

Comunicación de Datos I



PRÁCTICO 2 **NIVEL DE ENLACE DE DATOS**

DELIMITADOR DE FRAMES
CONTROL DE ERRORES
CONTROL DE FLUJO

Framing



- Al enviar información, en vez de enviar un stream continuo de información, se utilizan frames.
- Así es posible detectar errores por frame, y de existir un error, se puede retransmitir sólo el frame problemático.
- Para delimitar los frames existen diversos métodos:

- Cuenta de caracteres.



- Delimitación de frame por caracteres.

- Delimitación de frame por bits.

- Por lo general se utilizan combinaciones de éstos...

Framing



Determine, para la siguiente información a enviar, cómo será transportada a través del vínculo de transmisión y cómo será entregada al receptor luego de eliminar los caracteres producidos por el proceso de framing:

23-7D-03-5E-45-5E-5E-7E

Delimitadores de bloques: **7E**

Caracter de transparencia: **7D**

Procedimiento en el emisor

Cuando se quiere transmitir **7E**, se envía **7D-5E**

Cuando se quiere transmitir **7D**, se envía **7D-5D**

Cuando se quiere transmitir **00..1F**, se envía **7D-caracter+20**

Framing



Se desea transmitir como dato:

23-7D-03-5E-45-5E-5E-7E

El emisor agrega inicio de bloque, transparencia, y fin de bloque

7E- 23-7D-5D-7D-23-5E-45-5E-5E-7D-5E-7E

El receptor busca inicio de bloque, caracteres de transparencia, y fin de bloque

23-7D-03-5E-45-5E-5E-7E

Framing



Utilizando delimitación de bloque por secuencia de bits,
para los bits de información:
101101111101011111110101 determine como será
transportada.

Delimitación: Se utilizan flags de tipo: **01111110**

Transparencia: Cuando en los datos aparecen 5 unos
seguidos se le agrega un 0 luego del 5to bit.

EJ. 00111111010 --> 0011111011010

Entonces: 101101111101011111110101 será transportado
como:

01111110 101101111100101111101110101 **01111110**

Control de Errores



- Códigos correctores de errores
- Códigos detectores de errores
 - Bit de paridad
 - Códigos cíclicos
 - Utilizados en Ethernet, 802.11, HDLC, etc. (CRC)
 - Checksum IP
 - Muy simple y rápido (aritmetica módulo 1)
 - Cálculo incremental
 - Indiferente a orden de bytes en palabras
 - Puede realizarse en paralelo
 - Capacidad de detección limitada
 - Utilizado en las niveles IP y transporte de la arquitectura TCP/IP (IP, ICMP, UDP, TCP)

Checksum IP



- **Generación del checksum**
 - Agrupar los bytes en grupos de a 2 (16 bits). Si hay un número impar rellenar con 0
 - Poner campo checksum en cero
 - Suma módulo 1 con acarreo de los grupos de 16 bits
 - Complementar a 1 el resultado y colocarlo en el campo checksum
- **Chequeo del checksum**
 - Sumar en complemento a 1 la totalidad de grupos de 16 bits con acarreo (incluido checksum)
 - Si el resultado es todos 1, el resultado es correcto

Checksum de acuerdo al Protocolo



VERS	HLN	TIPO SERV.	LONG. TOTAL	
IDENTIFICACION			FLA GS	OFFS. FRAGMENTO
TTL	PROTOD.		CHECKSUM HEADER	
DIRECCION IP ORIGEN				
DIRECCION IP DESTINO				
OPCIONES				PAD
DATOS				

IP

PORT ORIGEN (16)			PORT DESTINO (16)		
NUMERO DE SECUENCIA					
NUMERO DE ACK					
LNG HDR (4)	RES (6)	UAPRSF RCSSM CKEINN	LONG. VENTANA (16)		
CHECKSUM (16)			PUNTERO D. URGENT. (16)		
OPCIONES (0 ó MAS GRUPOS DE 32 BITS)					
DATOS (OPCIONAL)					

