

# Redes de Datos-Control del enlace de Datos

*Jhon Jairo Padilla Aguilar*  
PhD. Ingeniería Telemática

# Punto de Partida

- Suponemos que:
  - El problema de comunicación a nivel físico ya ha sido solucionado (modulaciones, canales, sincronización).
  - Ahora se pretende que haya comunicación entre dos computadores.
  - Se deben solucionar problemas como el *sobreflujo en el receptor, la detección de errores y el control de estos.*

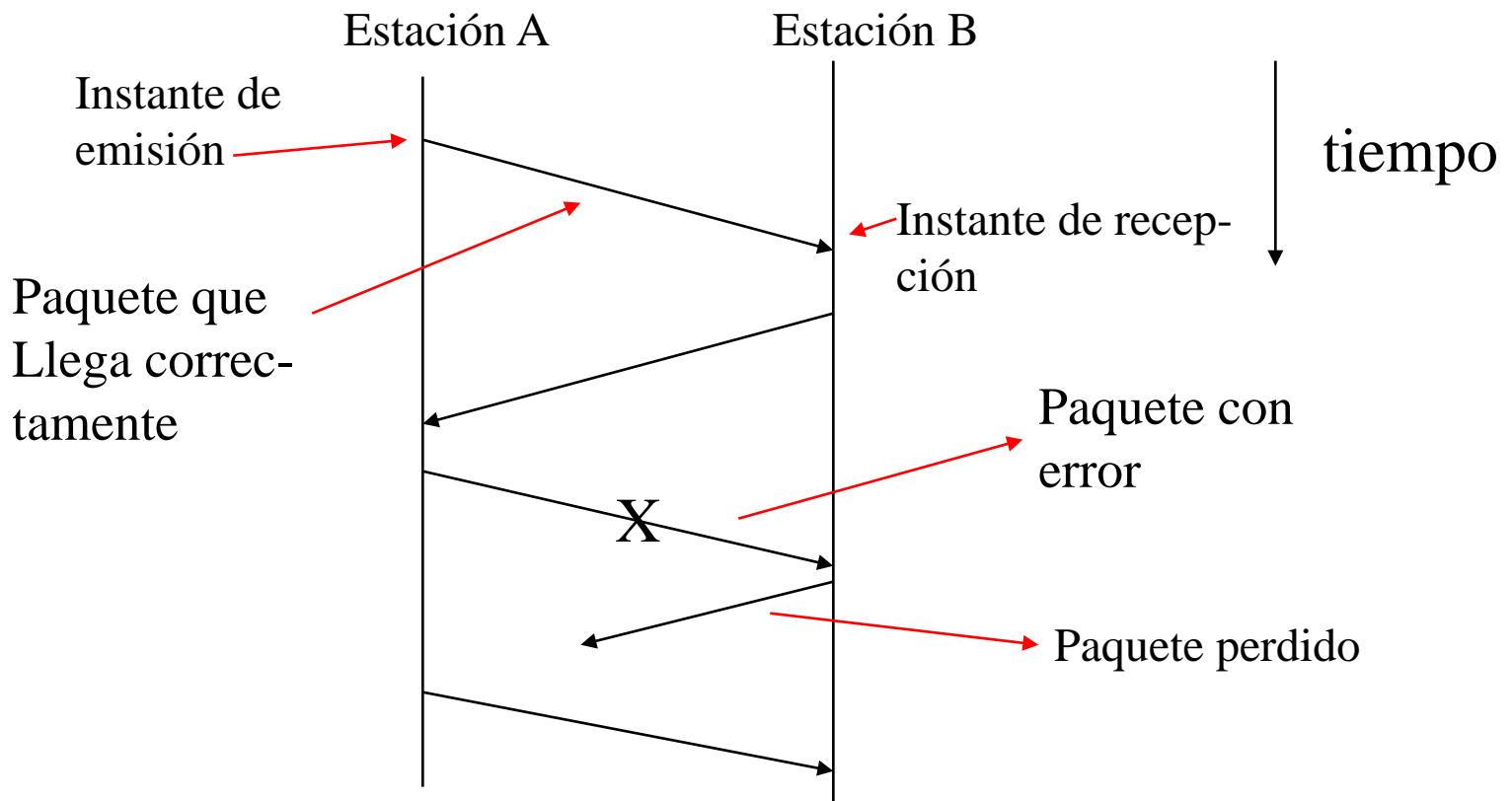
# Control de Flujo

- El problema:
  - El receptor tiene una cantidad finita de memoria y una velocidad de procesamiento máxima.
  - Si el transmisor envía datos en gran cantidad y muy rápido, el receptor podría no alcanzar a procesar los datos porque se sobrecarga el buffer de almacenamiento o porque no alcanza a procesar los paquetes anteriores.

# Control de flujo

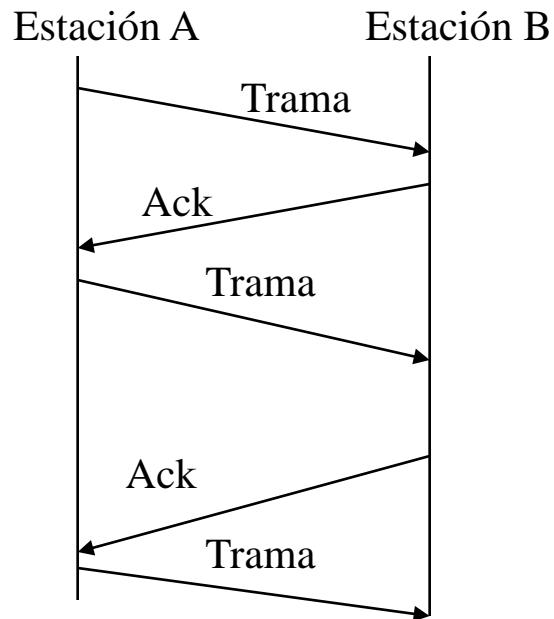
- La solución:
  - Consiste en que el receptor pueda controlar la velocidad a la que le genera paquetes el transmisor.
  - Para ello existen métodos para que el receptor pueda detener el envío de paquetes del transmisor mientras procesa los datos actuales.

# Diagramas de flujo de información vs. tiempo



# Métodos de Control de flujo

- Parada y Espera (Stop & Wait)
  - El transmisor sólo puede enviar el siguiente paquete si el receptor lo autoriza mediante una confirmación del paquete anterior.
  - Existen paquetes de confirmación positiva (ACK)



