1

La capa de Transporte

Principios de transferencia de datos fiable

2

Tema 3: La capa de Transporte.

Transferencia de datos fiable: principios generales.

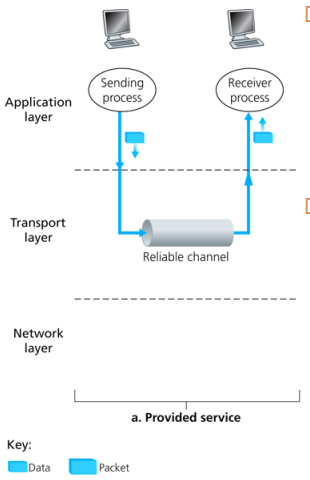
3

- Problema muy relevante en Redes de Computadores.
- En principio aplicable a cualquier capa.
  - En la arquitectura de protocolos TCP/IP:
    - Capas de Aplicación, de Transporte y de Enlace.
- Modelo de servicio.
  - No bits corruptos.
  - No pérdida de bits.

  } Abstracción de un canal fiable
- Por simplificación:
  - Análisis de transferencia de datos UNIDIRECCIONAL.

Transferencia de datos fiable: abstracción.

4



- Protocolo para transferencia fiable de datos en capa "N".
  - Complicado si capa "N–1" brinda servicio NO fiable.
  - Sencillo si capa "N–1" brinda servicio fiable.
- En la figura: abstracción para provisión de un servicio "rdt" (rdt, **r**eliable **d**ata **t**ransfer).
  - P.e: rdt en capa 4 (capa de Transporte).

Transmisión:
• De paquetes de datos y de control
• En sentido contrario ídem

Transferencia de datos fiable: implementación del servicio.

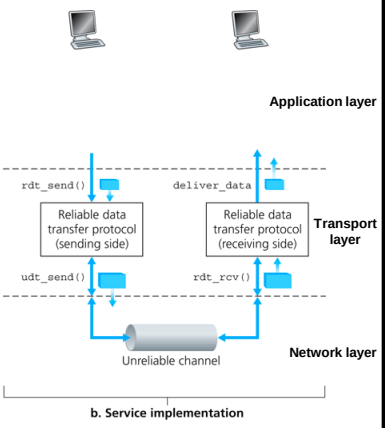
5

EMISOR

- Llamada a "**rdt_send ( )**".
  - Invocación del "rdt emisor" por la capa superior (capa de Aplicación).
  - Pasa "datos" para ser enviados.
- Llamada a "**udt_send ( )**".
  - Invocación del canal (capa inferior, capa de Red) por el "rdt emisor".
  - Pasa "datos" a enviar al canal.

RECEPTOR

- Llamada a "**rdt_rcv ( )**".
  - Invocación del "rdt receptor" por el canal (capa inferior, capa de Red).
  - Toma "datos" recibidos del canal.
- Llamada a "**deliver_data**".
  - Invocación de la capa superior (capa de Aplicación) por el "rdt receptor".
  - Entrega "datos" recibidos.



Transmisión:
- De paquetes de datos y de control
- En sentido contrario ídem

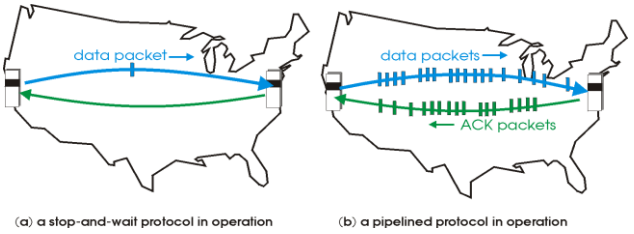
udt: **u**nreliable **d**ata **t**ransfer
rdt: **r**eliable **d**ata **t**ransfer

Universidad de Alcalá

Protocolos fiables con Procesamiento en Cadena.

6

- PROCESAMIENTO EN CADENA (*pipelining*).
  - Procedimiento para mejorar el rendimiento el protocol S&W (*Stop and Wait*).
  - Basado en múltiples paquetes pendientes de reconocimiento.



