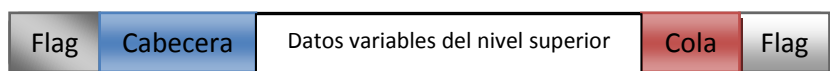


# CAPITULO 9. CONTROL DE ENLACE DE DATOS

- CREACIÓN DE TRAMAS:

- Empaqueta los bits en tramas, de forma que una trama se pueda distinguir de la otra.
- El entramado separa un mensaje desde el origen a un destino, o de otros mensajes a otros destinos, añadiendo las direcciones del emisor y del destinatario.
- Tramas de tamaño fijo:
  - No existe la necesidad de definir las fronteras de las tramas.
  - El tamaño del mismo se puede usar como delimitador.
  - Frecuente en redes de área extendida.
- Tramas de tamaño variable:
  - Frecuente en redes de área local.
  - Es necesario definir el final de una trama y el principio de la siguiente.
  - Protocolos orientados a carácter:
    - Los datos a transportar son caracteres de ocho bits que pertenecen a un sistema de codificación como el ASCII.
    - La cabecera contiene las direcciones de intercambio y otra información de control.
    - La cola lleva bits redundantes para la detección o corrección de errores. Son también múltiplos de ocho.
    - Se añade un flag (un byte) al principio y final de cada trama para delimitarlas.
    - Los flag están formados por caracteres especiales que dependen del protocolo.



- Un problema es que el patrón usado para el flag sea parte de la información.
- Para resolverlo se añadió una estrategia de transparencia a nivel de byte (byte stuffing) al tramado orientado a carácter.
- Byte stuffing es el proceso de añadir un byte extra, denominado carácter de escape (ESC) y que tiene un patrón de bit bien definido, cada vez que hay un carácter de flag o un escape en los datos.



- Protocolos orientados a bit:
  - Similar al protocolo orientado a carácter, pero enviado secuencia de bits.
  - El flag tiene un patrón especial de 8 bits (01111110).
  - Un problema es que el patrón usado para el flag sea parte de la información.
  - Para resolverlo se añadió una estrategia de transparencia a nivel de bit (bit stuffing) al tramado orientado a bit.
  - Bit stuffing es el proceso de añadir un 0 extra siempre que aparecen cinco unos consecutivos después de un 0 en los datos.

(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0



- CONTROL DE FLUJO Y ERROR:
  - Control de flujo:
    - Define un conjunto de procedimientos usados para restringir la cantidad de datos que el emisor puede enviar antes de esperar una confirmación.
  - Control de error:
    - Incluye tanto la detección como la corrección de error.
    - El control del error en el nivel de enlace se basa en una petición de repetición automática (ARQ, Automatic Repeat Request), que es la retransmisión de los datos.
- PROTOCOLOS:
  - Los protocolos se implementan normalmente en software usando algún lenguaje de programación habitual.
  - La discusión sobre protocolos se divide en aquellos que pueden ser usados en canales:
    - Sin ruido:
      - Libres de error y por tanto ideales.
      - No se pueden usar en la vida real.
      - Lista de protocolos:
        - Simplest.
        - Parada y espera.
    - Con ruido:
      - Generan errores.
      - Usados en la vida real.
      - Lista de protocolos:
        - ARQ con parada y espera.
        - ARQ con vuelta atrás.
        - ARQ con repetición selectiva.
  - En las redes de la vida real, los protocolos de enlace a datos se implementan bidireccionales.
  - La información de control de flujo y errores como los acuse de recibo o ACK (Acknowledgment) y acuse de recibo negativo o NAK (Negative acknowledgment) se incluyen dentro de las tramas de datos usando una técnica denominada superposición de confirmaciones (piggybacking).
- CANALES SIN RUIDO:
  - Protocolo Simplest:
    - Es el más simple.
    - No tiene control de flujo o errores, es unidireccional y el receptor nunca puede ser desbordado por las tramas entrantes.
    - Algoritmo del emisor:

```
while( true )
{
    EsperarEvento();           // dormir hasta que llegue un evento
    if( Evento(SolicitudEnviar ) )
    {
        ObtenerDatos();
        GenerarTrama();
        EnviarTrama();
    }
}
```

