Transporte fiable
Selective repeat

Area de Ingeniería Telemática
http://www.tlm.unavarra.es

Arquitectura de Redes, Sistemas y Servicios
Grado en Ingeniería en Tecnologías de Telecomunicación, 2º

Temario

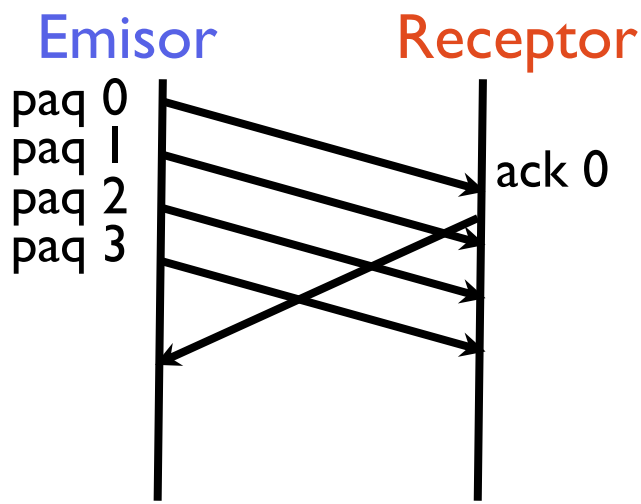
1. Introducción
2. Arquitecturas de conmutación y protocolos
3. Introducción a las tecnologías de red
4. Control de acceso al medio
5. Conmutación de circuitos
6. Transporte fiable
7. Encaminamiento

Temario

1. Introducción
2. Arquitecturas de conmutación y protocolos
3. Introducción a las tecnologías de red
4. Control de acceso al medio
5. Conmutación de circuitos
6. **Transporte fiable**
7. Encaminamiento

Protocolos más eficientes

- Para aumentar la eficiencia, se envían varios paquetes (ventana de paquetes) mientras llega el ACK
  Varios paquetes en la red por confirmar
  - Se usan más números de secuencia que 0 y 1
  - Emisor y receptor necesitarán buffer para varios paquetes
  - Varias políticas para reaccionar a los errores
    - Go-Back N
    - Selective repeat
    - Los dos son conocidos tambien como Ventana deslizante (sliding window)



Protocolo Go back-N

- Empezando por lo más simple:
- Número de secuencia en el paquete
- Se permite una ventana de N paquetes sin confirmar
- **Cada ACK confirma todos los paquetes anteriores (cumulative ACK)**
- Timeout al iniciar la ventana
- **Si caduca el timeout se retransmite la ventana**

- **Estado en el emisor**

  base= numero de secuencia mas bajo que aun no ha sido confirmado

  nextseqnum= siguiente numero de secuencia que voy a usar

  buffer con los paquetes enviados hasta que se confirmen y los descarte

- **Estado en el receptor**

  expectedseqnum= siguiente numero de secuencia que espero recibir

Resumen Go-back N

- Simple (¿sabríais programarlo?)
- Más eficiente que stop & wait
- ¿Cuál es la velocidad máxima?
- N s / RTT ?
- ¿y si el tiempo de transmisión no es despreciable?

- Mejoras fáciles a go-back N?

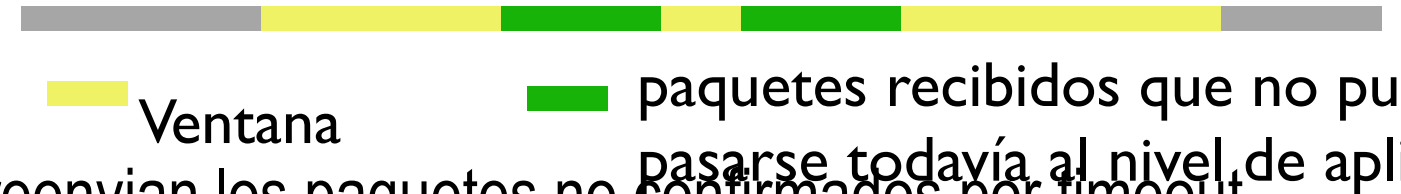
Mejoras a Go-back N

- Aprovechar los paquetes que llegan aunque se hayan perdido los anteriores?
  - Receptor más complicado. Necesita buffer para los paquetes recibidos pero no entregados a la aplicación
  - Reenviar si se reciben ACKs duplicados?