(a) a stop-and-wait protocol in operation (b) a pipelined protocol in operation

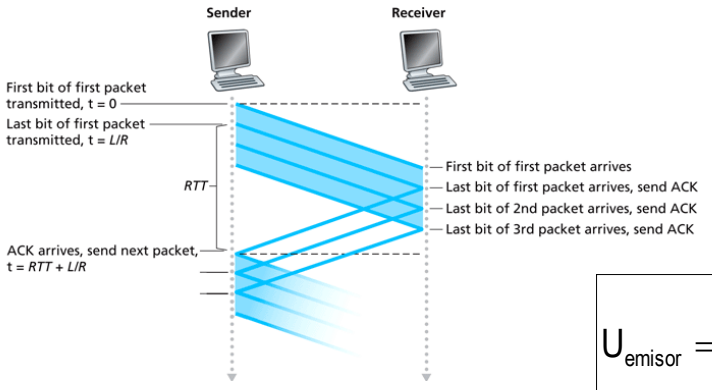
  - Consecuencias:
    - Incremento del rango de números de secuencia.
    - Almacenamiento (*buffering*) en Emisor, y a veces en el Receptor.*
  - Dos estrategias básicas:
    - Retroceder N (*Go-Back-N, GBN*). También denominado "repetición NO selectiva".
    - Repetición Selectiva (*Selective Repeat, SR*).

Universidad de Alcalá

Tema 3: La capa de Transporte

Procesamiento en Cadena: rendimiento.

7



$$U_{emisor} = \frac{k\,L/R}{RTT + L/R}$$

- Tasa de utilización del canal (U_{emisor}).
  En principio la utilización se multiplica por k.
  k: número de paquetes que se envían en cadena.

Procesamiento en Cadena: desafíos.

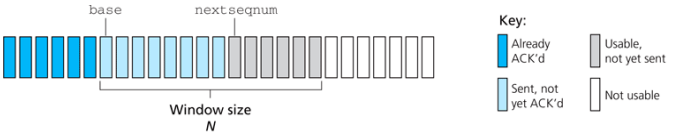
8

- Desafíos.
  - ¿Cuántos paquetes de datos pendientes de ACK se permiten?
  - ¿Cómo se indica la recepción errónea de un paquete?
  - ¿Cómo reacciona el Emisor cuando se corrompe un paquete?

Go-Back-N.

9

- Pueden haber hasta N paquetes consecutivos sin confirmar.*
- Nº de secuencia de k-bits en la cabecera de cada paquete.
- Dentro de los números de secuencia hay que distinguir:
  - **`base`**: número de secuencia de paquete enviado y NO reconocido más antiguo.
  - **`nextseqnum`**: número de secuencia para el siguiente paquete a enviar.
- **`base`** y **`nextseqnum`** permiten identificar cuatro intervalos en el rango de números de secuencia:
  - **`[0,base-1]`**: relativos a paquetes transmitidos y reconocidos.
  - **`[base,nextseqnum-1]`**: relativos a paquetes transmitidos y NO reconocidos.
  - **`[nextseqnum,base+N-1]`**: disponibles, pueden emplearse para enviar paquetes.
  - **`[base+N,2^k-1]`**: NO disponibles para ser utilizados.



Universidad de Alcalá

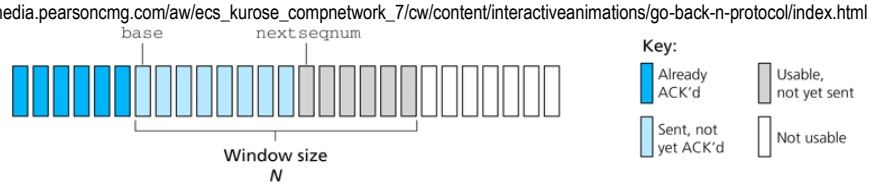
Figure 3.19 ♦ Sender's view of sequence numbers in Go-Back-N

Go-Back-N.

10

- Ventana de tamaño N (ventana de transmisión).
  - Rango de nºs de secuencia para:
    - Paquetes transmitidos pendientes de reconocimiento, y/o
    - Paquetes que pueden transmitirse.
- Cuando el protocolo opera la ventana se "desliza" hacia delante sobre el espacio de nº de secuencia.
  - Protocolo de "ventana deslizante" (*sliding window*).
- Ver applet:

https://media.pearsoncmg.com/aw/ecs_kurose_compnetwork_7/cw/content/interactiveanimations/go-back-n-protocol/index.html



Universidad de Alcalá

Figure 3.19 ♦ Sender's view of sequence numbers in Go-Back-N

Go-Back-N.