- Algoritmo del receptor:

```
while( true )
{
    EsperarEvento();      // dormir hasta que llegue un evento
    if( Evento( NotificacionLlegada ) )
    {
        RecibirTrama();
        ExtraerDatos();
        EntregarDatos();
    }
}
```

- Protocolo con parada y espera (Stop and Wait protocol):

- Evolución del simplex con control de flujo y usando enlace semiduplex.
- Algoritmo del emisor:

```
puedoEnviar = true;    // permite salir a la primera trama
while( true )
{
    EsperarEvento();      // dormir hasta que llegue un evento
    if( Evento( SolicitudEnviar ) AND puedoEnviar )
    {
        ObtenerDatos();
        GenerarTrama();
        EnviarTrama();
        puedoEnviar = false; // no se puede enviar hasta ACK
    }
    EsperarEvento();      // dormir hasta que llegue un evento
    if( Evento( NotificacionLlegada ) ) // ACK recibido
    {
        RecibirTrama();
        puedoEnviar = true;
    }
}
```

- Algoritmo del receptor:

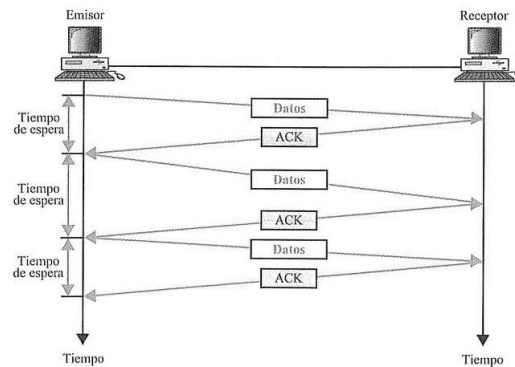
```
while( true )
{
    EsperarEvento();      // dormir hasta que llegue un evento
    if( Evento( NotificacionLlegada ) )
    {
        RecibirTrama();
        ExtraerDatos();
        EntregarDatos(); // Envía datos al nivel de Red
        EnviarTrama();   // Envía trama ACK
    }
}
```

- CANALES CON RUIDO:

- Protocolo Stop-and-Wait ARQ:

- Para detectar y corregir las tramas corruptas, es necesario añadir bit de redundancia a las tramas de datos.
- Cuando la trama llega al receptor, se comprueba y si está corrupta se descarta en silencio.
- Se enumeran las tramas para que el receptor sepa si es una trama correcta, duplicada o fuera de orden.
- La corrección de errores o las tramas perdidas, se hace manteniendo una copia de la trama enviada y retransmitiendo la trama cuando expira el temporizador.
- La trama ACK también necesita bits redundantes y número de secuencia.

- El emisor simplemente descarta una trama ACK corrupta o ignora una fuera de orden.
- Números de secuencia:
  - Hay que minimizar el tamaño de la trama, buscando el rango mínimo de números de secuencia que proporcione una comunicación no ambigua.
  - Los números de secuencia se pueden repetir cíclicamente.
  - Los números de secuencia se basan en aritmética módulo 2.
- Números de confirmación:
  - El número de confirmación siempre anuncia el número de secuencia de la siguiente trama esperada en aritmética módulo 2.



- Algoritmo del emisor:

```

Sn = 0; // Primero se envía la trama 0
puedoEnviar = true; // Permite salir a la primera trama
while( true )
{
    EsperarEvento(); // Dormir hasta que llegue un evento
    if( Evento( SolicitudEnviar ) AND puedoEnviar )
    {
        ObtenerDatos();
        GenerarTrama( Sn );
        AlmacenarTrama( Sn );
        EnviarTrama( Sn );
        IniciarTemporizador();
        Sn = Sn + 1;
        puedoEnviar = false; // No envío hasta llegada ACK
    }
    EsperarEvento(); // Dormir hasta que llegue un evento
    if( Evento( NotificacionLlegada ) ) // Llegada de ACK
    {
        RecibirTrama( ackNo );
        if( noCorrupta AND ackNo == Sn ) // ACK OK
        {
            PararTemporizador();
            PurgarTramas( Sn - 1 ); // Copia no necesaria
            puedoEnviar = true;
        }
    }
    if( Evento( TimeOut ) )
    {
        IniciarTemporizador();
        EnviaTrama( Sn - 1 ); // Reenvia una copia
    }
}