- Confirmar/rechazar paquetes por separado

Selective Repeat

- El receptor confirma (ACK) individualmente cada paquete
  - Mantiene en buffer los paquetes recibidos a la espera de reconstruir la secuencia y pasarlos al nivel de aplicación



- Se reenvian los paquetes no confirmados por timeout
  - **Timeout individual por cada paquete**
- Ventana de N paquetes que pueden enviarse sin recibir ACK

Eventos en el emisor (SR)

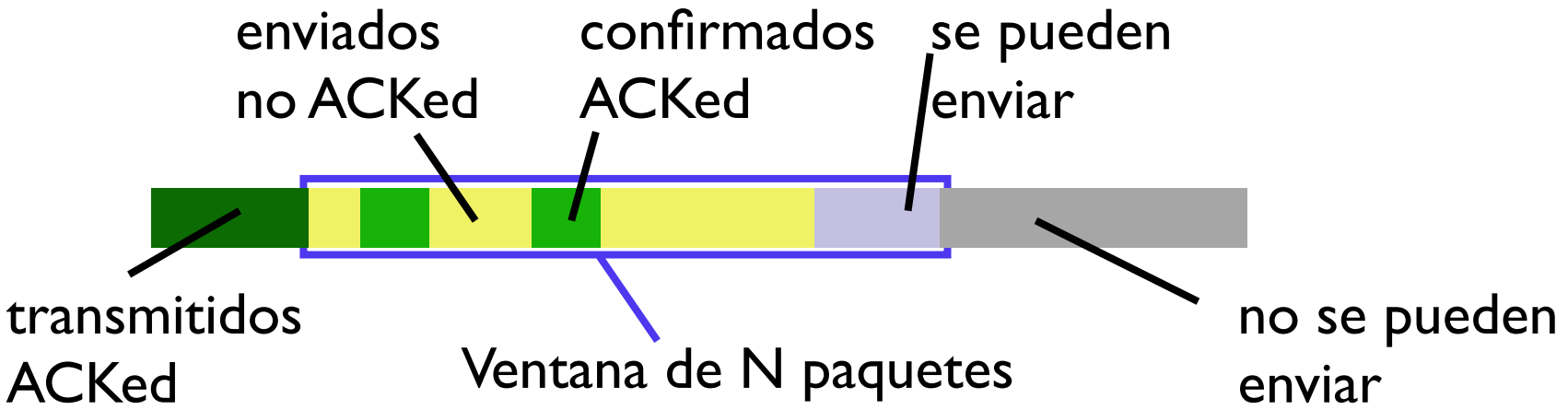
- **ACK recibido**
  - Se cancela el timeout de ese paquete
  - Si se puede avanzar la ventana se avanza hasta donde se pueda
  - Si la ventana avanza se envían paquetes nuevos si hay disponibles y se inician sus timeouts
- **Timeout de un paquete**
  - El paquete se reenvía
  - Se reinicia el timeout de ese paquete

Eventos en el receptor (SR)

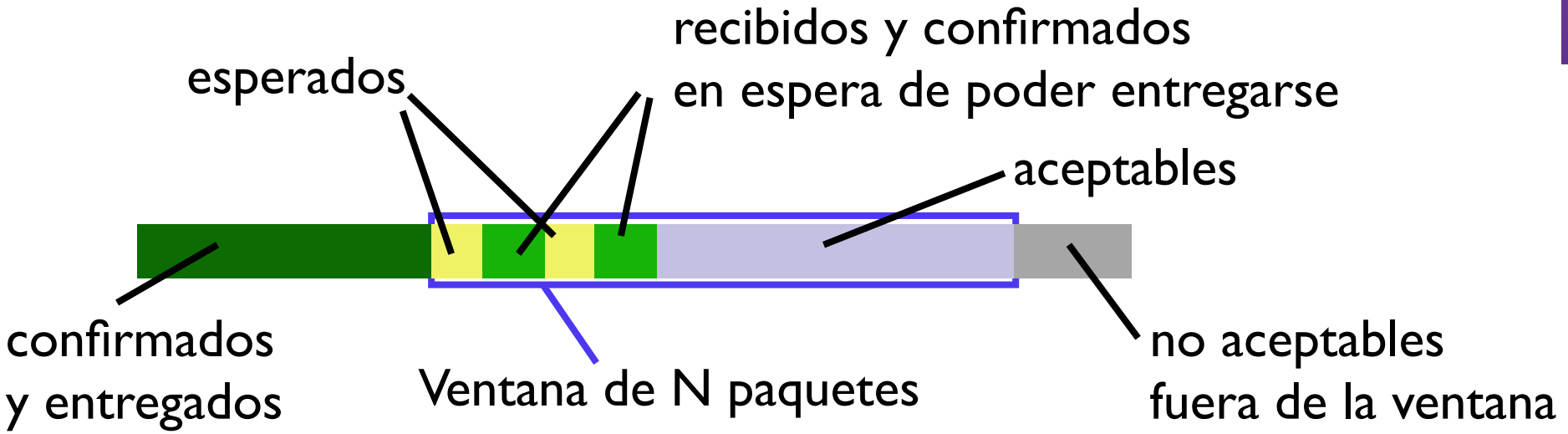
- El receptor tiene un buffer (y por tanto ventana) limitada
- **Recibidos datos en la ventana**
  - Se envía ACK de ese paquete
  - Se guarda el paquete en su posicion del buffer
  - Si el paquete es el esperado se entregan todos los paquetes continuos disponibles en la ventana al nivel superior y se avanza hasta donde se pueda
- **Recibidos datos anteriores a la ventana**
  - Se ignoran los datos
  - Se envía ACK de ese paquete
- **Recibidos datos posteriores a la ventana**
  - Se ignoran y no se envía nada