TCP

TIPO (8)	CODIGO (8)	CHECKSUM (16)
INFORMACION DEPENDIENTE DEL TIPO		

ICMP

PORT ORIGEN (16)	PORT DESTINO (16)
LONGITUD (16)	CHECKSUM (16)
DATOS	

UDP

DIRECCION IP ORIGEN (32)		
DIRECCION IP DESTINO (32)		
CERO (8)	PROT (8)	LONG. TCP o UDP (16)

Pseudoheader

- IP: cubre sólo el header IP
- ICMP: cubre el frame ICMP completo
- UDP: cubre todo el segmento UDP más el pseudoheader
- TCP: cubre todo el segmento TCP más el pseudoheader

Cálculo del Checksum del datagrama IP



- Separar el header del frame IP

4500 003C: Vers, Hlen, Tipo, Long
0A1C 4000: Ident., Flags, Offset
FF06 XXXX: TTL, Protoc., Checksum
A8B0 0319: Dirección Origen
A8B0 036C: Dirección Destino

- Poner el campo checksum en 0
- Sumar módulo 1 con acarreo
- Complementar el resultado
- Colocar el resultado final en el campo ChecsumHeader

VERS	HLEN	TIPO SERV.	LONG. TOTAL	
IDENTIFICACION			FLA GS	OFFS. FRAGMENTO
TTL	PROTOD.		CHECKSUM HEADER	
DIRECCION IP ORIGEN				
DIRECCION IP DESTINO				
OPCIONES				PAD
DATOS				

Control de Flujo



- El control de flujo es necesario para no saturar al receptor de uno a más emisores
- Se utilizan mecanismos de confirmaciones
- Existen distintas técnicas
 - Parada y Espera
 - Ventana Fija
 - Ventana Deslizante

