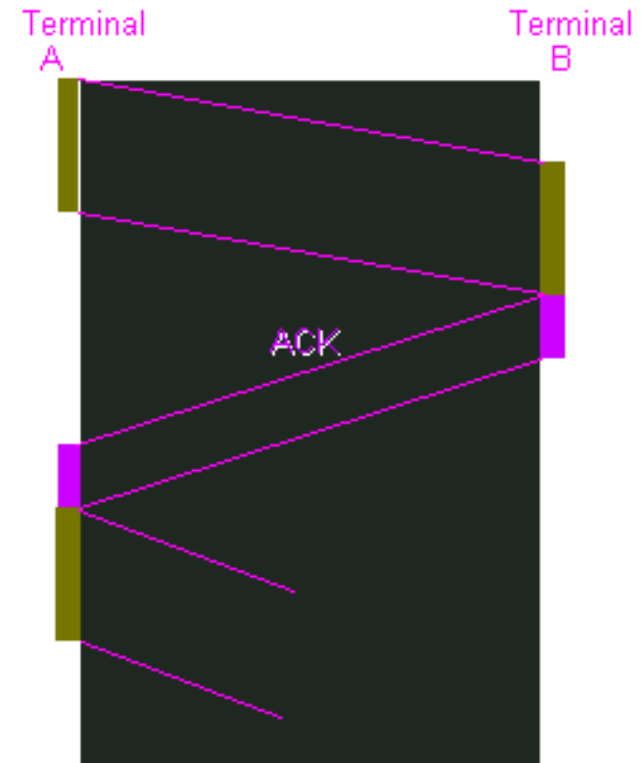
Control de flujo de parada y espera

Parada y espera

- El emisor, antes de enviar la siguiente trama, espera a que el receptor mande una confirmación de que ha llegado la trama anterior.

Pros y contras.

- Bueno para pocas tramas y de gran tamaño.
- El emisor puede pasar tiempo sin hacer nada esperando al ack.
- Este protocolo necesita líneas bidireccionales (duplex o semiduplex)



No se va a ver la gestión de errores.

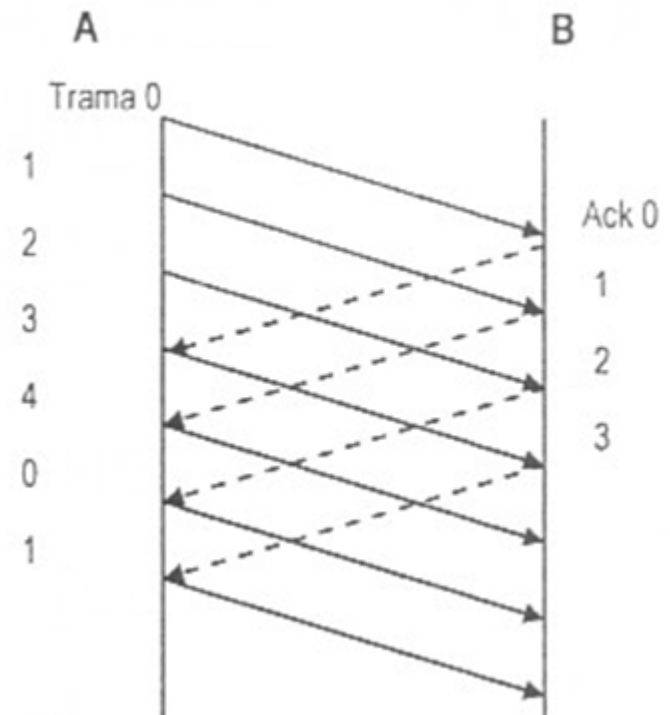


Capa de enlace

Control de flujo con ventana deslizante

Ventana deslizante

- Nace para mejorar la baja eficiencia de los protocolos de parada y espera.
- No se espera el ACK antes de enviar la siguiente trama.
- **Tamaño de la ventana:** número máximo de tramas que se pueden enviar sin que se hayan recibido ACKs previos.