Selective Repeat

- Ventana deslizante del emisor

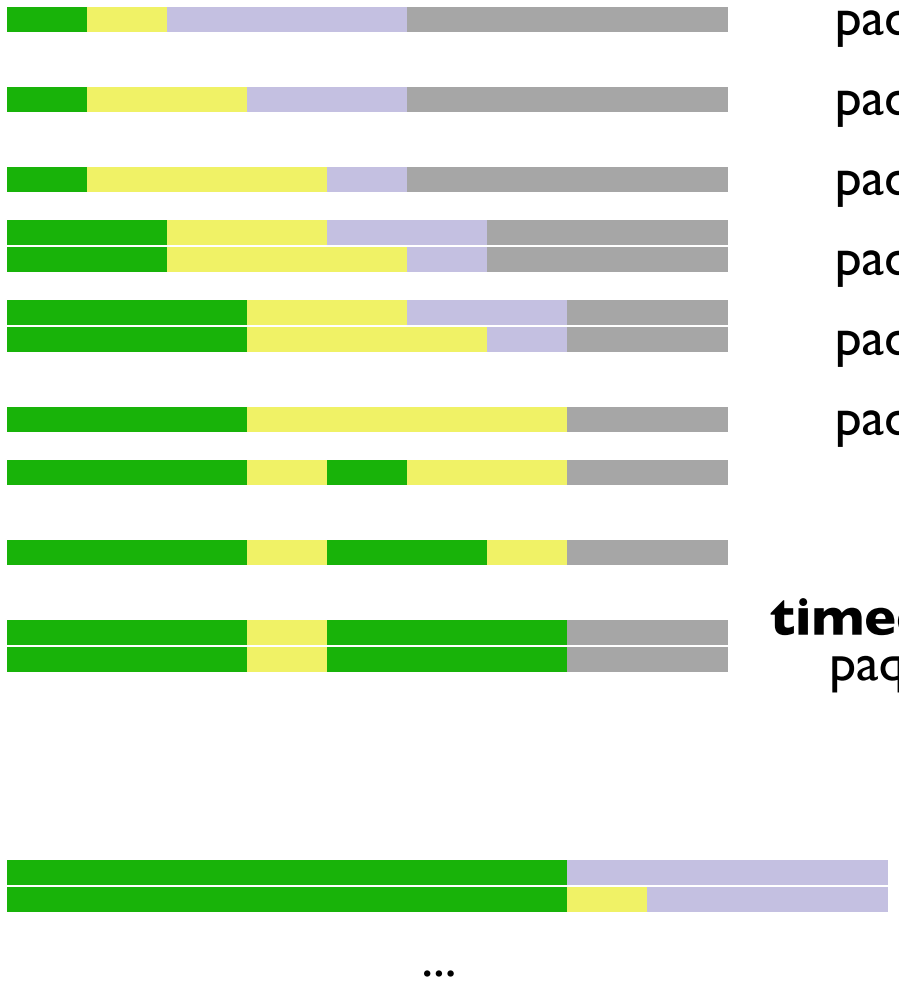


- Ventana deslizante