# Utilización del medio con Parada y Espera

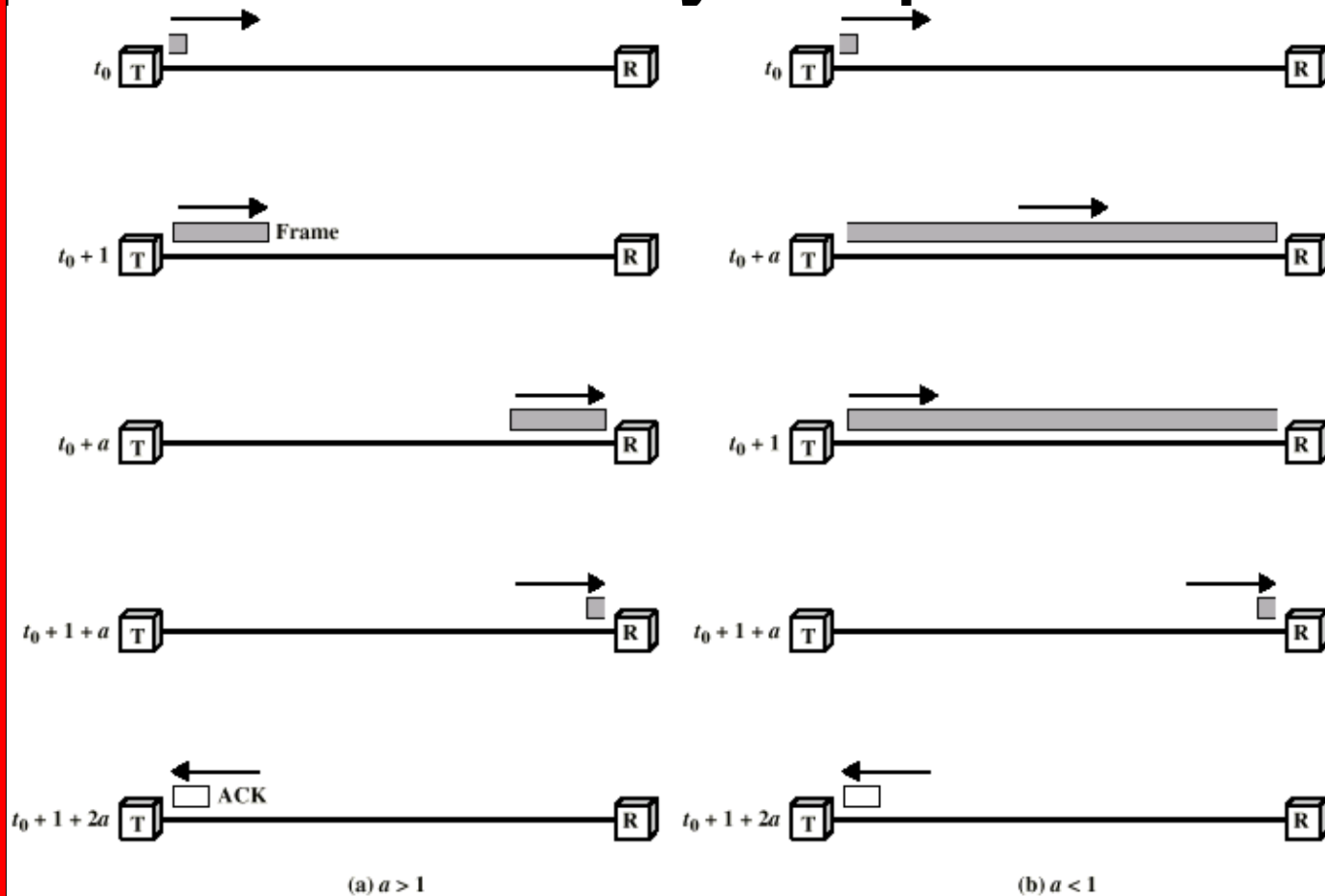


Figure 7.2 Stop-and-Wait Link Utilization (transmission time = 1; propagation time =  $a$ )

# Control de flujo con ventana deslizante (sliding window)

- El inconveniente del método de parada y espera es que *sólo puede haber una trama en tránsito en cada momento*. Esto hace que la utilización del enlace sea ineficiente.
- La eficiencia del enlace puede mejorarse si se permite transitar varias tramas al mismo tiempo.



# Control de flujo con ventana deslizante

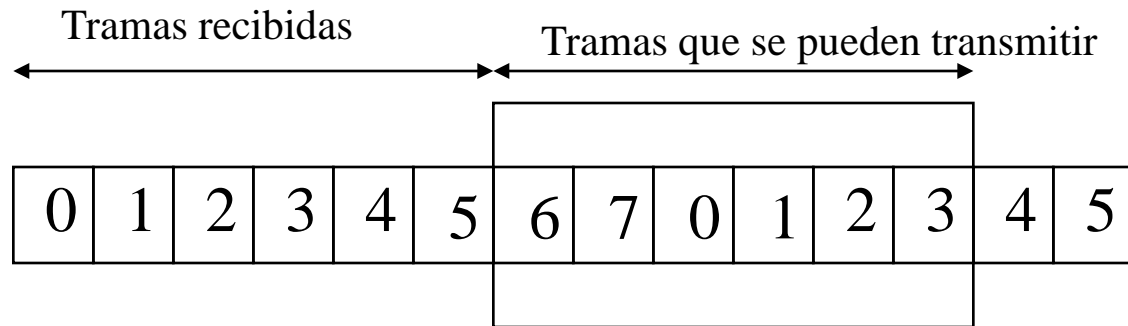
- Se supone un enlace full duplex.
- La estación receptora (B) puede almacenar  $n$  tramas.
- La estación transmisora (A) puede enviar  $n$  tramas consecutivas sin recibir confirmación de B.
- Para llevar el control de qué tramas se han confirmado, las tramas llevan un número de secuencia.

# Control de flujo con ventana deslizante

- B puede confirmar enviando una trama ACK (RR, Receive Ready) junto con el número de la siguiente trama que espera recibir.
- Esto permite confirmar varias tramas simultáneamente.
- La estación A tiene una lista de los números de secuencia de las tramas que se le permiten transmitir (ventana).
- La estación B tiene una lista de los números de secuencia de las tramas que está esperando recibir.