```

- Algoritmo del receptor:

```

Rn = 0; // Primera trama que se espera
while( true )
{
    EsperaEvento(); // Dormir hasta llegada evento

```

```

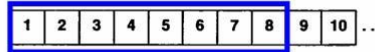
if( Evento( NotificaciónLlegada ) )
{
    RecibirTrama();
    if( seqNo == Rn AND tramaNoCorrupta )
    {
        ExtraerDatos();
        EntregarDatos();
        Rn = Rn + 1;
        EnviarTrama( Rn ); // Enviar trama ACK
    }
}
}

```

- Eficiencia:
  - Este protocolo no es eficiente si el canal es pesado y largo.
  - Pesado quiere decir que el canal tiene un gran ancho de banda.
  - Largo quiere decir que el tiempo de ida y vuelta es largo.
  - El producto de ambos factores se denomina producto ancho de banda-retardo.
- Envío adelantado (pipelining):
  - Esto es el comienzo de una tarea antes de finalizar la anterior.
  - El pipelining mejora la eficiencia de la transmisión si el número de bits de la transmisión es grande con respecto al producto ancho de banda-retardo.
- Petición de respuesta automática con vuelta atrás N:
  - Este protocolo puede enviar varias tramas antes de recibir confirmaciones, haciendo un uso más eficiente del ancho de banda del canal.
  - Se mantiene una copia de estas tramas hasta que llegan las confirmaciones.
  - Números de secuencia:
    - Los números de secuencia son módulo  $2^m$ , donde  $m$  es el tamaño del campo de número de secuencia en bit.
    - Los números de secuencia se pueden repetir cíclicamente.
  - Ventana deslizante:
    - Es un dispositivo de control de flujo de tipo software, es decir, el control del flujo se lleva a cabo mediante el intercambio específico de caracteres o tramas de control, con los que el receptor indica al emisor cuál es su estado de disponibilidad para recibir datos.
    - Se puede ver como un concepto abstracto que define el rango de números de secuencia que usan el emisor y el receptor.
    - El rango que afecta al emisor se denomina ventana deslizante del emisor y el que afecta a receptor ventana deslizante del receptor.
    - La ventana es una caja imaginaria que cubre los datos de secuencia de las tramas de datos que pueden estar en tránsito.
    - El tamaño máximo de la ventana es  $2^m - 1$ .
    - La ventana de envío divide los números de secuencia en posibles en cuatro regiones:
      - La primera región, define la secuencia de números pertenecientes a las tramas ya confirmadas.
      - La segunda región, define el rango de números de secuencia pertenecientes a las tramas que se han enviado y tienen el status desconocido (Unacknowledge).
      - La tercera región, define el rango de números de secuencia pertenecientes a las tramas que se pueden enviar. Sin embargo, los paquetes de datos aun no se han recibido del nivel de red.
      - La cuarta región, define los números de secuencia que no pueden ser usados hasta que no se deslice la ventana.

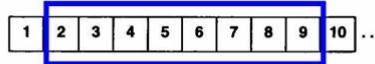
- Hay tres variables que definen su tamaño y posición en cualquier instante:
  - $S_f$ : Primera trama pendiente.
  - $S_n$ : Siguiete trama a enviar.
  - $S_{size}$ : Tamaño de la ventana.
- La ventana de envío puede deslizarse una o más entradas cuando llega una confirmación válida.

ESTADO INICIAL DE LA VENTANA DESLIZANTE



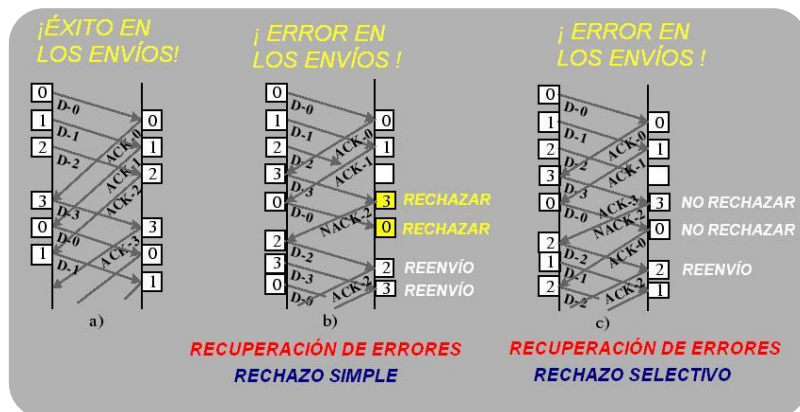
(a)