del receptor

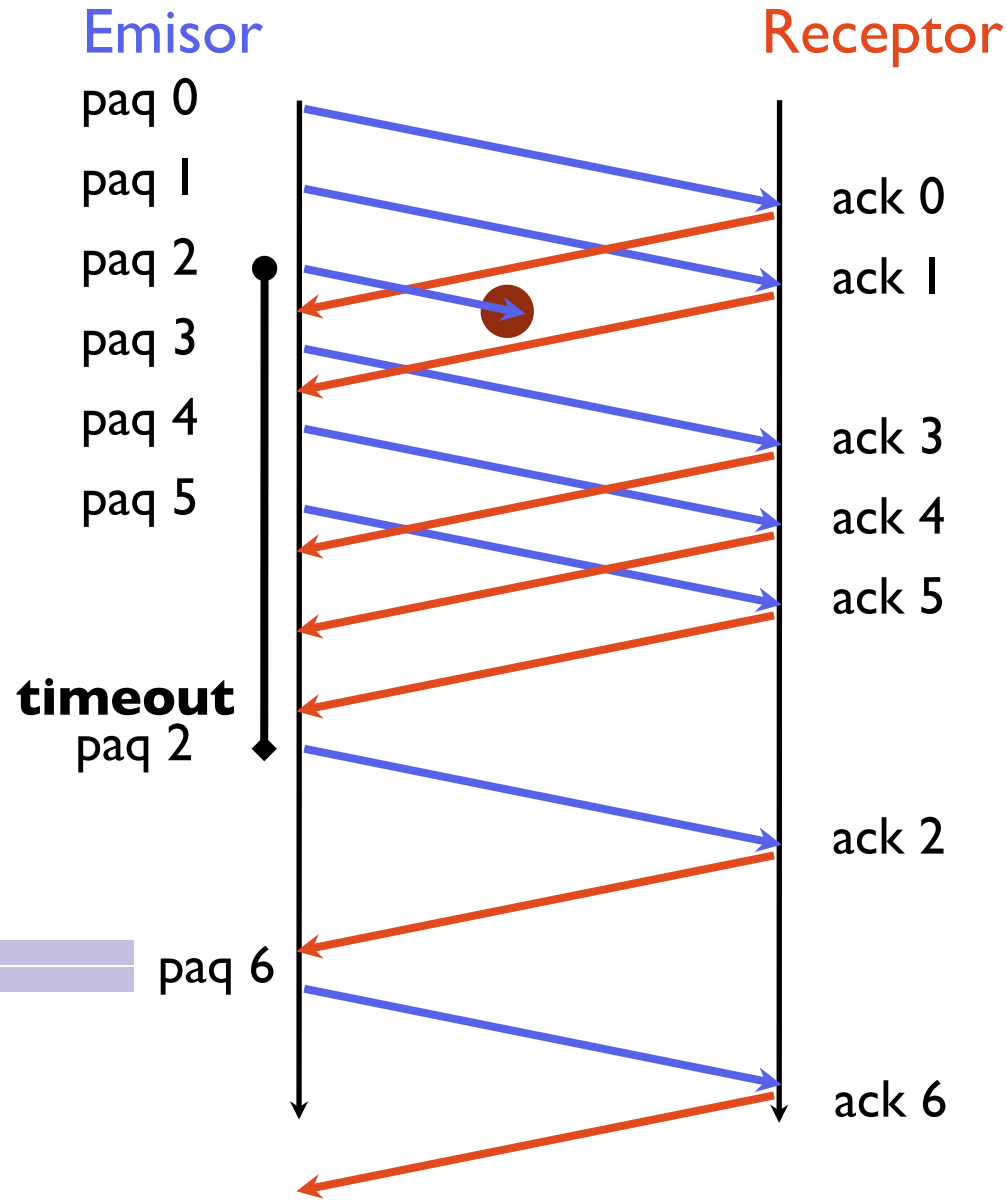


Selective repeat

Ventana de 4 paquetes



...