# Manejo de la ventana deslizante

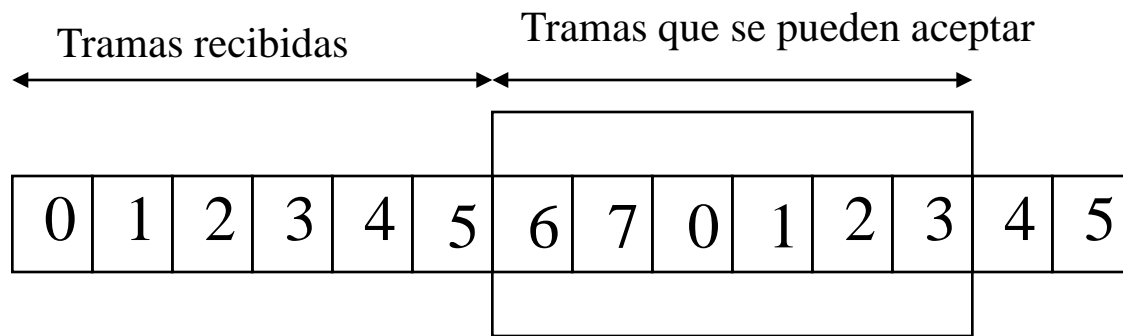
Trans-  
misor



→  
 La ventana se cierra  
 Cuando se envían tramas

→  
 La ventana se abre cuando se  
 Reciben confirmaciones

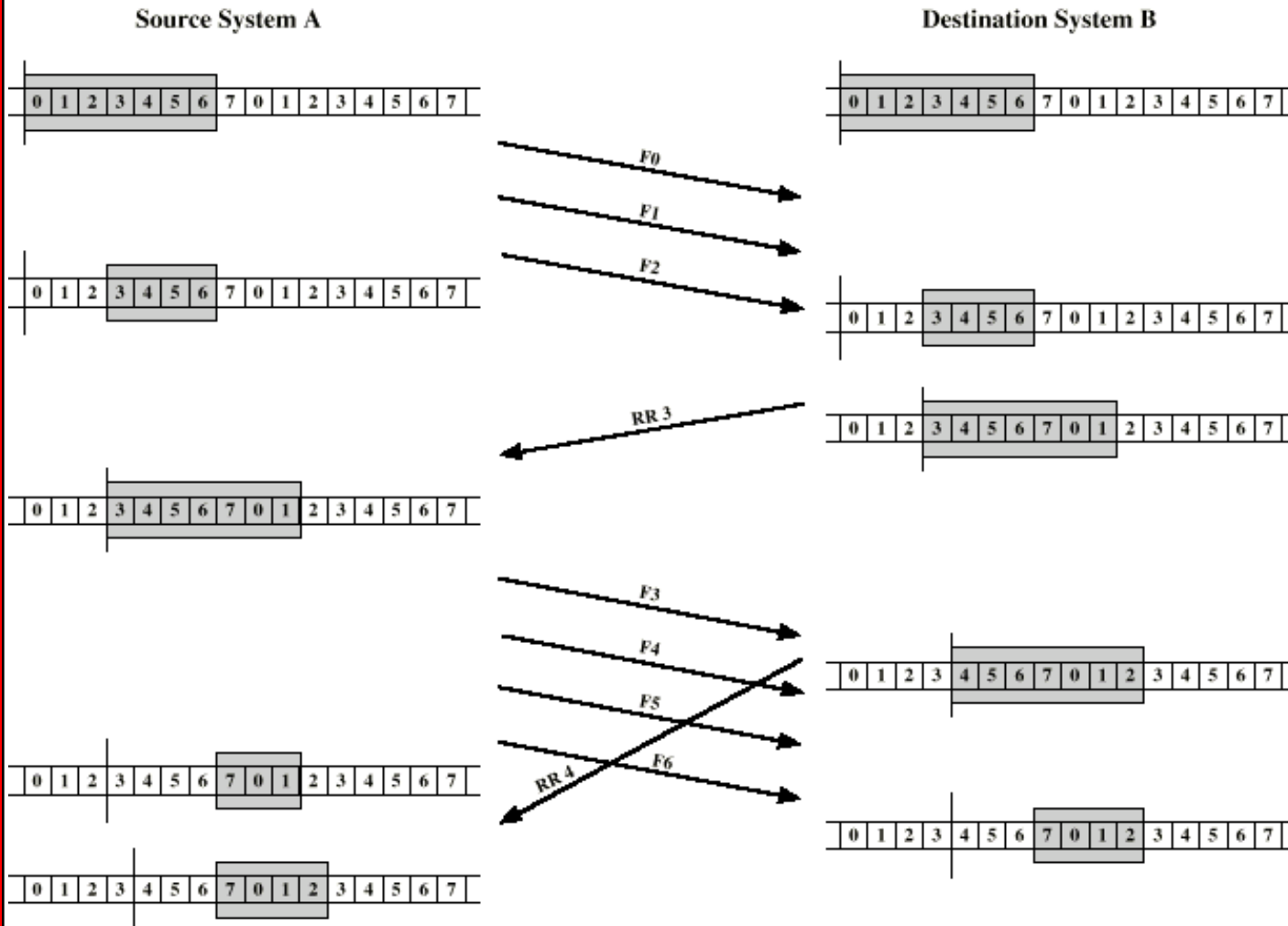
Recep-  
tor



→  
 La ventana se cierra  
 Cuando se reciben tramas

→  
 La ventana se abre cuando  
 Se envían confirmaciones

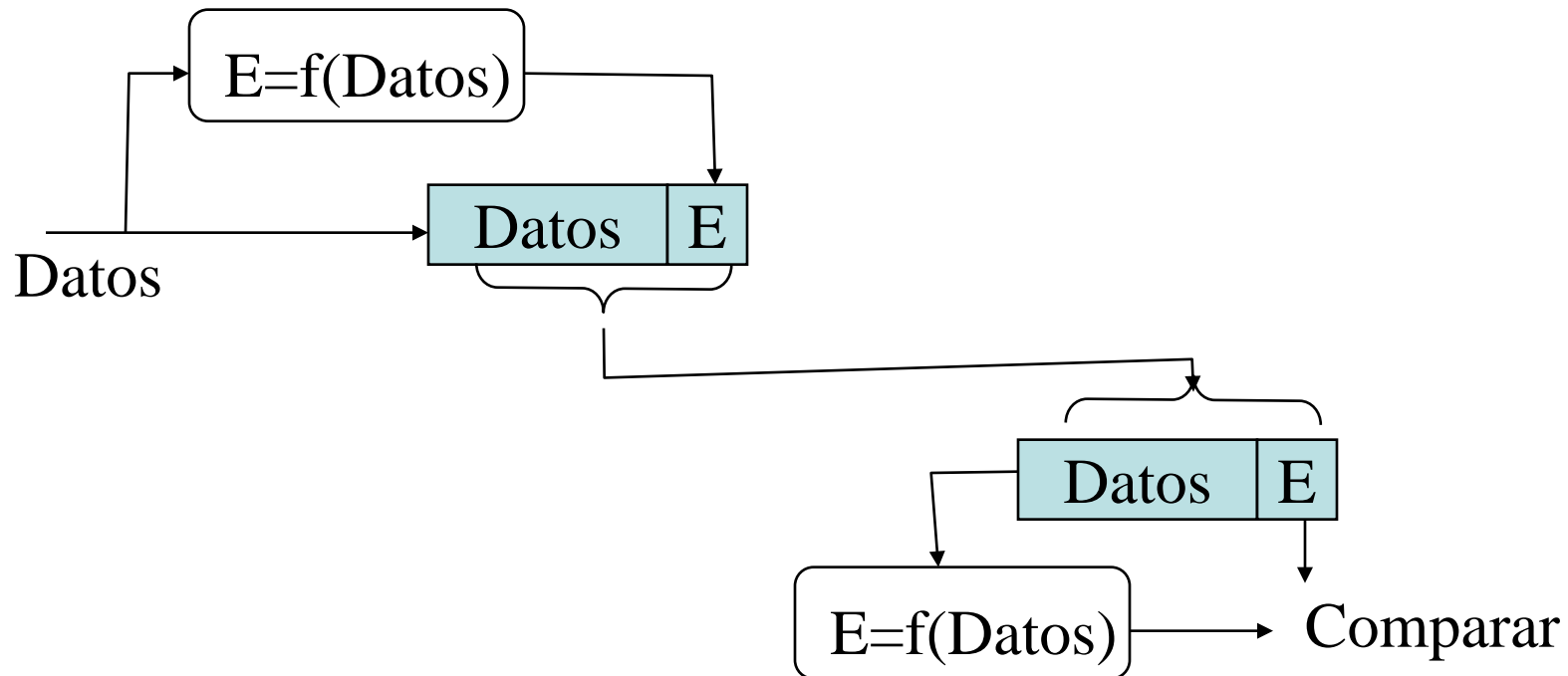
# Ejemplo de Ventana deslizante



# Detección de errores

- Recordemos que la BER depende de:
  - $E_b/N_0$
  - Por tanto de la Potencia de la señal y de la velocidad de tx de datos.
- La probabilidad de error en una trama aumenta con la longitud de la trama.

# Principio de la detección de errores



# Comprobación de paridad

- Es el método más sencillo
- Se añade un bit de paridad al final de los datos.
- El valor del bit de paridad se determina así:
  - Si la paridad es impar:
    - Si el número de unos de los datos es impar,  $p=0$
    - Si el número de unos de los datos es par,  $p=1$
  - Si la paridad es par:
    - Si el número de unos de los datos es impar,  $p=1$
    - Si el número de unos de los datos es par,  $p=0$
- *Desventaja:* Si ha habido un número de bits erróneos par, no se detecta el error.

# Comprobación de redundancia cíclica

- CRC (Cyclic redundancy check)
- Es muy común y muy potente
- Se supone una cadena de datos de  $k$  bits ( $M$ )
- Se genera una cadena de  $n$  bits denominada FCS (Frame Check Sequence) (la llamaremos  $F$ ).
- Se concatenan las dos cadenas ( $M$  y FCS) de manera que el resultado debe ser divisible por un número predeterminado ( $P$ ).
- $P$  tiene  $n+1$  bits.