Capa de enlace

Control de flujo. Piggybacking

En las comunicaciones reales, lo más común es que se envíen datos en las dos direcciones.

Es posible combinar la transmisión de una señal de ACK para una trama anterior con la transmisión (de receptor a emisor) de datos.

Se necesita línea duplex.

No se va a tratar la gestión de errores.



Capa de enlace

Control de errores.

Consiste en enviar algunos bits añadidos a los datos con información que permita detectar o corregir los errores.

El porcentaje de redundancia se calcula como $(\text{Numero de bits de control} / \text{bits totales}) * 100$.

Existe una amplia bibliografía relativa al control de errores, aquí veremos algunos ejemplos por encima.



Capa de enlace

Control de errores. Algunos tipos

ECO. El receptor envía una copia exacta de la información recibida al emisor.

Paridad lineal. Se añade un bit extra, indicando si el número de bits con valor a 1 es par o impar.

Paridad de bloque. Se distribuyen los datos en una tabla y se calcula paridad por cada línea y columna.

Para el mensaje: 1100101 0110110 1011010
1001111 0111001 1100111 1010000

la tabla sería:


1	1	0	0	1	0	1	0
0	1	1	0	1	1	0	0
1	0	1	1	0	1	0	0
1	0	0	1	1	1	1	1
0	1	1	1	0	0	1	0
1	1	0	0	1	1	1	1
1	0	1	0	0	0	0	0
1	0	0	1	0	0	0	0

Capa de enlace

Actividades

Calcular la paridad bidimensional del siguiente mensaje:
1001101, 1111010, 1100110, 1110001, 1101001, 1110111,
0010111





Capa de enlace


Control de errores. CRC

Al principio de la comunicación, emisor y receptor acuerdan un Polinomio Generador.

Al iniciar la transmisión se añaden un número predeterminado de ceros a la información a enviar y se divide utilizando el polinomio generador.

El receptor realiza nuevamente una división sobre los datos recibidos y si el resto es 0 indica que la trama se ha recibido sin errores.

Finalmente se descartan los bits añadidos en el transmisor para quedarnos con el mensaje original.



Capa de enlace

Control de errores. CRC - Ejemplo

OPERACIONES EN EL EMISOR

- Mensaje original a enviar: 100101
- Polinomio Generador (uno de ellos): $x^2 + 1. \Rightarrow 101$
- Se añaden dos ceros al mensaje: 10010100
- Dividir 10010100 entre 101. Da como resto 11
- Mensaje enviado: 1001010011

OPERACIONES EN EL RECEPTOR

- Si como consecuencia del paso por el medio físico, el mensaje recibido no tiene errores será: 1001010011.
- Se divide 1001010011 entre el polinomio generador (101).
Como el resto de esta operación es 0 se concluye que no hay errores.
- El receptor elimina los dos bits extra y pasa la información a las capas superiores: 100101.
- Nota: si el mensaje se ha visto modificado durante su paso por el medio físico, la división en el receptor no tiene 0 como resto.
El receptor concluye en este caso que se ha producido un error.

Capa de enlace

Actividades

Considerando el polinomio generador: $x^2 + 1$. \Rightarrow 101
Indicar la trama que se transmite para los siguientes mensajes:

- 1001010
- 101011101

Comprueba en el receptor si los mensajes recibidos han sido correctos.



Capa de enlace

Corrección de errores - Retransmisión

Es el método de corrección más sencillo.

Se detecta el error y se pide al emisor que vuelva a enviar la trama.

Se tienen que memorizar las tramas enviadas hasta la recepción de un ACK que confirme que el envío de información fue exitosa.





Capa de enlace

Corrección de errores – Código Hamming. Ejemplo 7-4

El código Hamming permite detectar y corregir errores en el equipo receptor.

En función del número de bits redundantes podrán corregirse mayor número de errores. Para el ejemplo aquí mostrado, sólo es posible corregir un error en cada grupo de cuatro bits enviados.