11

- Funciones del emisor, qué hace cuando se produce:
  - Recepción de una PDU de la capa superior para ser transmitida.
    - Si hay menos de N paquetes no reconocidos (ventana no llena) ➔ Enviar, nextseqnum + 1
      - Y si no hay paquetes pendientes de confirmar, se inicia el Temporizador.
    - En caso contrario (ventana llena) ➔ Notificación a la capa superior.
  - Fin del Temporizador.
    - GBN emplea temporizador de "cuenta atrás" para el paquete "n" sin confirmar más antiguo, el de inicio de ventana en ese momento.
    - Vence temporizador ➔ Se retransmite el paquete "n" y todos los siguientes de la ventana.
  - Recepción de un ACK (incluye nº de secuencia).
    - ACK con nº de secuencia "n", confirma todos los paquetes hasta "n" incluido ➔ ACK acumulativo.
    - ¿Quedan paquetes pendientes de confirmación?
      - Si ➔ Se reinicia el Temporizador.
      - NO ➔ Se detiene (no se inicia) el Temporizador.
    - Pueden recibirse ACKs duplicados (ver receptor).

Go-Back-N.

12

- Funciones del receptor.
  - Receptor conoce el nº de secuencia del siguiente paquete esperado:
    - **"numsecesperado"**, lo único que tiene que manejar.
  - Si se recibe bien un paquete con "**numsecesperado**":
    - Se entrega a la capa superior.
    - Se envía ACK con ese nº de secuencia.
    - Se incrementa el nº de secuencia esperado ➔ **numsecesperado + 1**
  - En cualquier otro caso:
    - Se descarta el paquete recibido.
    - Se envía ACK con **numsecesperado -1.**
      - Contiene nº de secuencia del último paquete recibido sin error y entregado a la capa superior.
      - ACK duplicado, para que el emisor sepa que tiene que retransmitir a partir de ese nº de secuencia.

Go-Back-N: ejemplo.

13

- Tamaño de ventana.
  - Cuatro (4) paquetes.
- Se pierde paquete nº 2.
  - Paquetes nº 3, 4 y 5 se descartan si no ha llegado antes el paquete nº 2.

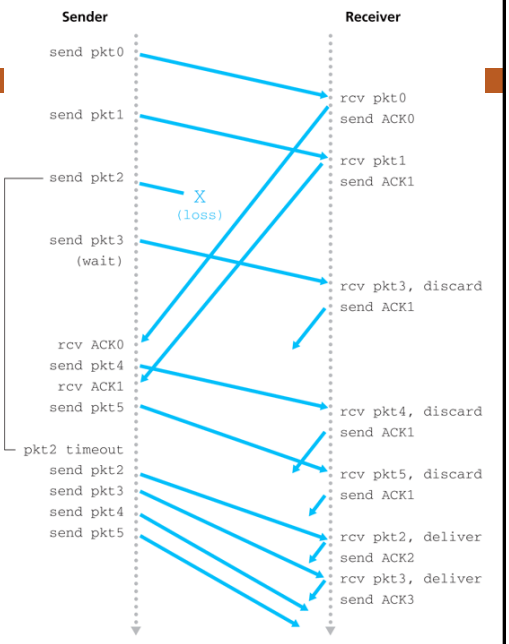


Figure 3.22 ♦ Go-Back-N in operation

Repetición Selectiva (SR, Selective Repeat).

14

"Longitud" en bits del enlace/canal.

- Problema de GBN.
  - Si tamaño de la ventana y producto ancho de banda-retardo son grandes, pueden haber muchos paquetes "viajando" por el canal.
    - Un paquete erróneo ➔ Retransmisión de muchos paquetes.
    - Caos si la probabilidad de error es grande.
    - Problemas de rendimiento.
- Solución: Repetición Selectiva (SR).
  - Se basa en retransmitir SÓLO paquetes erróneos.
  - Emisor sólo reenvía paquetes para los que NO recibe ACK.
  - Receptor confirma individualmente paquetes bien recibidos.
    - Requiere un *buffer* en el Receptor para la entrega en orden a la capa superior.
  - Ver applet

https://media.pearsoncmg.com/aw/ecs_kurose_compnetwork_7/cw/content/interactiveanimations/selective-repeat-protocol/index.html

Universidad de Alcalá
Tema 3: La capa de Transporte

Repetición Selectiva (SR, Selective Repeat).