ESTADO DE LA VENTANA UNA VEZ DESLIZADA



(b)

- La ventana de recepción es un concepto abstracto que define una caja imaginaria de tamaño 1 con una única variable  $R_n$ .
- La ventana desliza cuando llega una trama correcta y el desplazamiento solo se produce en una entrada cada vez.
- Temporizadores:
  - Se utiliza un temporizador para la primera trama pendiente ya que esta expira primero.
- Confirmación:
  - El receptor envía una ACK si la trama ha llegado bien y en orden.
  - En caso contrario mantiene silencio y descarta todas las tramas siguientes hasta que reciba la que está esperando.
  - El receptor no tiene que confirmar cada trama recibida. Puede enviar confirmaciones acumulativas para varias tramas.
- Reenvío de trama:
  - El silencio del receptor hace que expire el temporizador en el emisor, teniendo que volver atrás a reenviar todas las tramas, comenzando con la que expiró el temporizador.
- Tamaño de las ventanas:
  - Debe ser menor que  $2^m$  en el emisor.
  - En el receptor es siempre 1.



- Algoritmo del emisor:

```

Sw = 2m - 1;
Sf = 0;
Sn = 0; // Primero se envía la trama 0

while( true )
{
    EsperarEvento(); // Dormir hasta llegada de evento
    switch( Evento )
    {
        case SolicitudEnviar:
            if( Sn - Sf >= Sw ) // Si ventana llena
            {
                dormir();
            }
            ObtenerDatos();
            GenerarTrama( Sn );
            AlmacenarTrama( Sn );
            EnviarTrama( Sn );
            Sn = Sn + 1;
            if( temporizador parado )
            {
                IniciarTemporizador();
            }
            break;
        case ModificacionLlegada:
            Recibir( ACK );
            if( corrupta( ACK ) )
            {
                dormir();
            }
            if( ( ackNo > Sf ) AND ( ackNo <= Sn ) )
            {
                while( Sf <= ackNo )
                {
                    PurgarTrama( Sn - 1 );
                    Sf = Sf + 1;
                }
                PararTemporizador();
            }
            break;
        case TimeOut:
            IniciarTemporizador();
            Temp = Sf;
            while( Temp < Sn )
            {
                EnviarTrama( Sf );
                Sf = Sf + 1;
            }
            break;
        default:
            break;
    }
}

```

- Algoritmo del receptor:

```

Rn = 0;

while( true )
{
    EsperarEvento();
    if( Evento( ModificacionLlegada ) )
    {
        RecibirTrama();
        if( tramacorrupta )
        {
            dormir();
        }
    }
}

```

```

        if( seqNo == Rn )
        {
            EntregarDatos();
            Rn = Rn + 1;
            EnviarACK( Rn );
        }
    }
}

```

○ Petición de respuesta automática con recepción selectiva:

- El protocolo ARQ con vuelta atrás N, es muy ineficiente en enlaces ruidosos.
- En un enlace ruidoso una trama tiene mayor posibilidad de dañarse, lo que significa reenviar múltiples tramas, sobreutilizando el ancho de banda y haciendo la transmisión más lenta.
- ARQ con repetición selectiva es más eficiente para canales ruidosos, pero el procesamiento en el receptor es más complejo.
- Tamaños de ventanas:
  - Tanto en el emisor, como en el receptor, el tamaño de la ventana ha de ser  $2^{m-1}$ .
- Ventanas:
  - El protocolo usa las mismas variables que en vuelta atrás N.
  - Este protocolo permite que lleguen fuera de orden tantas tramas como el tamaño de la ventana de recepción y que se almacenen hasta que haya un conjunto de tramas fuera de orden listas para enviar al nivel de red.
  - Sin embargo el receptor nunca envía tramas fuera de orden al nivel de red.
- Algoritmos del emisor:

```

Sw = 2m-1;
Sf = 0;
Sn = 0;

while( true )
{
    EsperarEvento();
    switch( Evento )
    {
        case SolicitudEnviar:
            if( Sn - Sf >= Sw )
            {
                dormir();
            }
            ObtenerDatos();
            GenerarTrama( Sn );
            AlmacenarTrama( Sn );
            EnviarTrama( Sn );
            Sn = Sn + 1;
            IniciarTemporizador();
            break;
        case NotificaciónLlegada:
            Recibir( ACK );
            if( corrupta( ACK ) )
            {
                dormir();
            }
            switch( FrameType )
            {
                case NAK:
                    if( nakNo entre Sf y Sn )
                    {
                        reenviar( nakNo );
                        IniciarTemporizador( nakNo );
                    }
                    break;
                case ACK:
                    if( nakNo entre Sf y Sn )
                    {