# Comprobación de redundancia cíclica

- En el receptor se recibe la cadena M:FCS y se divide por el número predeterminado. Si el residuo es cero, quiere decir que es correcta la cadena de bits recibida.
- Proceso:
  - $2^n M$  : se rota n veces a la izquierda a M (añade n 0s al resultado).
  - $(2^n M)/P=Q+(R/P)$
  - $T=2^n M+R$  ; T es la cadena Transmitida

# Polinomios generadores

- El número P se puede expresar como un polinomio de una variable muda X. Los coeficientes son los bits del número binario.
- Ejemplo:
  - Para P=11001  $P(x) = x^4 + x^3 + 1$
- Estándares:
- CRC-16  $= x^{16} + x^{15} + x^2 + 1$
- CRC-CCITT  $= x^{16} + x^{12} + x^5 + 1$
- CRC-32  $= x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8$   
 $+ x^7 + x^5 + x^4 + x^2 + 1$

# Ejemplo

◆ M=1010001101 (10 bits)

◆ P=110101 (6 bits)

101000110100000	110101	
<u>110101</u> ↓		1101010110
0111011		
<u>110101</u> ↓↓		
00111010		
<u>110101</u> ↓↓		
00111110		
<u>110101</u> ↓↓		
00101100		
<u>110101</u> ↓		
0110010		
<u>110101</u> ↓		
0001110 → R		

101000110101110	110101
<u>110101</u>	1100010
0111011	
<u>110101</u>	
00111010	
<u>110101</u>	
00111110	
<u>110101</u>	
00101111	
<u>110101</u>	
0110101	
<u>110101</u>	
0000000 → R	

# Errores que puede detectar CRC

- Todos los errores de un único bit
- Todos los errores dobles (Si  $P$  tiene al menos 3 unos)
- Cualquier número impar de errores (si  $P$  contiene el factor  $x+1$ )
- Cualquier error a ráfagas en que la long de la ráfaga sea menor que la long de  $P$
- La mayoría de ráfagas de mayor longitud

# Comprobación de suma (Check Sum)

- Se toman bloques de datos organizados en filas y columnas de bits
- Se hace una operación XOR por filas o por columnas o ambas.
- El resultado se anexa como FCS

# Check Sum

1101	0
1011	1
1100	0
1110	1

Si se usan los dos resultados (filas y columnas) es una comprobación de Suma cruzada.

0101

# Control de errores

- El objetivo es corregir los errores que se han detectado en tramas mediante los métodos de detección de errores.
- Además se deben recuperar las tramas perdidas.
- Los errores son corregidos mediante retransmisión de las tramas.

# Control de errores

- Los métodos de control de errores usan varias estrategias:
  - Detección de errores
  - Confirmaciones positivas (tramas correctas)
  - Retransmisión después de la expiración de un intervalo de tiempo (**tramas perdidas: errores en dirección destino**)
  - Confirmación negativa y retransmisión (**tramas con errores: en la información útil**)



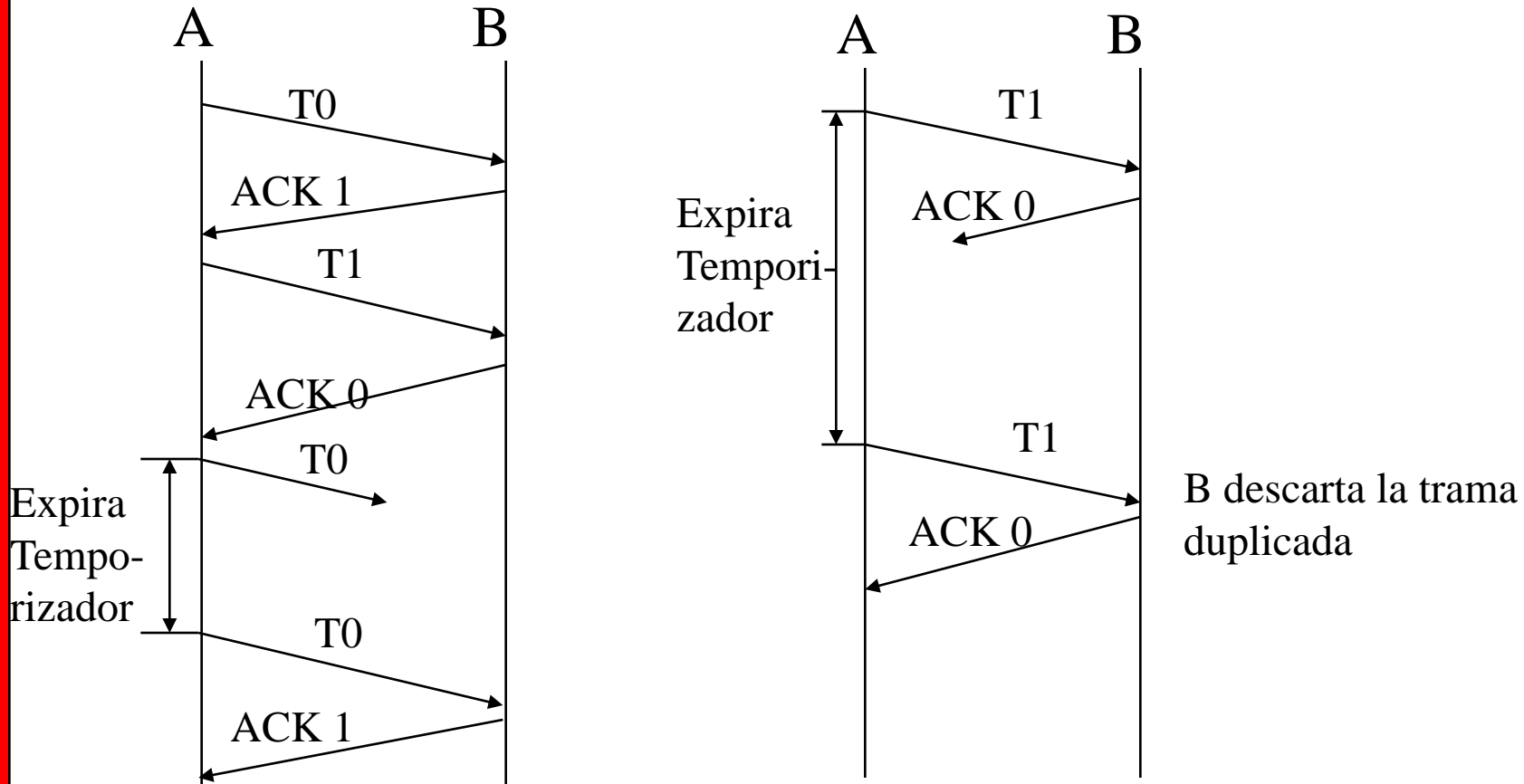
# Control de errores

- Los métodos de control de errores se denominan comúnmente ARQs (Automatic Repeat Request).
- Tipos de ARQs:
  - ARQ con parada y espera
  - ARQ con adelante-atrás-N
  - ARQ con rechazo selectivo

# ARQ con parada y espera

- Se basa en la técnica de control de flujo con parada y espera.
- La estación transmisora transmite una trama y espera a que sea confirmada positivamente para enviar la siguiente.
- En caso de trama errada o trama perdida se emplea un **temporizador de espera de confirmación** para retransmisión en caso de no llegar ésta.
- Las tramas se identifican con un número de secuencia 0 y 1 para evitar duplicaciones de tramas (caso de pérdida de trama).

