

Brevísima presentación sobre protocolos

Marzo - 2005

Qué es un protocolo (i)

- Son cierto tipo de acuerdo sobre el intercambio de información en el sistema
Se vuelve una norma a seguir para integrar entidades activas en la aplicación
 - usuarios,
 - objetos activos,
 - sistemas, ...
- Los protocolos son una especie de algoritmos distribuidos.
 - Un protocolo implementa una función de comunicación donde se define un servicio.

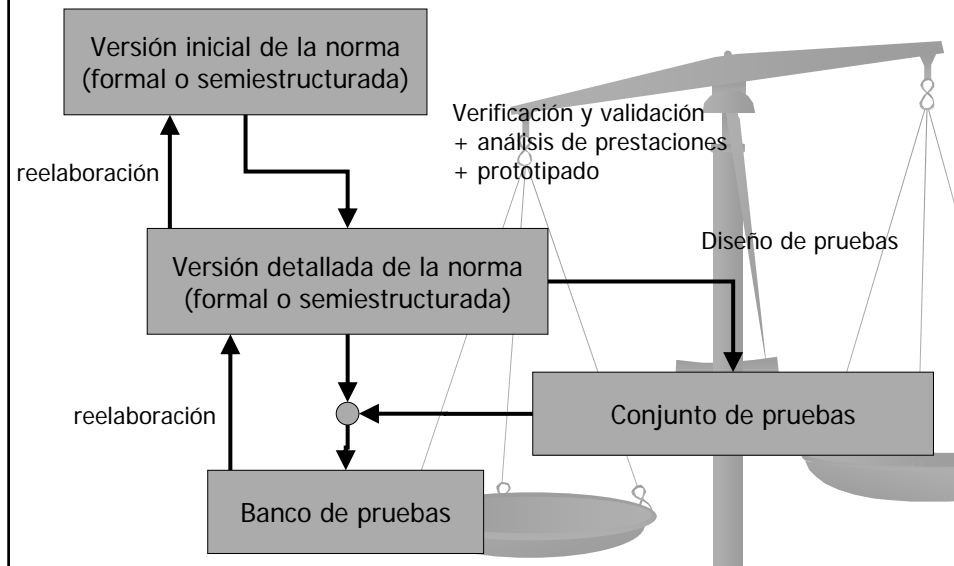
El protocolo como un lenguaje

- La definición de un protocolo se asemeja mucho a la definición de un lenguaje:
 - Se define un formato preciso para los mensajes válidos (SINTAXIS)
 - Se definen reglas de procedimiento para el intercambio de datos (pasos a seguir en el intercambio de mensajes) (ALGORITMO/COORDINACION)
 - Definen un vocabulario de mensajes válidos junto a su significado (SEMANTICA)

Elementos de un protocolo

- La especificación completa de un protocolo contiene:
 - El *servicio* proporcionado por el protocolo.
 - Los *supuestos* sobre el entorno en el que se ejecuta el protocolo.
 - El *vocabulario* de los mensajes empleados en la implementación del protocolo.
 - El *formato* (codificación) de cada mensaje del vocabulario.
 - Las *reglas de procedimiento* que mantienen la consistencia de los intercambios de mensajes.

Ciclo de vida de un protocolo



Diseñando un protocolo

- En primer lugar hay que enumerar el vocabulario de primitivas.
 - Ej.: {conecta, envía, error, ack, desconecta}
- En segundo lugar, el formato de cada primitiva.
 - Ej.
 - conecta(identidad)
 - envía(destinatario, mensaje)
 - error(mensaje), ...

Diseñando un protocolo

- En tercer lugar, las reglas por las que se rigen las secuencias de mensajes.
 - Posibilidades:
 - Hacerlo informalmente en texto.
 - Hacerlo de modo formal mediante:
 - Diagramas de secuencia de tiempo.
 - Diagramas *tipo de proceso* en SDL (System Design Language)
 - Diagramas de estado y/o actividad en UML 2.0
 - Lenguajes especiales (Promela , ...) ...