Aquí ACK no indica
el que espero recibir
sino indica el que **confirmo**

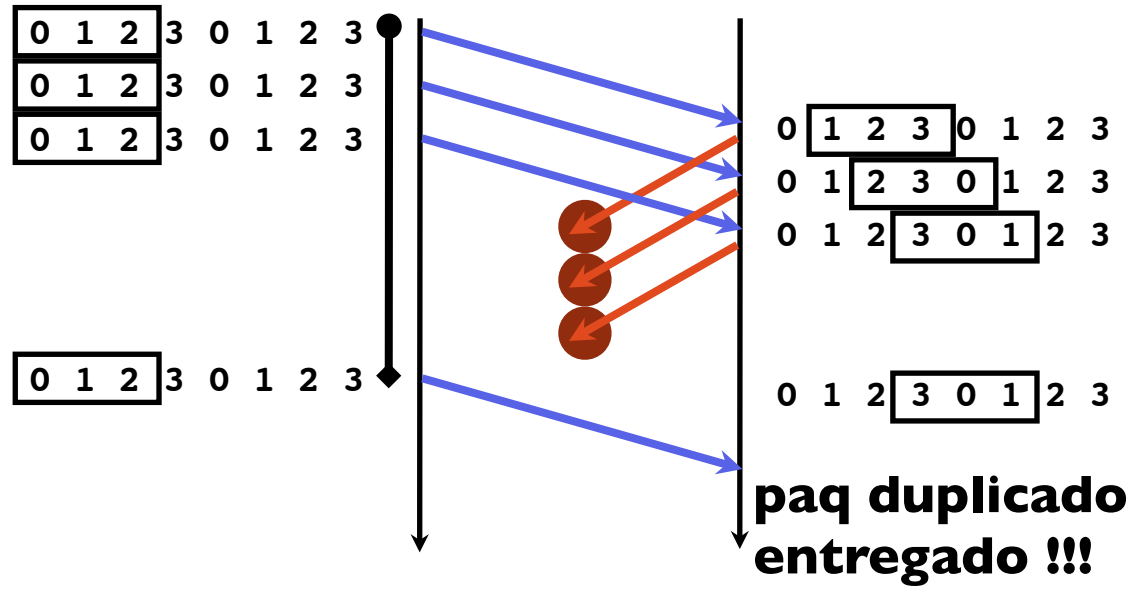
Problemas de selective repeat

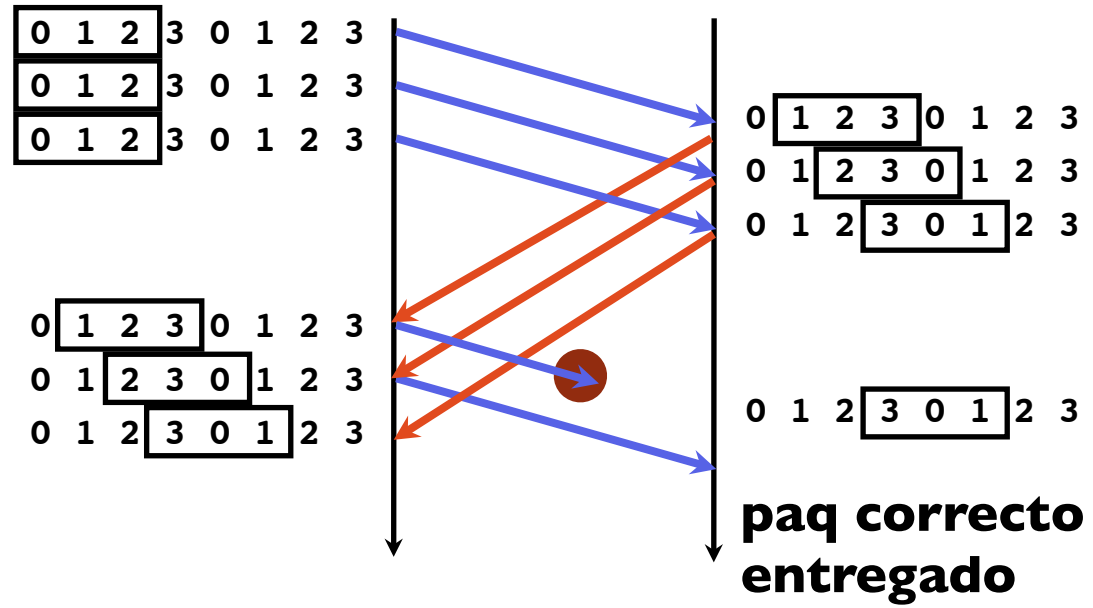
- Más complejo de programar: buffer de paquetes en el receptor, array de temporizadores en el emisor...

- Normalmente para implementar ventanas deslizantes el numero de secuencia es un campo de la cabecera
  - tiene un numero de bits limitado
  - las reglas anteriores hay que programarlas con contadores que se dan la vuelta
  - En go-back N es facil porque con n bits puedo hacer go-back 2^n-1
  - Pero en selective repeat hay algun problema...

El problema del selective repeat

- Número de secuencia finito
- Ejemplo
  - seq= {0,1,2,3}
  - N=3
- El receptor no puede notar la diferencia entre los dos escenarios
- Y entrega datos duplicados como buenos
- Que relación debe haber entre #secuencia y N?

