

Tema 4: Protocolos de comunicación punto a punto

Protocolos ARQ
Protocolos de control de flujo
Los protocolos TCP y UDP
Interfaz con la aplicación

Tema 4: Protocolos de comunicación punto a punto

Protocolos ARQ
Protocolos de control de flujo
Los protocolos TCP y UDP
Interfaz con la aplicación

Protocolos ARQ

Una vez tenemos un método para detectar errores, nos falta determinar el mecanismo que nos permita pedir retransmisiones de la información

En general podemos usar 3 métodos:

Stop & Wait

Go-Back-N

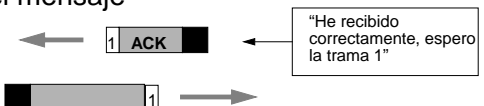
Selective Repeat

Stop & Wait:

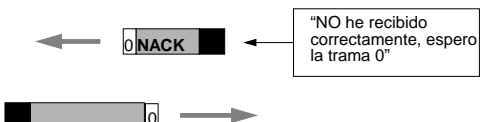
Enviamos una trama numerada y con un CRC



Si la recepción ha sido correcta contestamos con una trama de confirmación (ACK), también numerada y protegida con un CRC. El Tx. nos enviará la siguiente trama. del mensaje

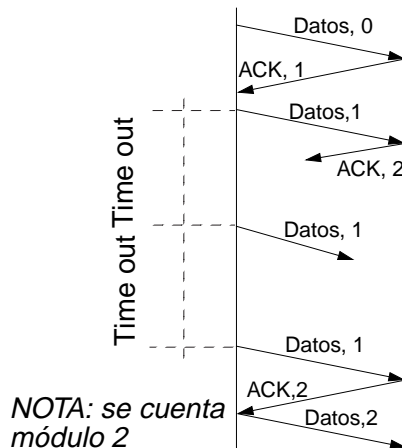


Si en cambio ha habido errores, contestamos con una trama que indica error (NACK). En ese caso el Tx. nos reenviará la misma trama.



Problema: ¿Qué pasa si alguna trama (de datos, ACK o NACK) se pierde?

La solución está en usar un *temporizador*: Si transmitimos una trama, y vence el temporizador sin haber recibido ACK o NACK de dicha trama, la volvemos a transmitir



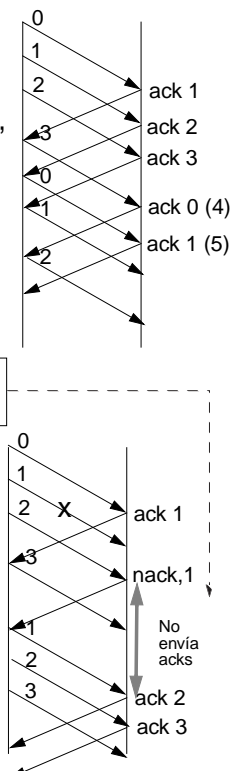
Escogemos un valor para N (ej, N=3)

Podemos tener "volando" sin confirmar hasta un máximo de N tramas

Las numeramos módulo (N+1)

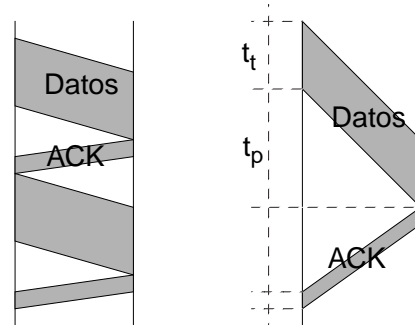
También podría seguir enviando nack, 1...