15

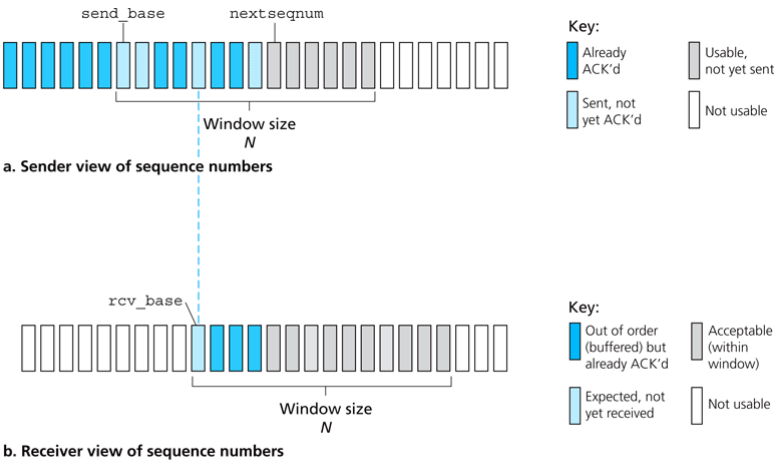


Figure 3.23 ♦ Selective-repeat (SR) sender and receiver views of sequence-number space

Repetición Selectiva (SR, Selective Repeat).

16

- Funciones del emisor, qué hace cuando se produce:
  - Recepción de una PDU de la capa superior para ser transmitida.
    - Si hay nº de secuencia disponible dentro de la ventana del emisor ➔ Se envía y nextseqnum + 1.
      - Se inicia el temporizador para el paquete que se envía.
    - En caso contrario (ventana llena) ➔ Se notifica a la capa superior.
  - Fin de temporizador.
    - Hay un temporizador en el emisor por cada paquete enviado sin confirmar.
    - Vence temporizador asociado a un paquete ➔ Se retransmite el paquete.
  - Recepción de un ACK (incluye nº de secuencia).
    - Si el nº de secuencia está dentro de la ventana ➔ Se marca paquete como confirmado.
    - Si el nº de secuencia es el más bajo (inferior) de la ventana ➔ Avanza la ventana hasta el siguiente nº de secuencia más bajo por confirmar.

Repetición Selectiva (SR, Selective Repeat).

17

- Funciones del receptor, qué hace cuando:
  - Se recibe un paquete bien:
    - Con nº de secuencia en orden (nº de secuencia = `rcv_base`).
      - Se entrega a la capa superior junto con otros paquetes almacenados en orden si los hay, y avanza la ventana en ese mismo número de paquetes.
      - Se envía al emisor ACK selectivo.
    - Con nº de secuencia fuera de orden, pero dentro de la ventana del Receptor, esto es, en el intervalo [`rcv_base`, `rcv_base+N-1`]:
      - Se almacena en el buffer.
      - Se envía al emisor ACK selectivo.
    - Con nº de secuencia en el intervalo [`rcv_base-N`, `rcv_base-1`]*:
      - Se envía ACK al emisor, aunque ya haya sido reconocido anteriormente.
  - En cualquier otro caso se descarta el paquete.

SR: ejemplo.

18

- Tamaño de ventana en emisor y receptor.
  - Cuatro (4) paquetes.
- Se pierde paquete nº 2.

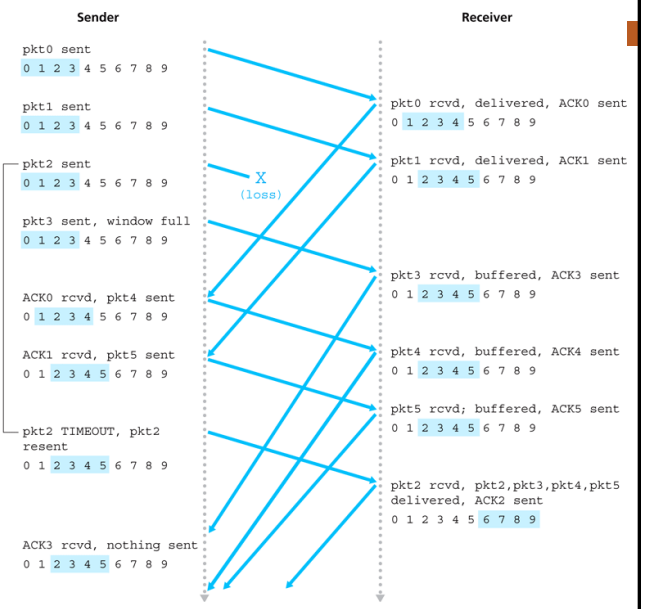


Figure 3.26 ♦ SR operation

SR: problemática.