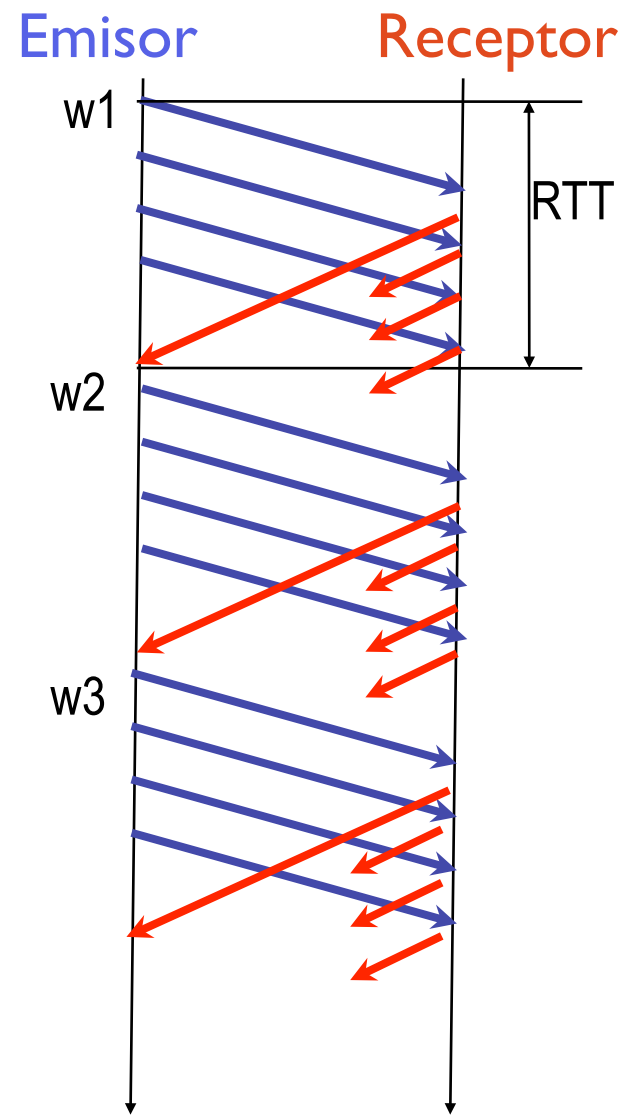




Eficiencia Selective Repeat

- Transporte fiable garantizado (en un escenario en el que se pierdan paquetes)
- Eficiencia mejor que stop-and-wait
- Parecida a la del go-back N
- El caso mejor es igual que en go-back N
- Si w>C*RTT
  - thrmax = C
- Si w<C*RTT
  - thrmax = w/RTT
- O bien
  - thrmax = min{ w/RTT, C }

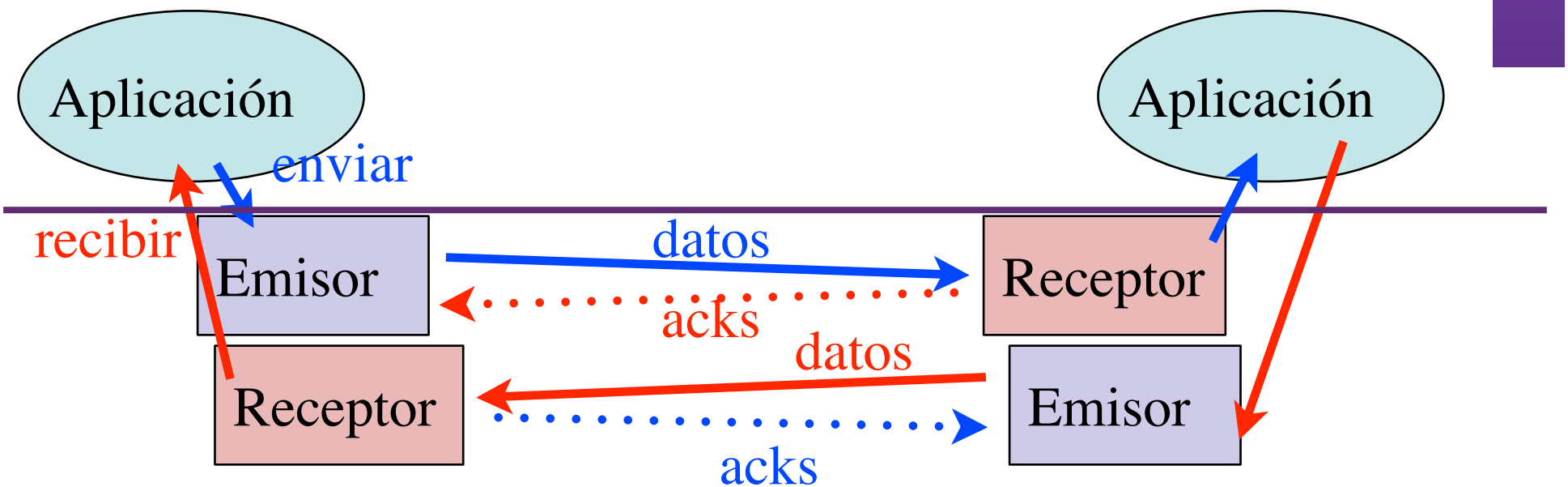
- Si hay errores depende de como sean estadísticamente los errores, es más difícil de analizar