```



```

                                while( Sf <= ackNo )
                                {
                                    PurgarTrama( Sf );
                                    PararTemporizador( Sf );
                                    Sf = Sf + 1;
                                }
                                break;
                            }
                            break;
                    case Timeout:
                        IniciarTemporizador( t );
                        EnviarTrama( t );
                        break;
                    }
                }
            }

```

- Algoritmo del receptor:

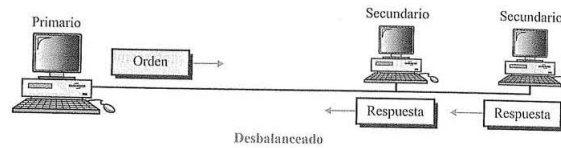
```

Rn          = 0;
NakSent     = false;
AckNeeded   = false;
Repetir( para todos los slots )
{
    Marcado( slot ) = false;
}
while( true )
{
    EsperarEvento();

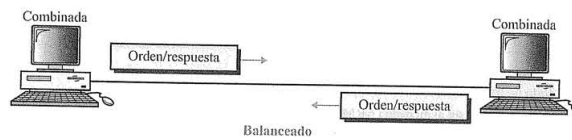
    if( Evento( NotificaciónLlegada ) )
    {
        RecibirTrama();
        if( corrupta( TRAMA ) AND ( NOT NakSent ) )
        {
            SendNAK( Rn );
            NakSent = true;
            dormir();
        }
        if( ( seqNo <> Rn ) AND ( NOT NakSent ) )
        {
            SendNAK( Rn );
            NakSent = true;
            if(seqNo in window AND NOT Marcada( seqNo ))
            {
                AlmacenarTrama( seqNo );
                Marcada( seqNo ) = true;
                while( Marcado( Rn ) )
                {
                    EntregarDtos( Rn );
                    Purgar( Rn );
                    Rn = Rn + 1;
                    AckNeeded = true;
                }
                if( AckNeeded )
                {
                    EnviarAck( Rn );
                    AckNeeded = false;
                    NakSent = false;
                }
            }
        }
    }
}

```

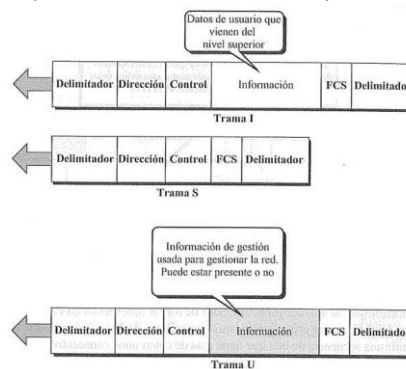
- Piggybacking:
  - Técnica que mejora la eficiencia de los protocolos bidireccionales.
  - Con esta técnica, en vez de enviar ACK en un paquete individual, éste es incluido dentro del próximo paquete a enviar.
  - Un punto importante sobre el piggybacking es que ambos nodos (A y B) deben tener el mismo algoritmo.
- CONTROL DE ENLACE DE DATOS DE ALTO NIVEL (HDLC, Hig-Level Data Link Control):
  - Protocolo orientado a bit sobre enlaces punto a punto o multipunto.
  - Implementan mecanismos ARQ.
  - Configuración y modos de comunicación:
    - Modo de respuesta normal (NRM):
      - La configuración está desbalanceada, también denominada maestro-esclava.
      - Una estación primaria puede enviar comandos.
      - Una estación secundaria sólo puede responder.