Parada y Espera

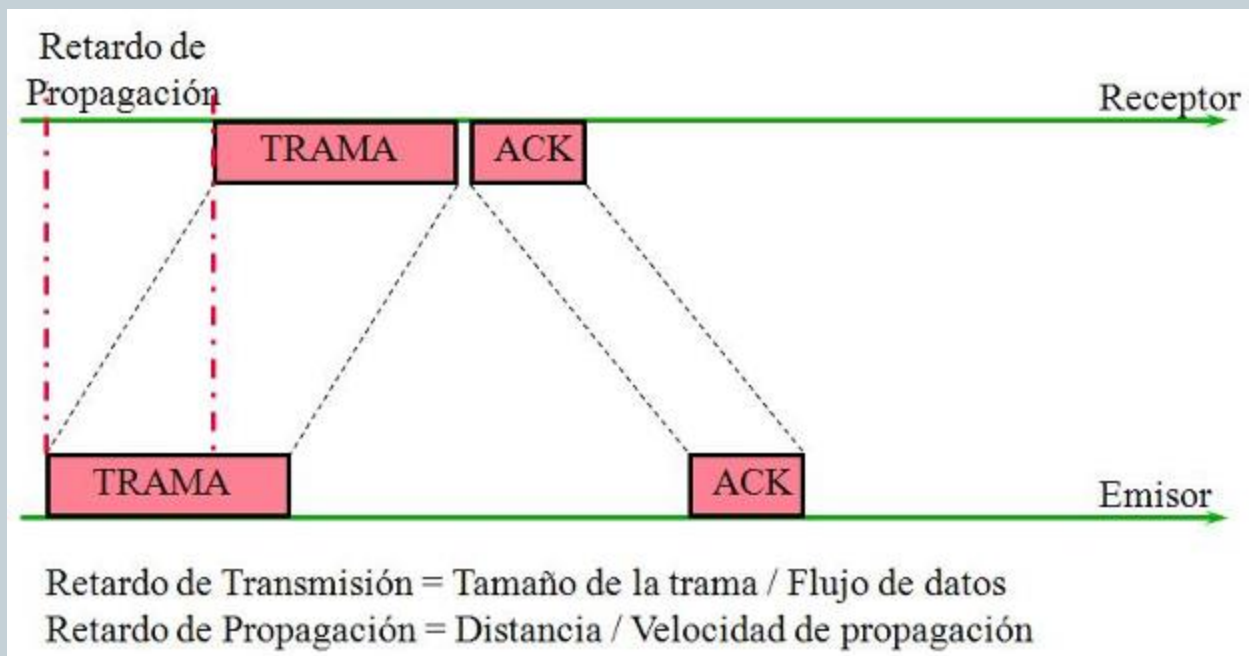


- El emisor envía una trama cada vez
- Utilizado para vínculos bidireccionales alternativos
- Sencilla implementación pero poco eficiente
- Hasta que no recibe confirmación (ACK=Acknowledgement) no envía la siguiente
- El proceso continúa hasta enviar el fin de transmisión
- En caso de recibir un NACK (rechazo de la recepción NO Acknowledgement) se reenvía el paquete anterior.