Si recibimos un NACK o vence un temporizador, retransmitimos a partir de la última de las tramas que ha sido confirmada



Go-Back-N

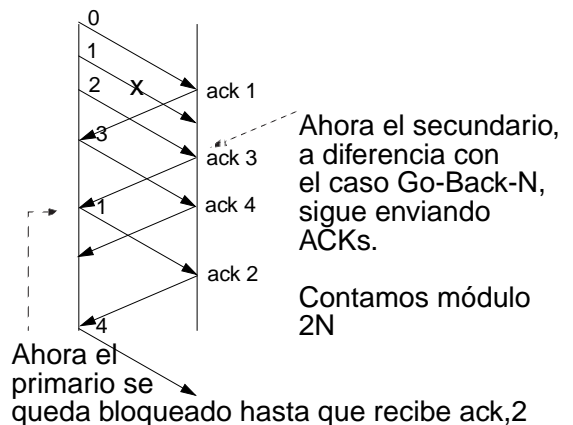
En enlaces de alta velocidad o en enlaces de larga distancia Stop&Wait es ineficiente:



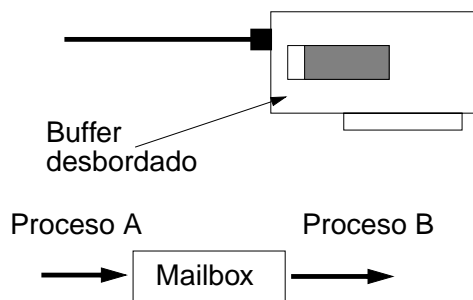
La idea es no esperar a tener una trama confirmada para enviar la siguiente, sino "adelantar" trabajo...

Selective-repeat

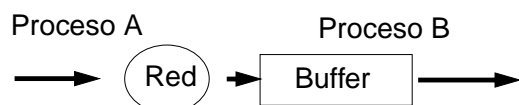
En este caso solo retransmitimos los paquetes de los que recibimos un NACK o para los que vence un temporizador (ej: N=3)



El receptor debe reordenar los paquetes !

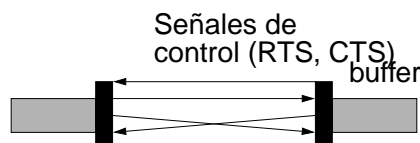


Si el consumidor es más lento que el productor, el proceso A quedará bloqueado para evitar desbordamiento del buffer



Si el consumidor es más lento que el productor, el proceso A debe quedar bloqueado para evitar desbordamiento del buffer. Al estar en otra máquina, debe haber un mecanismo que informe de esta condición (Control de flujo)

Métodos hardware:



Métodos software:

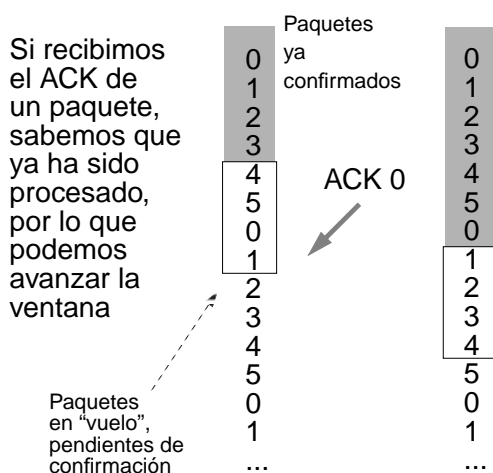


Envío de caracteres XON/XOFF

Mecanismo de ventana

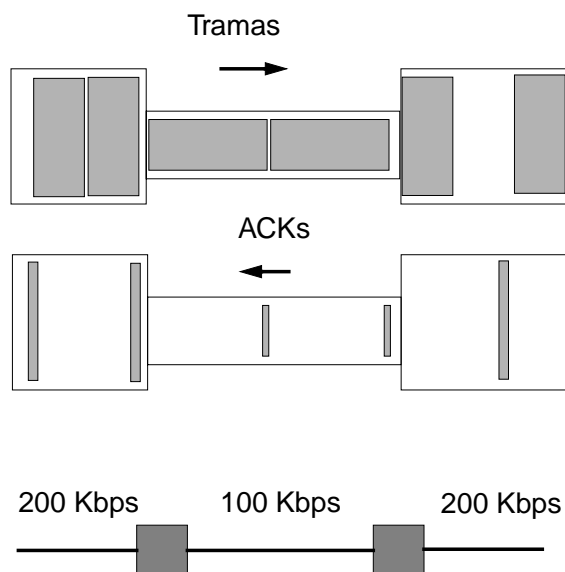
Mecanismo de ventana:

El Tx. no puede tener en "vuelo" más de W tramas sin confirmar



El sistema A podría tener un mayor control si por ejemplo pudiera variar el valor de W

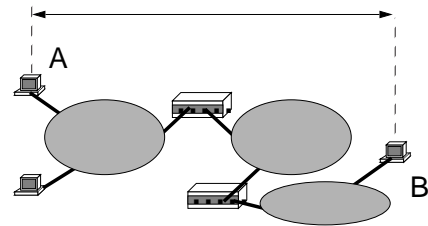
Autorregulación de la velocidad:



Tema 4: Protocolos de comunicación punto a punto

Protocolos ARQ
Protocolos de control de flujo
Los protocolos TCP y UDP
Interfaz con la aplicación

TCP ("Transmission Control Protocol")



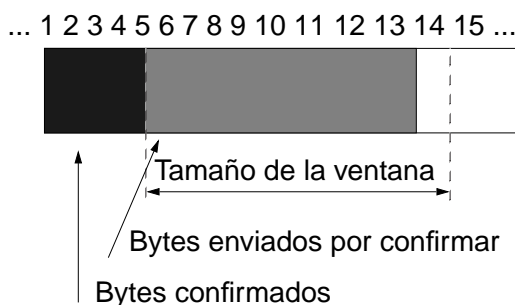
Es un protocolo extremo a extremo. Utilizando TCP aseguramos que la información llega de forma correcta (es decir, la información incorrecta es reenviada) y en el orden correcto.

Además ejerce un control de flujo por ventana de tamaño variable que permite al consumidor no ser inundado por la información del productor. También protege a la red de congestión.

Para conseguir lo anterior necesitamos usar un protocolo orientado a la conexión.

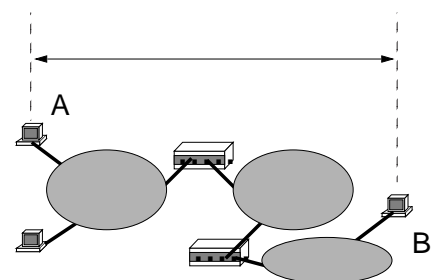
Protocolo de ventana

NO numeramos los paquetes sino los bytes.

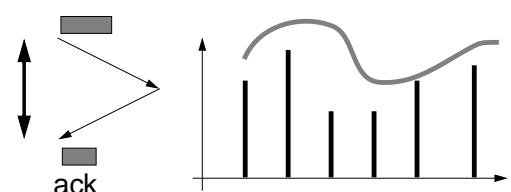


El tamaño de la ventana es variable.

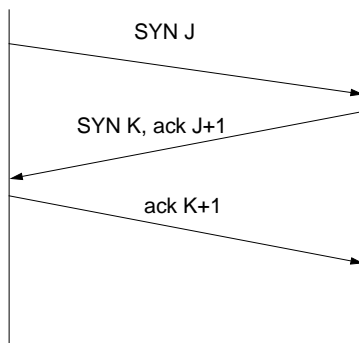
Problema: ¿Cómo fijar los valores de time-out?



Dependiendo de los retardos que sufren los ACKs, vamos fijando el valor del temporizador de forma adaptativa.



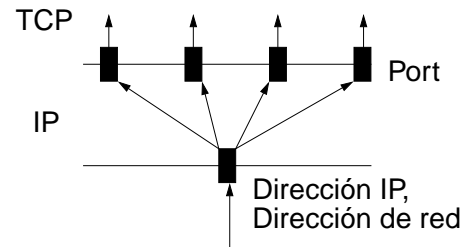
Establecimiento de la conexión ("Three Way Handshaking")



En este establecimiento de la conexión fijamos los números iniciales para los Bytes que intercambiaremos

Existe también un mecanismo para cerrar la conexión (FIN N; ACK N+1)

La conexión se identifica mediante la dirección IP del host y un número de puerto ("port") de 16 bits.



UDP ("User Datagram Protocol")

Es un protocolo NO orientado a la conexión. Añade una mínima funcionalidad a IP (Detección de error, Número de Puerto).

Ejemplo: ¿Qué pasa cuando con un browser consultamos una página web?