- Modo asíncrono balanceado (ABM):
  - La configuración está balanceada, cada estación puede funcionar como primaria o secundaria.
  - Los enlaces son punto a punto.

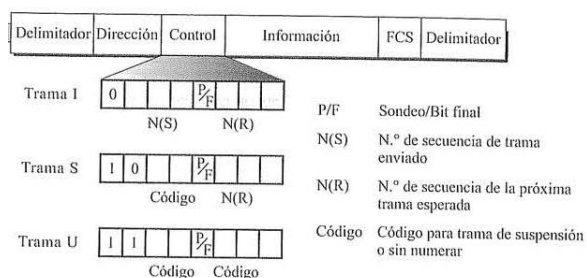


- Tramas:
  - Formato de trama:
    - Cada trama en HDLC puede contener hasta seis campos.



- Campos:
  - Campo etiqueta:
    - Secuencia de 8 bits, con patrón 01111110 que identifica el principio-final de trama o sincronización para el receptor.
  - Campo dirección:
    - Si se trata de una estación primaria, la dirección es a quien va dirigida la trama.
    - Si se trata de una estación secundaria, la dirección es de quien va dirigida la trama
    - Puede tener una longitud de uno o varios bytes, dependiendo del tamaño de la red.

- Un byte puede identificar 128 estaciones.
- En el último byte, el último bit siempre es 1 y el resto de bytes, si los hubiera, el último bit siempre es 0. Indicando al receptor que va a recibir más bytes de direcciones.
- Campo de control:
  - Segmento de uno o dos bytes de la trama usado para la gestión de flujo.
- Campo de información:
  - Contiene los datos de usuario desde el nivel de red o información de control.
  - Su longitud puede variar de una red a otra.
- Campo FCS (Frame Check Sequence):
  - O secuencia de control de trama.
  - Es el campo de detección de error de HDLC.
- Campos de control:
  - Determinan el tipo de trama y definen su funcionalidad.
  - Existen tres tipos de campos de control de tramas.



- Campo de control de trama I (Información):
  - Tipo de trama diseñado para transportar datos desde el nivel de red.
  - Pueden incluir piggybacking.
  - La longitud del campo puede ser de 1 byte (normal) o de 2 (extendida).
  - Reparto de bits:
    - Bit 7 (MSB):
      - Si es 0 significa que es trama I.
    - Bit 6, 5, 4:
      - Denominados N(S), definen el número de secuencia.
      - Si el campo es extendido puede aumentar el número de bits.
    - Bit 3:
      - Denominado bit P/F.
      - Si está a 1 puede significar muestra o final.
      - Muestra es cuando la trama se envía de una estación primaria a una secundaria.
      - Final es justo lo contrario a muestra.
    - Bit 2, 1, 0:
      - Denominados N(R).
      - Corresponde con el campo reconocimiento cuando se usa piggybacking.
- Campo de control de trama S (Supervisión):
  - Tipo de tramas diseñada para control de flujo u errores cuando el piggybacking no es posible o apropiado.
  - No contienen campo de información.

- Reparto de bits:
  - Bit 7, 6 (MSB):
    - Si es 10 significa que es trama S.
  - Bit 5, 4:
    - Denominados código, definen la trama S.
    - Si es 00 significa listo para recibir (RR).
    - Si es 10 significa receptor no listo para recibir (RNR).
    - Si es 01 significa rechazo (REJ).
    - Si es 11 significa rechazo selectivo (SREJ).
  - Bit 3:
    - Igual que en trama I.
  - Bit 2, 1, 0:
    - Denominados N(R).
    - Corresponde con el número de la confirmación (ACK) o del rechazo (NACK) dependiendo del tipo de trama S.
- Campo de control de trama U (Sin numeración):
  - Tipo de tramas diseñada para intercambiar información de control y gestión de la sesión entre dos dispositivos conectados.
  - Reparto de bits:
    - Bit 7, 6 (MSB):
      - Si es 11 significa que es trama S.
    - Bit 3:
      - Igual que en trama I.
    - Bit 5, 4 y 2, 1, 0:
      - Con estos 5 bits se pueden crear hasta 32 tipos de tramas U.