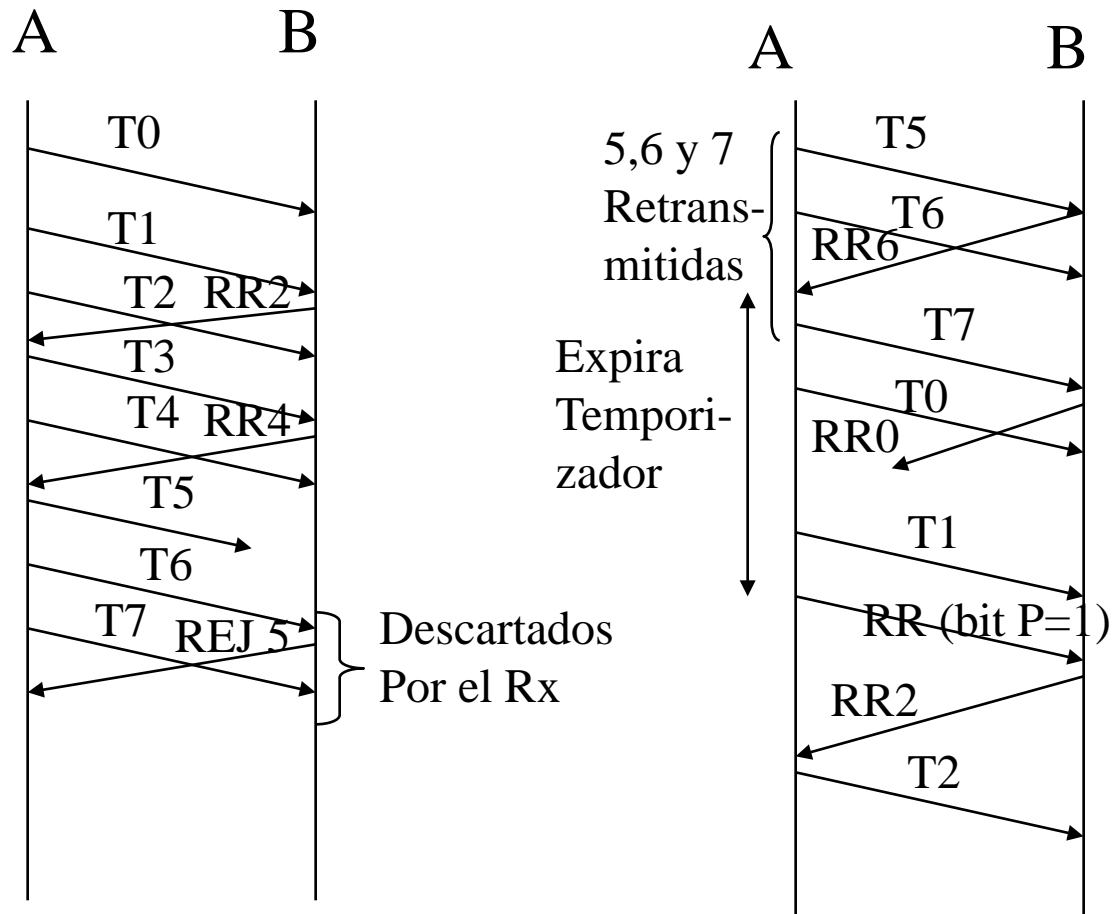
# ARQ con parada y espera



# ARQ con adelante-atrás-N

- Se basa en el método de control de flujo de ventana deslizante.
- Mientras no aparezcan errores, el receptor enviará una confirmación (RR, Receive Ready).
- En caso de error en el receptor, este enviará una trama de confirmación negativa (REJ, Reject).
- La estación receptora descartará esta trama y todas las tramas que se reciban en el futuro hasta que la trama errada se reciba correctamente.
- Cuando la estación transmisora recibe un REJ, deberá retransmitir todas las tramas a partir de la trama rechazada.

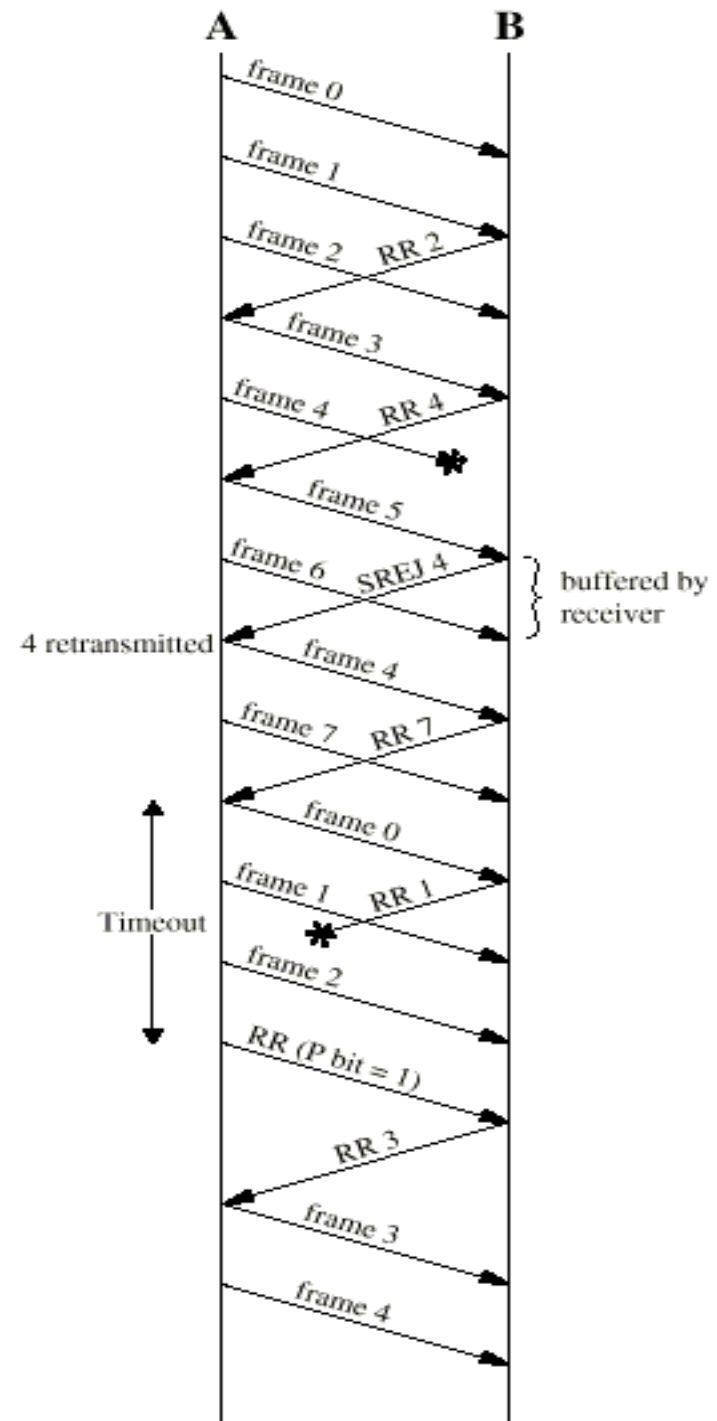
# ARQ con adelante-atrás-N



# ARQ con rechazo selectivo

- Se basa en el método de control de flujo con ventana deslizante.
- Se diferencia del ARQ con adelante-atrás-N en que cuando hay una confirmación negativa (rechazo selectivo ,SREJ), sólo se retransmite la trama rechazada y se regresa al orden que se llevaba.
- Es más complejo de realizar.
- Requiere más memoria.