Parada y Espera



- Mecanismo de transmisión

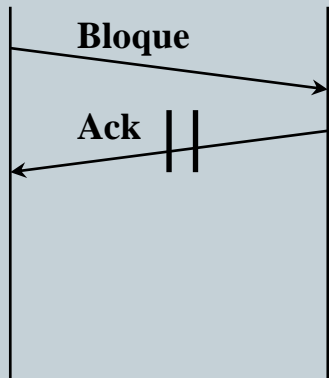


$$e = Tb / (Tb + Tok + 2 * dp)$$

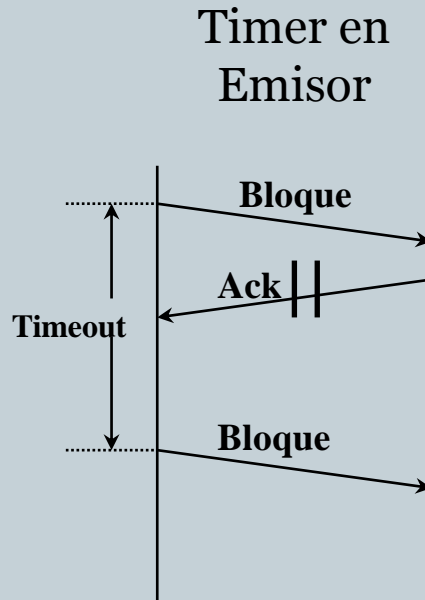
Parada y Espera



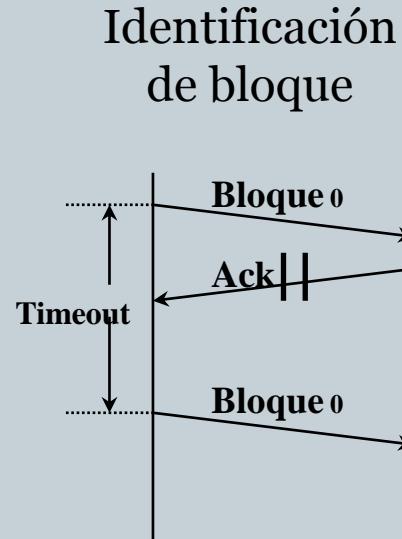
- Posibles situaciones de error



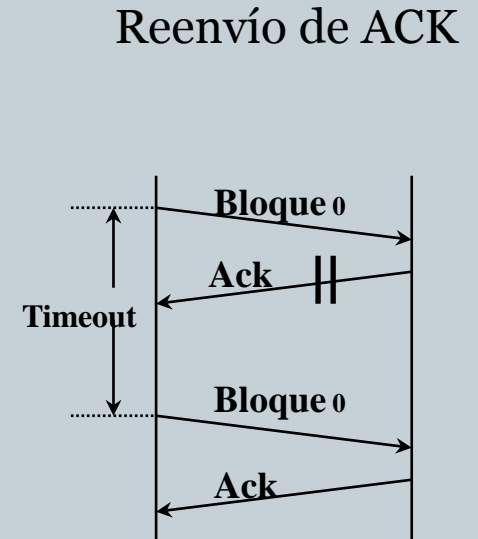
Deadlock
Emisor



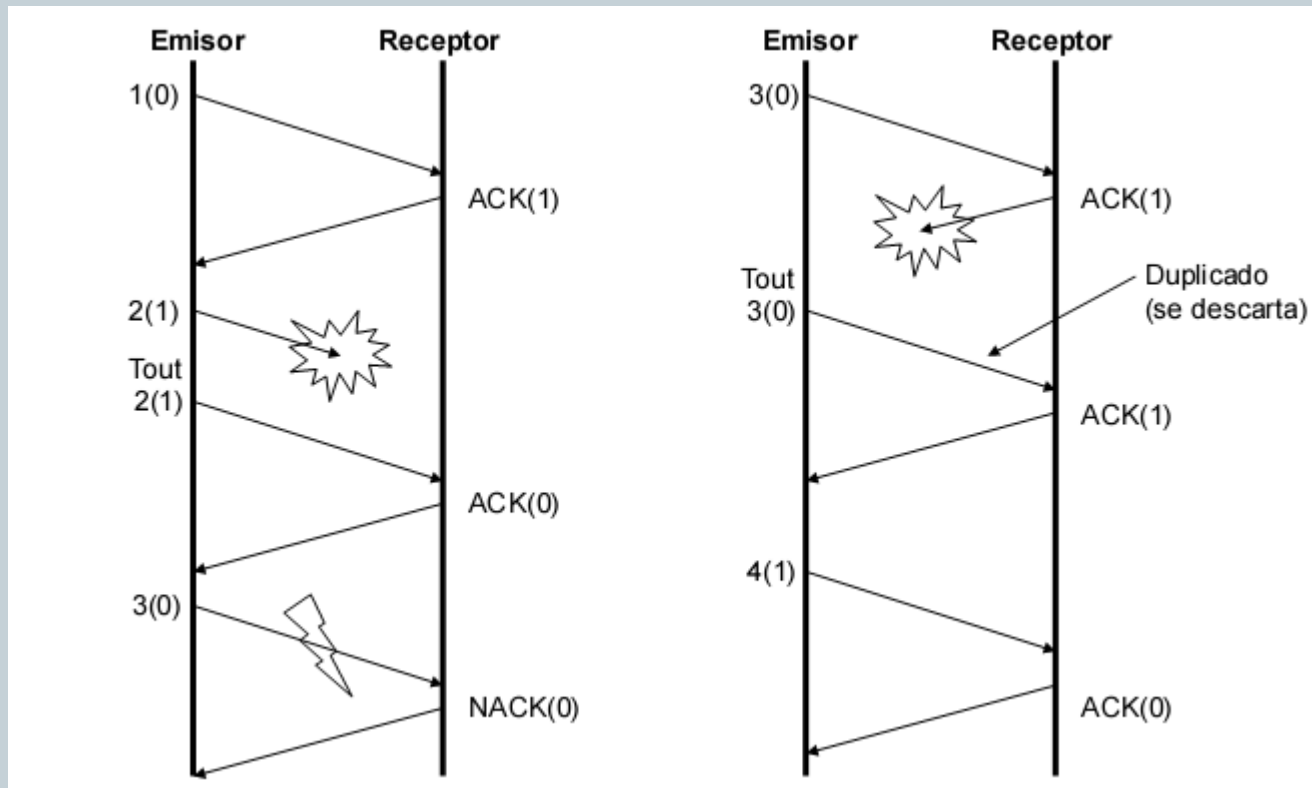
Bloque
Duplicado



Deadlock
Receptor



Parada y Espera