19

- Ejemplo:
  - Nºs secuencia: 0, 1, 2, 3.
  - Tamaño de ventana: 3
- Desde la perspectiva del Receptor:
  - En ambos casos el Rx ve lo mismo, pero:
    - Caso a), emisor re-envía el 1er paquete nº 0.
    - Caso b), emisor envía el 2º paquete nº 0.
  - Problema:
    - En el caso a) se entrega información duplicada como nueva.
    - Causa de esta mala interpretación:
      - Tamaño de ventana inapropiado.

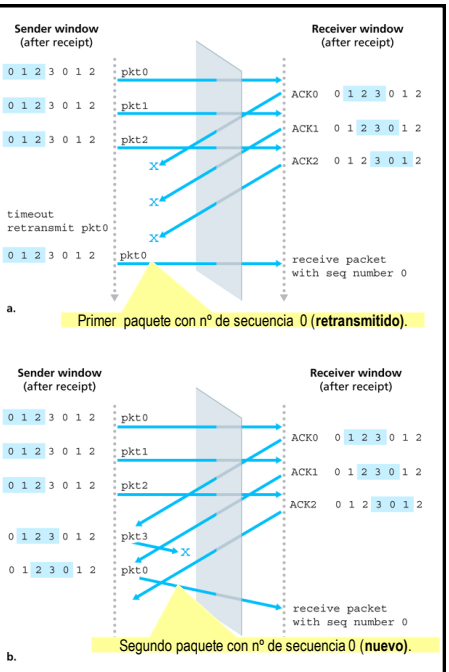


Figure 3.27 ♦ SR receiver dilemma with too-large windows: A new packet or a retransmission?

SR: tamaño de ventana.

20

- ¿Qué relación debe existir entre el tamaño de la ventana y el tamaño del espacio de nº de secuencia?

$$N \leq N_s/2 = 2^k/2$$

- Donde:
  - N: tamaño de ventana.
  - N_s: tamaño del espacio de nºs de secuencia.
  - K: número de bits utilizados para representar los nºs secuencia.

El tamaño de ventana (N) también depende de:
• Capacidad del receptor para recibir y almacenar datos en su *buffer* de recepción (**control de flujo**), y/o
• Nivel de congestión de la red (**control de congestión**).

Protocolos con procesamiento en "cadena": variantes.

21

Retroceder-N (GBN)

- Emisor
  - Admite hasta N paquetes sin confirmar.
  - Temporizador para el paquete sin confirmar más antiguo.
    - Si expira el temporizador se retransmiten todos los paquetes sin confirmar.
- Receptor.
  - Envía ACKs acumulativos.
  - No asienten paquetes fuera de orden.

Repetición selectiva (SR)

- Emisor
  - Admite hasta N paquetes sin confirmar
  - Temporizador para cada paquete sin confirmar.
    - Si expira un temporizador se retransmite sólo el paquete afectado.
- Receptor
  - Envía ACKs no acumulativos.
  - ACKs por cada paquete.
  - Sí asienten paquetes fuera de orden.

Transferencia fiable de datos: otro factor a tener en cuenta.

23

- Todo el análisis previo ha supuesto que los paquetes viajan por el canal sin "desordenarse".
  - Supuesto razonable si el canal es un cable.
- Si el canal es una red del tipo DATAGRAMAS los paquetes pueden "desordenarse" (caso de las redes IP).
  - El canal puede verse como un *buffer* que almacena paquetes y puede enviarlos en cualquier momento posterior.
  - Dado que los nºs de secuencia se reutilizan ➔ Pueden llegar paquetes duplicados en el receptor.
    - Solución: emisor no debe reutilizar un nº de secuencia hasta que esté "seguro" que el paquete antes enviado con ese mismo nº ya no está en la red.
      - Supone cierto valor máximo de tiempo de vida de un paquete en la red.