# ARQ con rechazo selectivo



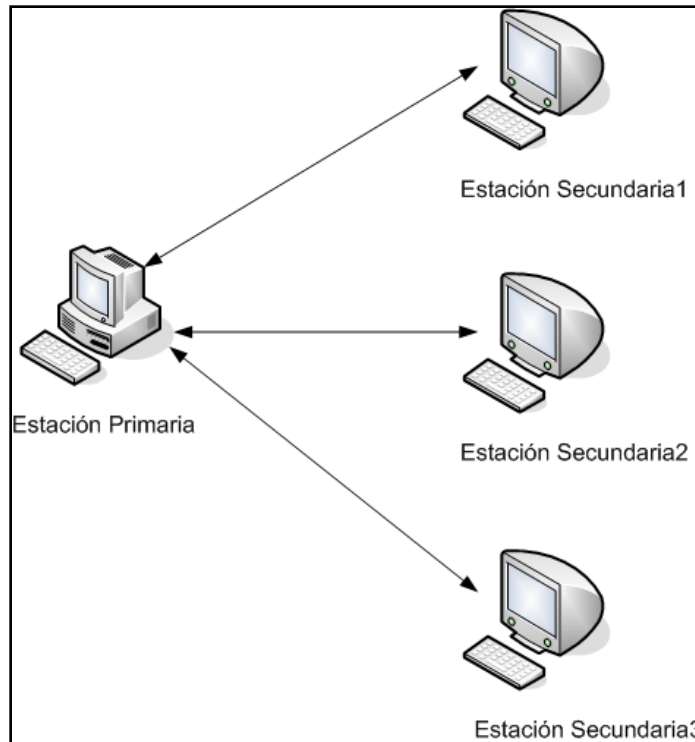
# El protocolo HDLC

- HDLC: High Level Data Link Control
- Normas: ISO 3309, ISO 4335
- Tipos de Estaciones:
  - *Primaria*: Coordina el orden de las comunicaciones. Emite órdenes
  - *Secundaria*: Funciona bajo el control de la estación primaria. Genera Respuestas
  - *Combinada*: Sus tramas pueden ser tanto órdenes como respuestas

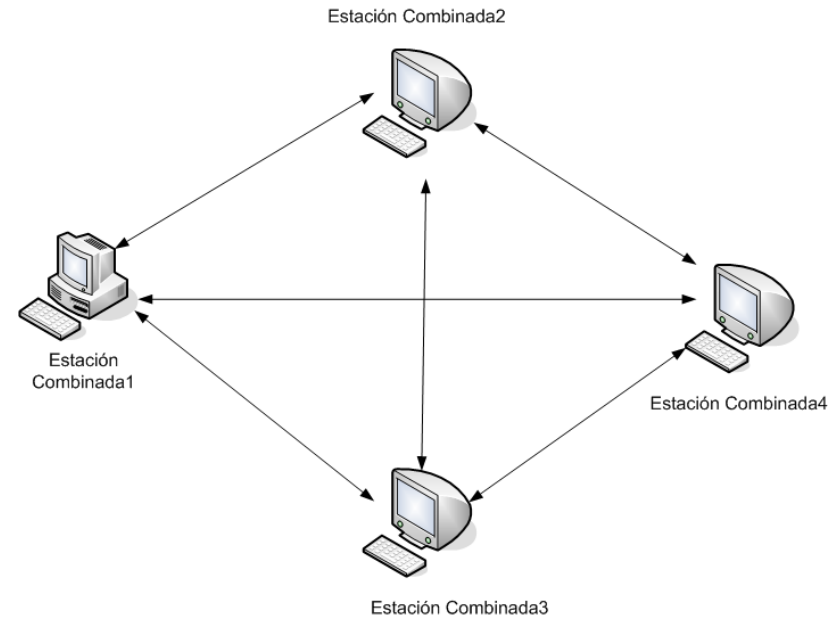


# Configuraciones del Enlace

## Modos de Transferencia de Datos



Modo No Balanceado

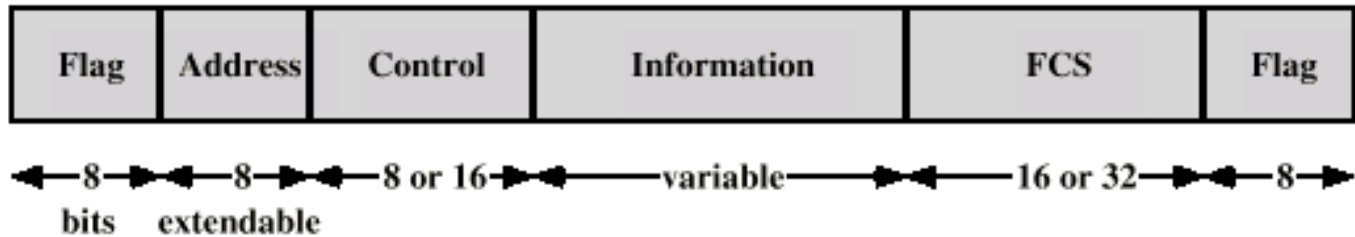


Modo Balanceado

# Modos de transferencia de datos

	<b>NRM (Normal Response Mode)</b>	<b>ARM (Asynchronous Response Mode)</b>	<b>ABM: Asynchronous Balanced Mode</b>
Configuración Enlace	No Balanceada	No Balanceada	Balanceada
Control Comunicación	Estación Primaria (envía órdenes. Las secundarias txten sólo cuando reciben una orden)	Estación primaria (pero acepta solicitudes de las secundarias)	Cualquier estación puede iniciar la comunicación
Tipos de Enlace	Punto-Punto, Punto-Multipunto	Punto-Punto, Punto-Multipunto	Punto-Punto (Full Duplex)

# Protocolo HDLC- Formato de la trama



Original Pattern:

1111111|11111110|11111110|11111110

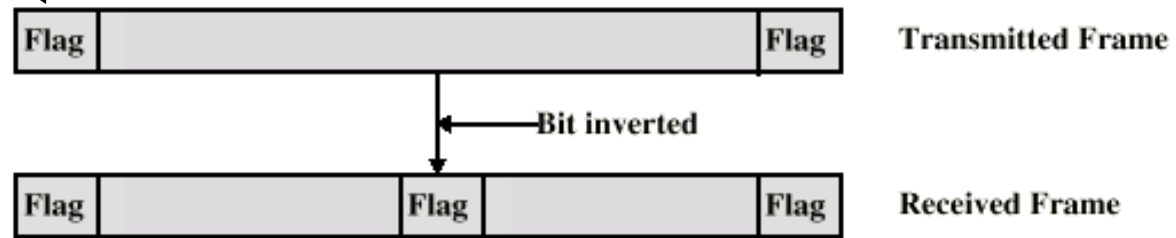
After bit-stuffing

11111101111110110111111010111111010

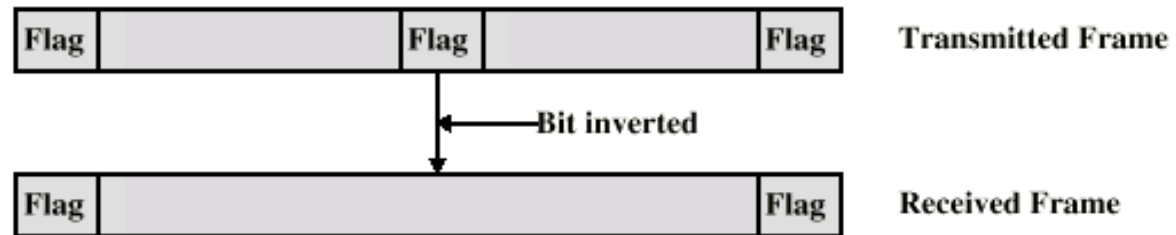
# Inserción de bits

7EH

(a) Example

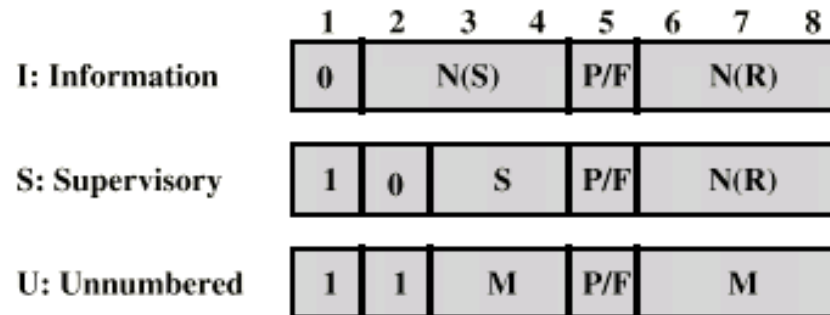


(b) An inverted bit splits a frame in two



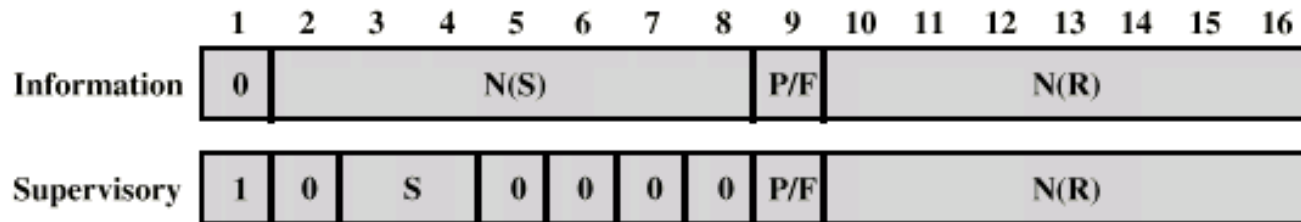
(c) An inverted bit merges two frames

# HDLC- Campo de control



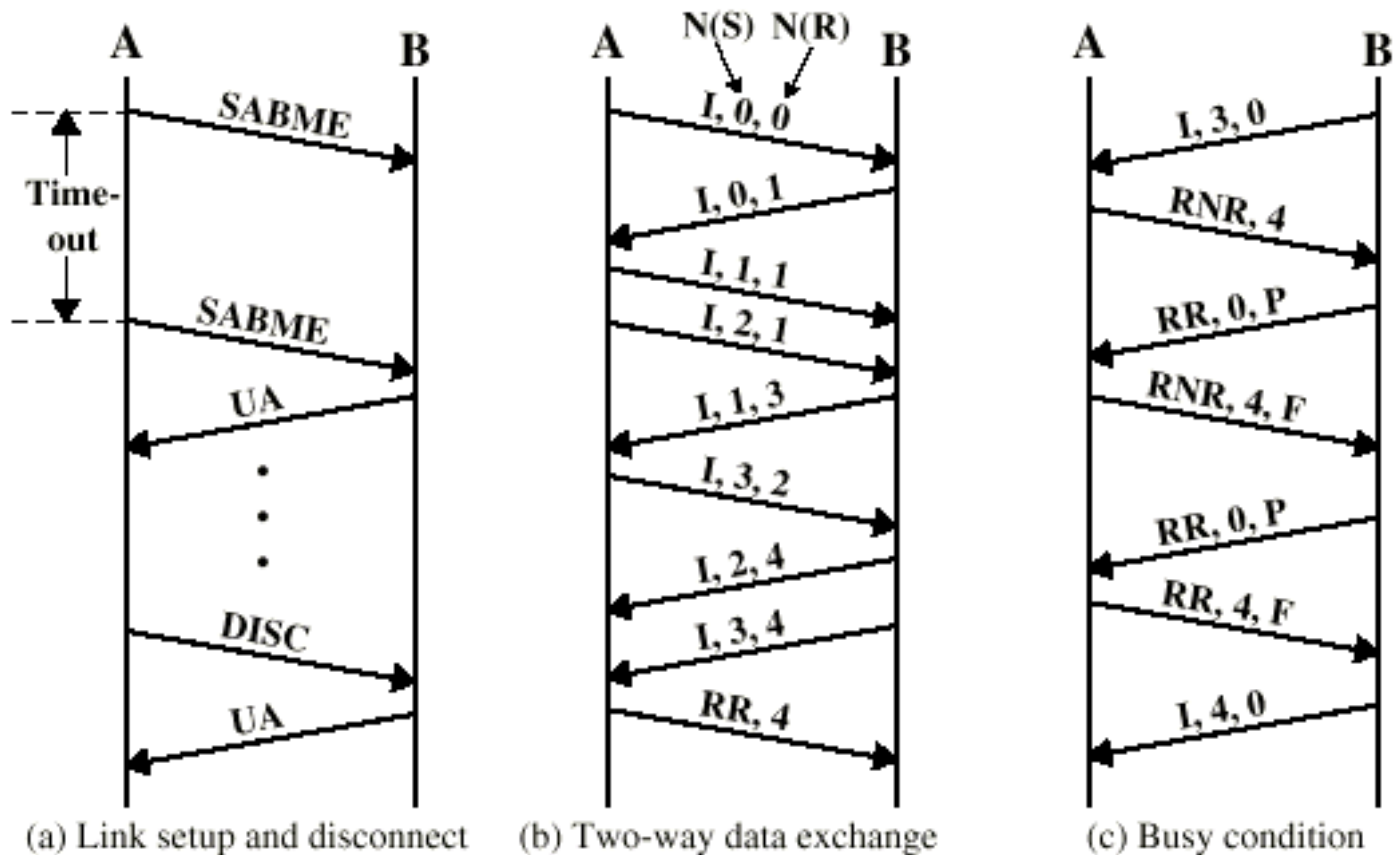
**N(S)** = Send sequence number  
**N(R)** = Receive sequence number  
**S** = Supervisory function bits  
**M** = Unnumbered function bits  
**P/F** = Poll/final bit

(c) 8-bit control field format

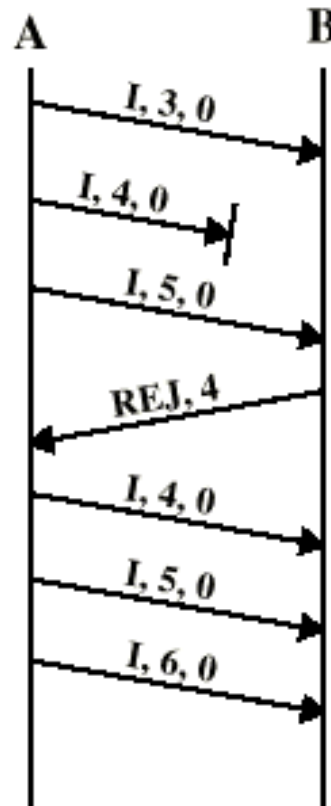


(d) 16-bit control field format

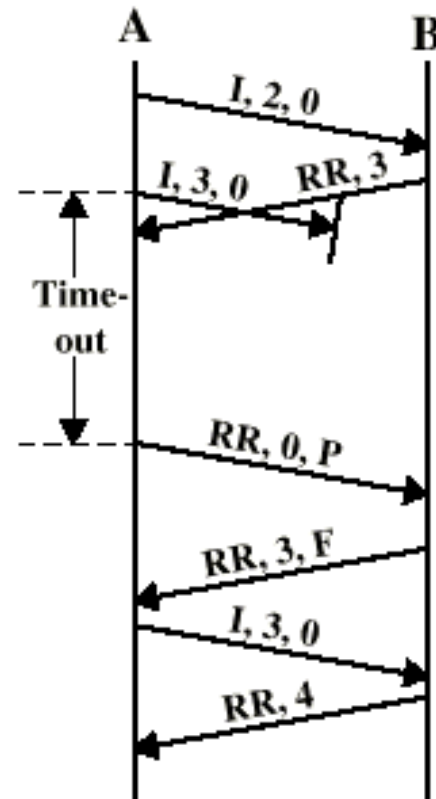
# Ejemplos de Operación (1)