Para conseguirlo debemos incluir tres bits extra.

Capa de enlace

Corrección de errores – Código Hamming. Ejemplo 7-4

OPERACIONES EN EL EMISOR

Consideraremos el envío del mensaje 1 0 1 1 (a3 a2 a1 a0)

Cálculo de los bits extra a enviar:

$$\underline{s0} = (a0 + a1 + a2) \bmod 2 \Rightarrow (1 + 1 + 0) \bmod 2 = \underline{0}$$

$$\underline{s1} = (a1 + a2 + a3) \bmod 2 \Rightarrow (1 + 0 + 1) \bmod 2 = \underline{0}$$

$$\underline{s2} = (a0 + a1 + a3) \bmod 2 \Rightarrow (1 + 1 + 1) \bmod 2 = \underline{1}$$

Mensaje enviado:

$$a3 \ a2 \ a1 \ a0 \ s2 \ s1 \ s0 \Rightarrow 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0$$

Capa de enlace

Corrección de errores – Código Hamming. Ejemplo 7-4

OPERACIONES EN EL RECEPTOR

Mensaje recibido sin errores. Tiene 7 bits y corresponde exactamente con los bits que se enviaron:

$a_3 a_2 a_1 a_0 s_2 s_1 s_0 \Rightarrow 1 0 1 1 1 0 0$ (si la recepción es correcta)

Cálculo de los bits de comprobación:

$$r_0 = (a_0 + a_1 + a_2 + s_0) \bmod 2 \Rightarrow (1 + 1 + 0 + 0) \bmod 2 = \underline{0}$$

$$r_1 = (a_1 + a_2 + a_3 + s_1) \bmod 2 \Rightarrow (1 + 0 + 1 + 0) \bmod 2 = \underline{0}$$

$$r_2 = (a_0 + a_1 + a_3 + s_2) \bmod 2 \Rightarrow (1 + 1 + 1 + 1) \bmod 2 = \underline{0}$$

Valores de los bits $r_2 r_1 r_0$							
000	001	010	011	100	101	110	111
OK	s_0	s_1	a_2	s_2	a_0	a_3	a_1

Capa de enlace

Corrección de errores – Código Hamming. Ejemplo 7-4

OPERACIONES EN EL RECEPTOR

Mensaje recibido con error.:

$a_3 a_2 a_1 a_0 s_2 s_1 s_0 \Rightarrow 1 0 \textcolor{red}{1} 1 1 0 0$ (dato enviado)

$a_3 a_2 a_1 a_0 s_2 s_1 s_0 \Rightarrow 1 0 \textcolor{red}{0} 1 1 0 0$ (dato recibido con error en a_1)

Cálculo de los bits de comprobación:

$$\underline{r_0} = (a_0 + a_1 + a_2 + s_0) \bmod 2 \Rightarrow (1 + 0 + 0 + 0) \bmod 2 = \underline{1}$$

$$\underline{r_1} = (a_1 + a_2 + a_3 + s_1) \bmod 2 \Rightarrow (0 + 0 + 1 + 0) \bmod 2 = \underline{1}$$

$$\underline{r_2} = (a_0 + a_1 + a_3 + s_2) \bmod 2 \Rightarrow (1 + 0 + 1 + 1) \bmod 2 = \underline{1}$$

$\underline{r_2 r_1 r_0} \Rightarrow \underline{111}$. Mirando la tabla se ve que el error esta en a_1

Valores de los bits $r_2 r_1 r_0$							
000	001	010	011	100	101	110	111
OK	s_0	s_1	a_2	s_2	a_0	a_3	a_1

Capa de enlace

Actividades

Utilizando el código Hamming 7-4 para el mensaje 1000 (d3 d2 d1 d0):

Determina, justificando tus respuestas:

- Bits extra enviados en el emisor
- Mensaje enviado.
- Bits de comprobación si todo el mensaje se ha recibido correctamente.
- Bits de comprobación si en la recepción se ha producido un error en el bit d0.