Sesiones/conexiones

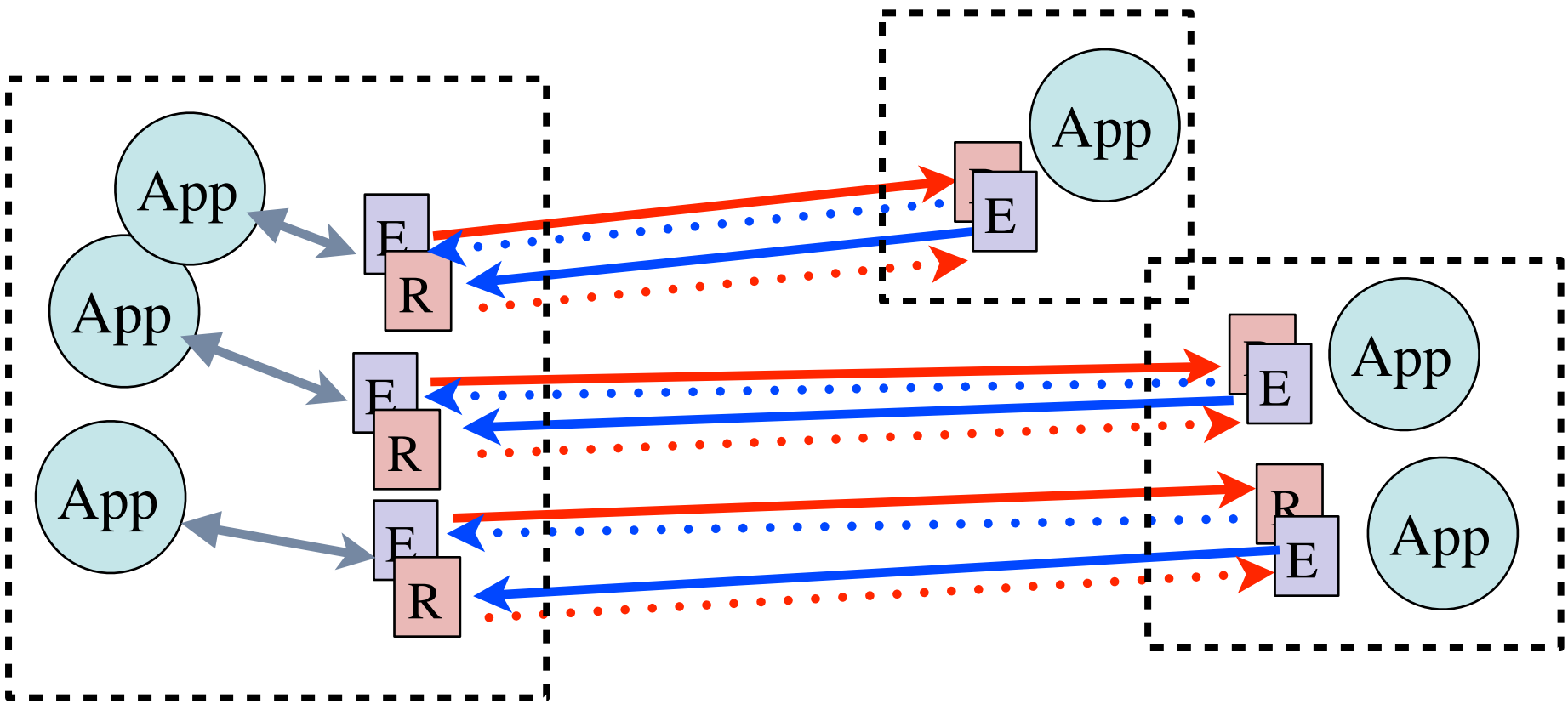
- Hasta ahora era un emisor y un receptor
- El nivel de transporte tiene que gestionar eso para diferentes sesiones de aplicación

  Los parametros de los protocolos stop&wait, go-back-N, selective repeat,
  - tienen sentido para una comunicacion entre un origen y un sentido
  - una conversación son dos sentidos emisor->receptor
  - un nivel de transporte tiene que gestionar varias conversaciones a la vez