# Ejemplos de operación (2)

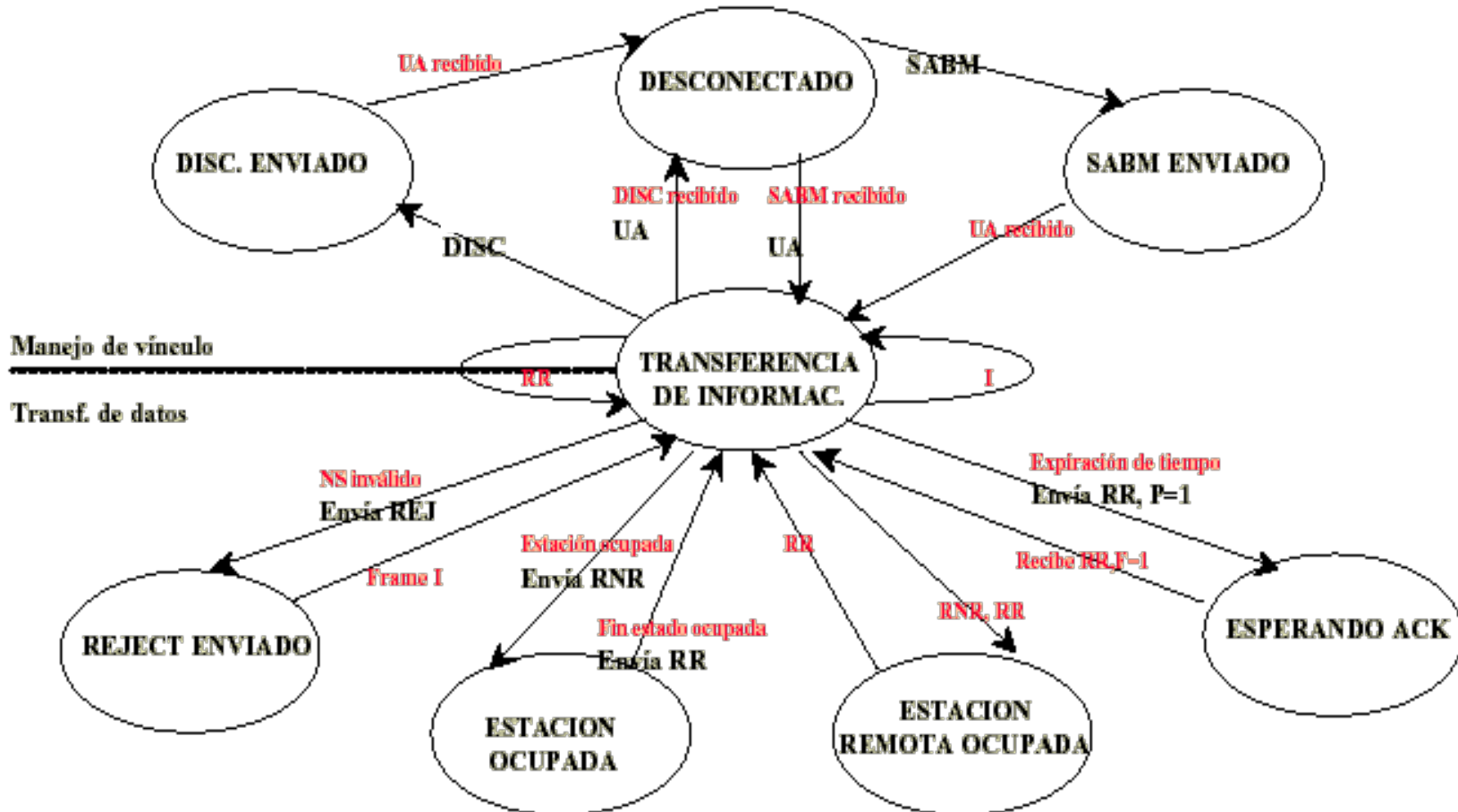


(d) Reject recovery



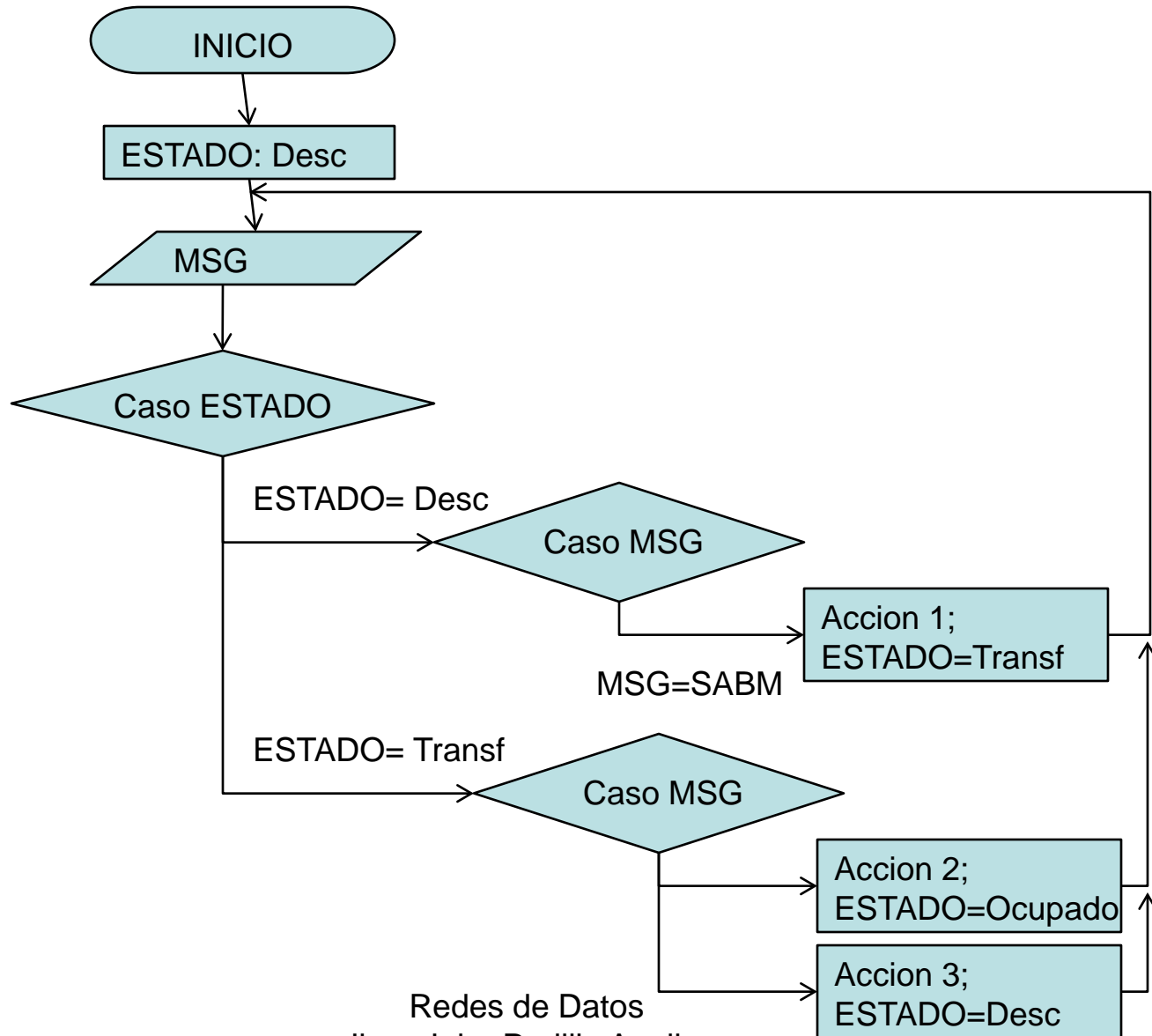
(e) Timeout recovery

# Diagrama de transición de estados





# Programa típico



# Trabajo para casa

- Diseñar un protocolo de comunicaciones basado en el protocolo HDLC que permita la comunicación de 20 estaciones remotas de recolección de información atmosférica con una estación central que contiene una base de datos y genera informes a partir de la información recolectada de las estaciones remotas. La comunicación se hace por medio de radio-teléfonos que permiten la comunicación en un solo sentido a la vez. Especificar la arquitectura de la red (organización), modo de operación de la red, métodos de detección de errores y control de errores, los mensajes que se transmitirán y su significado, los formatos de las diferentes tramas, el mapa de las direcciones a usar, los diagramas de mensajes que describen las diferentes situaciones que se puedan presentar en la comunicación.