Reglas del procedimiento

- Las reglas del procedimiento dicen qué secuencias de mensajes son admisibles en el protocolo.
 - Suelen expresarse como autómatas: guiados por eventos (que disparan transiciones)
 - Los eventos se almacenan en colas de entrada a las acciones.
- En SDL se utiliza algo parecido a diagramas de flujo con estados.

Ejemplo, el protocolo de Lynch

Un ejemplo no muy
lúcido ☺

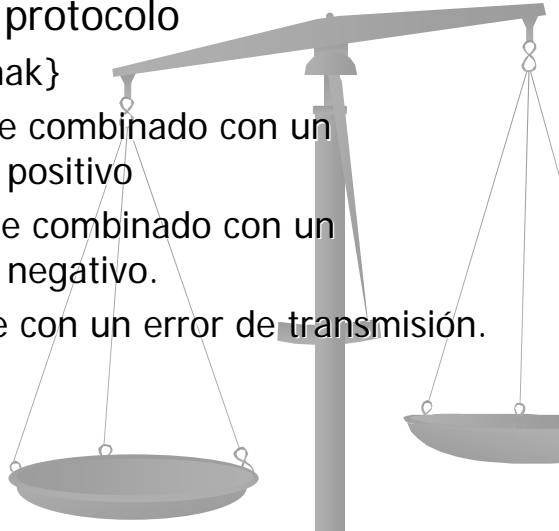
- Especificación del servicio:
 - Transferir archivos como secuencia de caracteres por la línea telefónica evitando errores de transmisión, suponiendo que pueden detectarse todos los errores.
 - Es una transferencia de archivos full-duplex.
 - Se envían reconocimientos positivos y negativos para el tráfico de A a B mediante la línea de B a A (y viceversa).
 - Cada mensaje tiene dos partes, una de mensaje, y otra de control que se aplica al tráfico en el canal contrario.

Ejemplo, el protocolo de Lynch

- Suposiciones sobre el entorno
 - El entorno consta de dos usuarios del servicio y un canal de transmisión.
 - Cada usuario pide un archivo y espera la vuelta.
 - Se supone que el canal distorsiona arbitrariamente el mensaje, pero no pierde, inserta, duplica, ni reordena los mensajes.
 - Se parte de la existencia de un módulo de nivel inferior que atrapa las distorsiones y reparte mensajes no distorsionados de tipo "err".

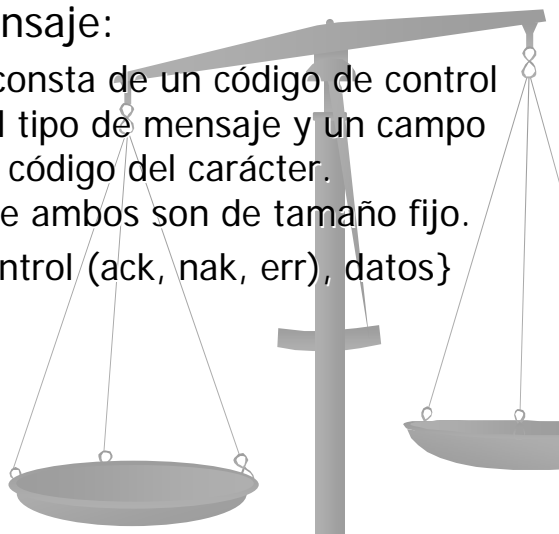
Ejemplo, el protocolo de Lynch

- Vocabulario del protocolo
 - $V = \{\text{ack}, \text{err}, \text{nak}\}$
 - ack: un mensaje combinado con un reconocimiento positivo
 - nak: un mensaje combinado con un reconocimiento negativo.
 - err: un mensaje con un error de transmisión.