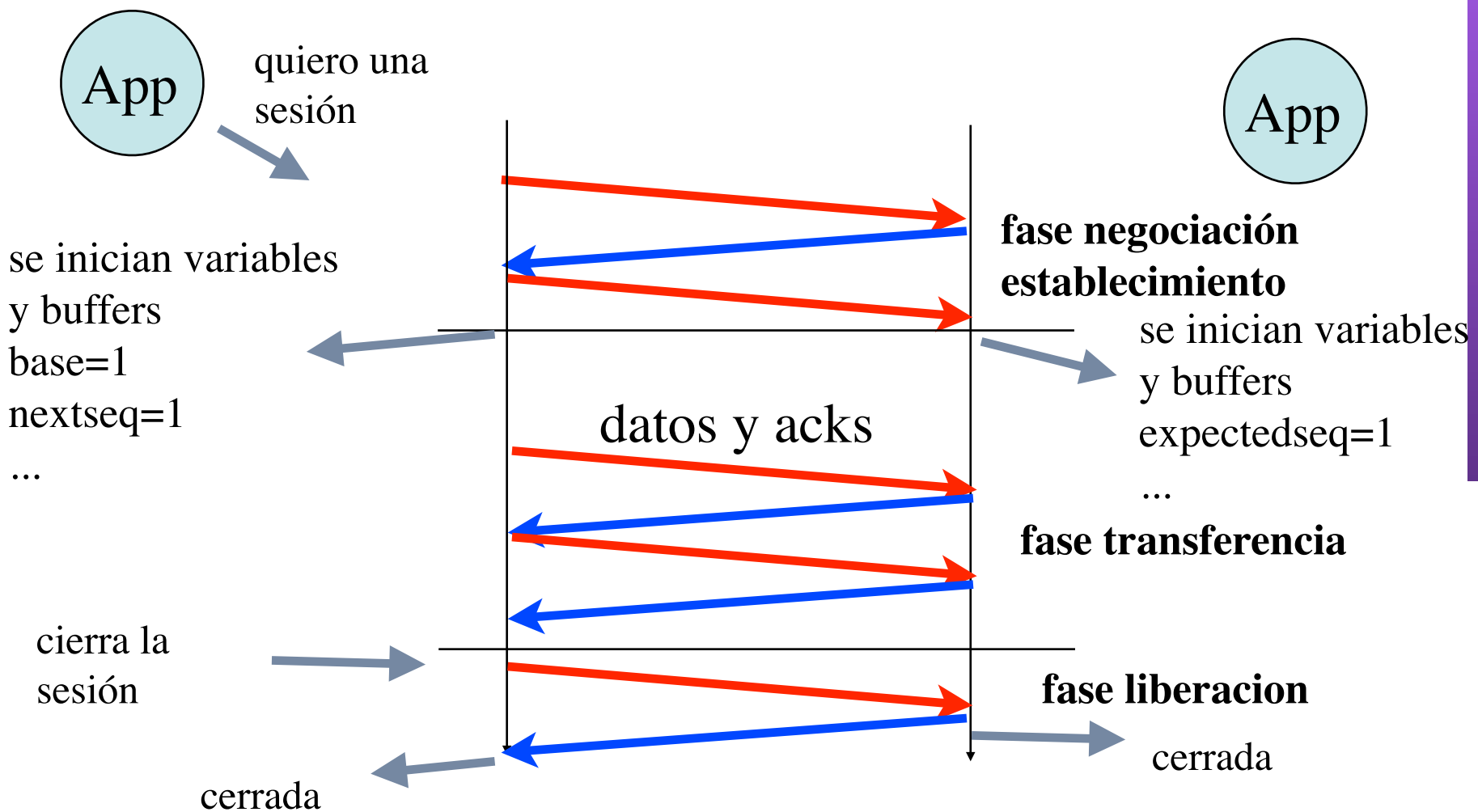
Sesiones/conexiones

- Cada pareja Emisor-Receptor necesita sus propias variables del protocolo de ventana deslizante (base, nextseq, expectedseq, bufferes de paquetes...)
- El nivel de transporte debe gestionar todo esto
- **Sesiones de nivel de transporte (se suelen llamar conexiones)**



Sesiones/conexiones

- Antes de empezar una comunicación hay que avisar para que se inicialicen las variables



- El nivel de transporte incluye también protocolos para el establecimiento y liberación de las sesiones del nivel de transporte

Conclusiones

- Hay mecanismos y protocolos que permiten conseguir un transporte fiable sobre una red no fiable
- Incluso hay mecanismos que permiten conseguir un transporte fiable y razonablemente eficiente, utilizando números de secuencia y ventanas deslizantes
  - Go-back N
  - Selective repeat
- Sus limitaciones de velocidad dependen de la ventana elegida y de su relación con los parámetros del canal (C,RTT)

- **Siguientes clases**
  - Problemas
  - Nivel de red y enrutamiento