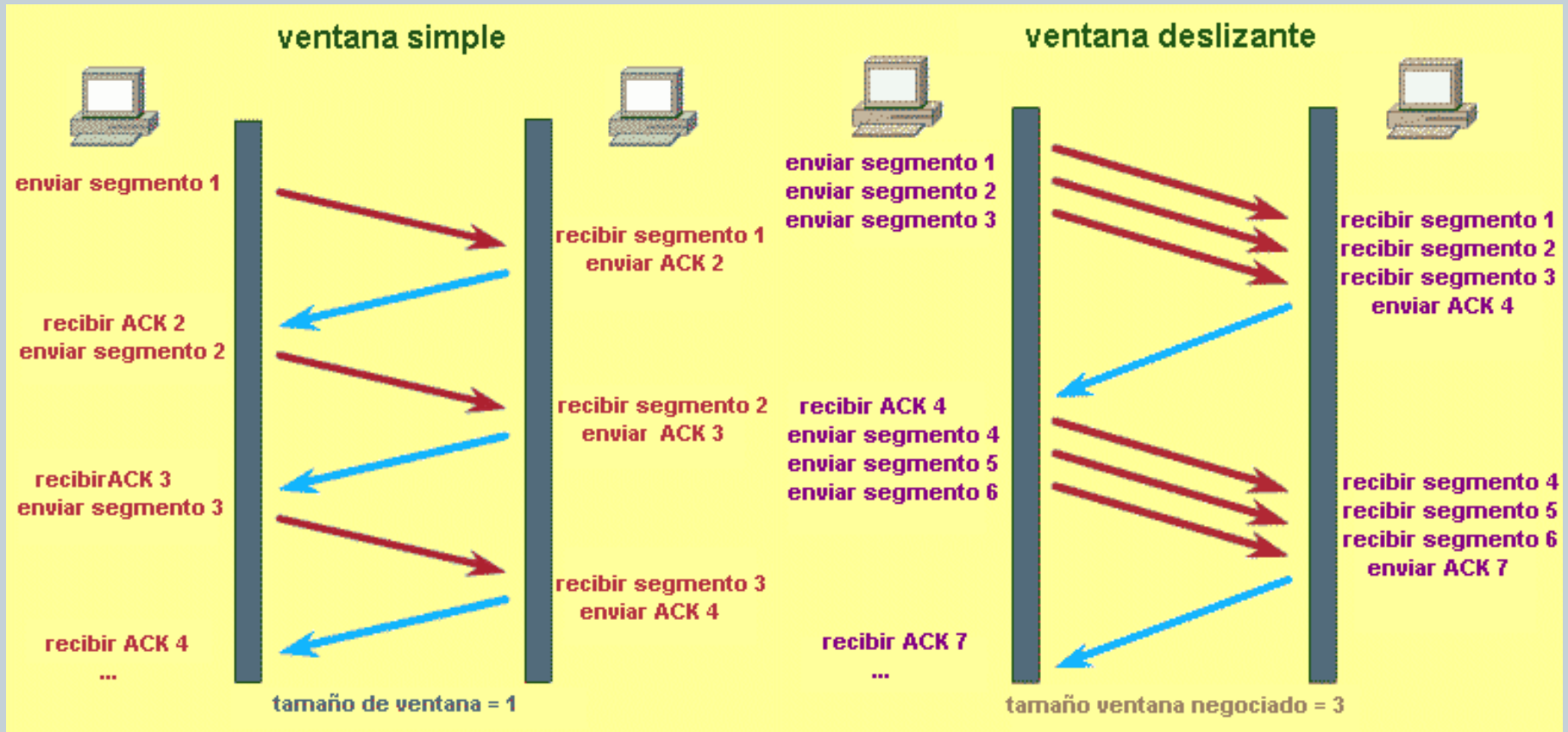
Ventana Fija



- Se envía una cantidad W de paquetes (tamaño de ventana), y se espera a que se reciba el ACK de todos los paquetes.
- Un permiso para todos los bloques de la ventana
- Puede utilizarse en vínculos bidireccionales alternativos.
- Requiere de mayor capacidad (W buffers)
- Aumenta eficiencia de Parada y Espera

$$e = Tb * W / (Tb * W + Tok + 2 * dp)$$

Parada y Espera vs. Ventana Fija



¿Preguntas?