Ejemplo, el protocolo de Lynch

- Formato del mensaje:
 - Cada mensaje consta de un código de control que identifica el tipo de mensaje y un campo de datos con el código del carácter.
(suponemos que ambos son de tamaño fijo.)
 - {etiqueta de control (ack, nak, err), datos}

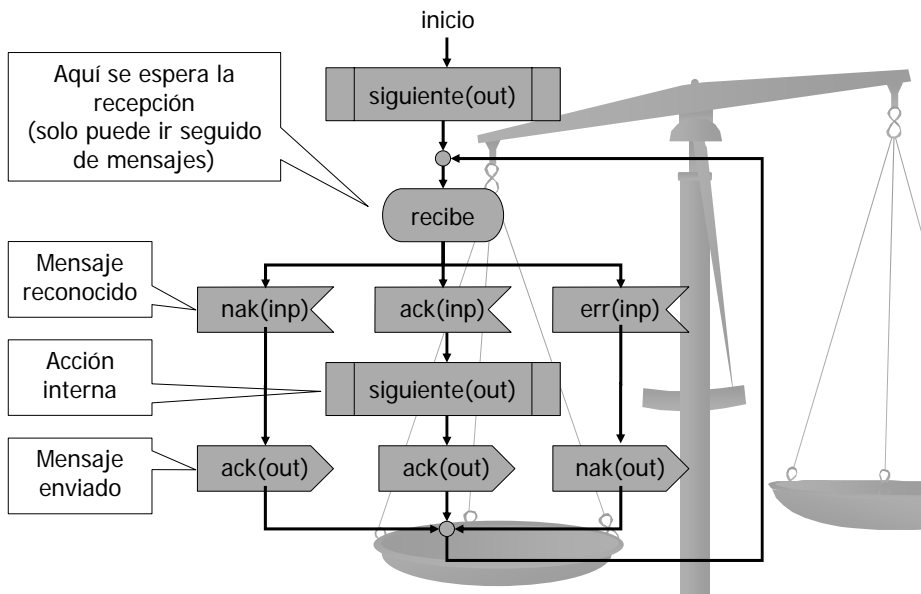


Ejemplo, el protocolo de Lynch

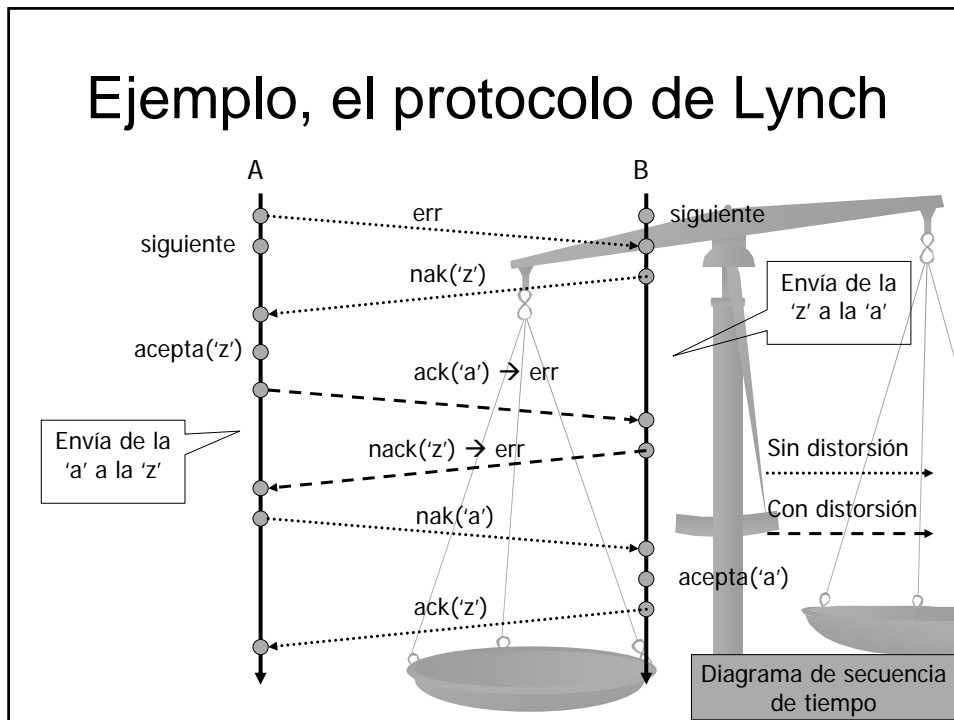
- Reglas de procedimiento:

1. Si la recepción anterior no tenía errores, el próximo mensaje en el canal contrario llevará un reconocimiento positivo; si la recepción tuvo errores, llevará un reconocimiento negativo.
2. Si la recepción previa llevaba un reconocimiento negativo, o la recepción anterior fue errónea, se retransmite el mensaje anterior; de otro modo, se consigue otro mensaje para una nueva transmisión.

Ejemplo, el protocolo de Lynch



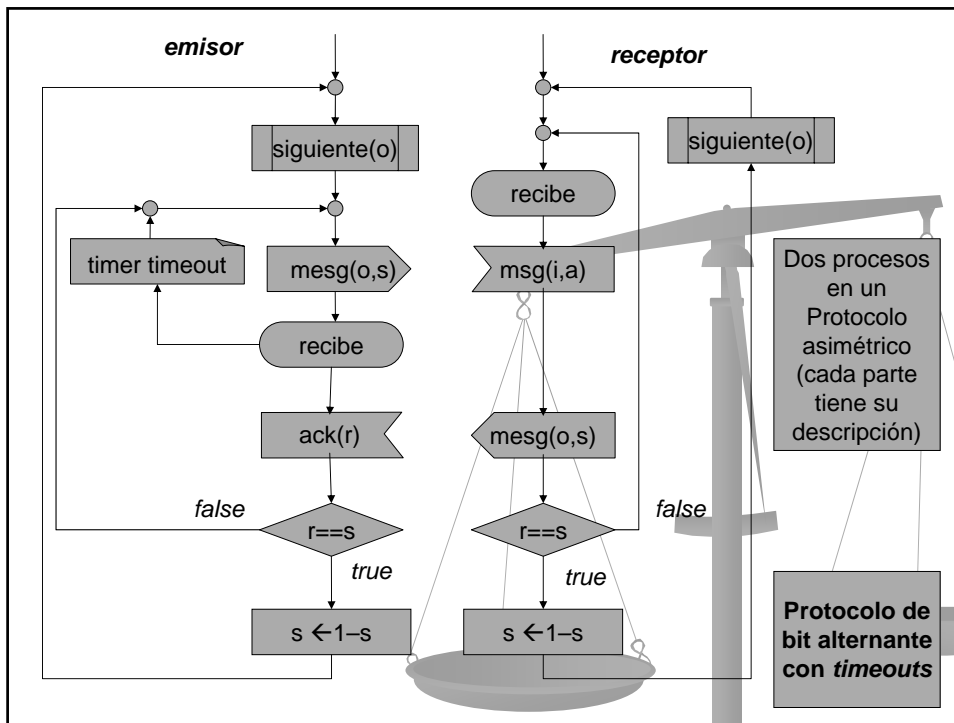
Ejemplo, el protocolo de Lynch



Ejemplo, el protocolo de Lynch

■ Carencias del diseño

- El envío/recepción debe ocurrir simultáneamente.
- El protocolo debe comenzar en puntos diferentes en cada uno de los dos procesos, para que estén "en fase".
 - Puede comenzarse con un mensaje "err"
- Se ha omitido: el receptor debe ser capaz de decidir si un dato recibido correctamente (almacenado temporalmente en "inp"), ha de ser almacenado.
- El protocolo, en resumen, contiene escenarios erróneos.



Propiedades de un buen protocolo

- Simplicidad: El caso de los protocolos de peso ligero.
- Modularidad: Una jerarquía de funciones.
- Protocolos bien formados (aunque no sobreespecificados) (completos y sólidos)
- Robustez
- Consistencia: Posibles fallos son:
 - Interbloqueos.
 - Bloqueos activos.
 - Terminaciones inadecuadas.