Código	Comando	Resp.	Significado
00 001	SNRM		Modo respuesta normal
11 011	SNRME		Modo respuesta normal ampliada
11 100	SABM	DM	Modo resp. asíncrona balanceada
11 110	SABME		Modo resp. asíncrona balanceada ampliada.
00 000	UI	UI	Info. sin numerar.
00 110		UA	Reconocimiento sin numerar.
00 010	DISC	RD	Desconexión.
10 000	SIM	RIM	Modo petición de info.
00 100	UP		Muestra sin numerar.
11 001	RSET		Reset.
11 101	XID	XID	Intercambio de ID.
10 001	FRMR	FRMR	Rechazo trama.

- PROTOCOLO PUNTO A PUNTO (PPP):
  - Protocolo punto a punto de nivel de enlace que controla y maneja la transferencia de datos.
  - Servicios proporcionados:
    - Define el formato de la trama.
    - Define como se negocia el establecimiento del enlace y el intercambio de datos.
    - Define como se encapsulan los datos de nivel de red en una trama del nivel de enlace.
    - Define la autenticación entre dos dispositivos.
    - Proporciona múltiples servicios de nivel de red.
    - Soporta una variedad de protocolos de nivel de red.
    - Proporciona conexiones sobre múltiples enlaces.
    - Proporciona configuraciones de dirección de red.

- Servicios no proporcionados:
  - No proporciona control de flujo.
  - Tiene un mecanismo muy sencillo de control de error (CRC).
  - No proporciona un mecanismo de direcciones sofisticado para manejar las tramas en una configuración multipunto.
- Tramado:
  - PPP es un protocolo orientado a byte.
  - Formato de trama:
    - Flag: Comienza y termina con un byte con el patrón 0x7E.
    - Dirección: Valor constante 0xFF que es la dirección broadcast.
    - Control: Valor constante 0xC0, que corresponde con tramas de usuario no numeradas en HDLC. **En wikipedia el valor del campo control es 0x03.**
    - Protocolo: Define que se transporta en el campo de datos.
    - Carga útil:
      - Secuencia de 1500 bytes máximo por defecto.
      - Como no se define el tamaño del campo de datos, es necesario rellenarlo si el tamaño es menor que el tamaño por defecto o negociado.
    - FCS (Frame Check Sequence): Secuencia de comprobación de tramas (CRC).

Flag	Dirección	Control	Protocolo	Carga útil	FCS	Flag
0x7E	0xFF	0xC0	1 ó 2 bytes	Variable	2 ó 4 bytes	0x7E

- Transparencia de byte:
  - PPP es un protocolo orientado a byte que usa transparencia de bytes con la secuencia de escape 0x7E, lo que significa que cada vez que aparece un patrón como el del flag en los datos, se rellena esta byte extra para decir al receptor que el byte siguiente no es un flag.

- Transición de fases:
  - Muerto: No hay portadora activa y la línea está en silencio.
  - Establecer: Se negocian las opciones entre las dos partes, cuando una de ellas comienza la comunicación.
  - Autenticar: Es opcional y ambos nodos pueden saltársela si lo deciden en la fase de establecimiento.
  - Red: Se negocian los protocolos de nivel de red antes de poder intercambiar datos a nivel de red.
  - Abrir: Comienzo de intercambio de paquetes de datos hasta que uno de los extremos quiera terminar la conexión.
  - Terminar: Corte de la conexión, limpiando datos remanentes y cierre del enlace.
- Multiplexación:
  - PPP usa otro conjunto de protocolos para establecer el enlace, autenticar a los socios involucrados y transportar los datos de nivel de red.
  - Protocolo de control de enlace (LCP, Link Control Protocol):
    - Es el responsable de establecer, mantener, configurar y terminar enlaces.
    - También proporciona mecanismos de negociación para definir las opciones entre ambos extremos.
    - Los paquetes LCP se transportan en el campo de carga útil, con el campo protocolo fijado a 0xC021.

Flag	Dirección	Control	Protocolo	Carga útil	FCS	Flag
0x7E	0xFF	¿?	0xC021	Variable	2 ó 4 bytes	0x7E

Paquete LCP			
Código	ID	Longitud	Información (Variable)

- En el campo Código se definen el tipo de paquete LCP, existiendo 11 tipos de estos:

Categoría	Código	Tipo	Descripción
Configuración del enlace durante el establecimiento	0x01	Petición de configuración	Lista de opciones y sus valores
	0x02	ACK de configuración	Acepta todas las propuestas
	0x03	NACK de configuración	Anuncio de opciones inaceptables
	0x04	Rechazo de configuración	Anuncio de opciones reconocidas
Terminación del enlace	0x05	Petición de terminación	Petición de corte de línea
	0x06	ACK de terminación	Acepta petición de terminación
	0x07	Rechazo de código	Anuncio de código desconocido
Monitorización del enlace y depuración	0x08	Rechazo de protocolo	Anuncio de protocolo desconocido
	0x09	Petición de eco	c de comunicación
	0x0A	Respuesta de eco	Respuesta al mantenimiento
	0x0B	Petición de descartar	Paquete a descartar

- En el campo ID se almacena un valor que casa una petición por una respuesta.
- En el campo Longitud se define la longitud del paquete LCP completo.
- En el campo Información se definen las opciones de negociación entre los extremos, dividiéndolo en tres campos: tipo, longitud y datos de opción.

Opción	Defecto
Tamaño del campo de carga útil	1500
Protocolo de autenticación	Ninguno
Compresión del campo de protocolo	Off
Compresión del campo de control y dirección	Off

- Protocolos de autenticación (AP, Authentication Protocol):**
    - La autenticación significa invalidar la identidad de un usuario que necesita acceder a un conjunto de recursos.
    - Protocolo de autenticación por contraseña (PAP, Password Authentication Protocol):**
      - El valor en el campo de protocolo es 0xC023.
      - El usuario que accede al sistema envía una identificación de autenticación (login).
      - El sistema comprueba la validez del login y acepta o no la conexión.
    - Protocolo de autenticación por reto (CHAP, Challenge Handshake AP):**
      - Método más seguro que PAP, debido a que el pass se mantiene en secreto y nunca se envía por la red.
      - El valor en el campo de protocolo es 0xC223.
      - El sistema envía al usuario un paquete reto conteniendo un valor (semilla).
      - El usuario aplica una función predeterminada con la semilla y la propia contraseña del usuario y crea un resultado que devuelve al sistema.
      - El sistema hace lo mismo, comparando ambos resultados y dando acceso si la comparación es igual y negándoselo en caso contrario.
  - Protocolo de control de red (NCP, Network Control Protocol):**
    - Es el encargado de negociar los parámetros específicos para cada protocolo utilizado por Internet, OSI, Xerox, DECnet, Novel, etc. Es decir, configura la red que va sobre el PPP.
    - Se define un protocolo de red específico para cada protocolo de red.
    - IPCP, Internet Protocol Control Protocol:**

- Configura el enlace usado para transportar paquetes IP en Internet.
- El valor en el campo protocolo es 0xc.

Flag	Dirección	Control	Protocolo	Carga útil	FCS	Flag
0x7E	0xFF	¿?	0x8021	Variable	2 ó 4 bytes	0x7E

Paquete LCP			
Código	ID	Longitud	Información IPCP

- Otros protocolos NCP para protocolos de nivel de red:
  - OSI: con valor en el campo de control de 0x8023.
  - IDP NS Xerox: con valor en el campo de control de 0x8025.
  - El valor de código y formatos de los paquetes para estos protocolos son:

Código	Tipo
0x01	Petición de configuración
0x02	ACK de configuración
0x03	NACK de configuración
0x04	Rechazo de configuración
0x05	Petición de terminación
0x06	ACK de terminación
0x07	Rechazo de código

- Datos del nivel de red:
  - Completada la configuración del nivel de red por uno de los protocolos NCP, los usuarios pueden intercambiar DTE (Data Terminal Equipment) o paquetes de datos de nivel de redes.
  - Nuevamente existen diferentes valores en el campo protocolo para los distintos niveles de red:
    - IP: valor 0x0021.
    - OSI: valor 0x0023.

- PPP MULTILINK:
  - Existencia de múltiples canales en un único enlace punto a punto.
  - Una trama PPP lógica se divide en varias tramas PPP reales.
  - Un segmento de la trama lógica es transportado en la carga útil de una trama PPP real.
  - Para mostrar que la trama PPP real transporta un fragmento de una trama PPP lógica, el campo protocolo se pone a 0x003D.
  - Es necesario añadir un número de secuencia a la trama real PPP.