



Tema 3

Comunicación entre procesos

Carlos Montellano



Contenido

- 1. Protocolos de Transporte**
- 2. Sockets y Puertos**
- 3. Comunicación síncrona y asíncrona**
- 4. Llamada a Procedimientos Remotos**

A faint, stylized map of South America is visible in the background of the slide, showing major geographical features and country borders.

Dos Características Clave en SD.

1. Ausencia de memoria compartida
 2. Existencia de múltiples elementos que pueden fallar independientemente.
- Esto implica que la comunicación se basa en **transferencia de mensajes** y la existencia de **mecanismos de comunicación** entre procesos.

A background map of South America, showing the continent's outline and major cities. The map is in a light blue and white color scheme, with a darker blue overlay at the top where the title is located.

Protocolos de Transporte

- Puertos: IP (comunicación de host a host), los protocolos a nivel de transporte permiten la comunicación de proceso a proceso.
- UDP:
 - Cada datagrama se encapsula en un paquete IP.
 - No ofrece garantía de entrega.
- TCP:
 - Proporciona un servicio de transporte más sofisticado.
 - Entrega fiable de secuencias de bytes: se reciben todos los datos y en el mismo orden.
 - Antes de transferir cualquier dato, el proceso emisor y receptor deben cooperar para establecer el canal de comunicación bidireccional.
 - Control de Flujo: el emisor se cuida de no saturar al receptor o a los nodos intermedios.

Middleware layers

Applications, services

Remote invocation, indirect communication

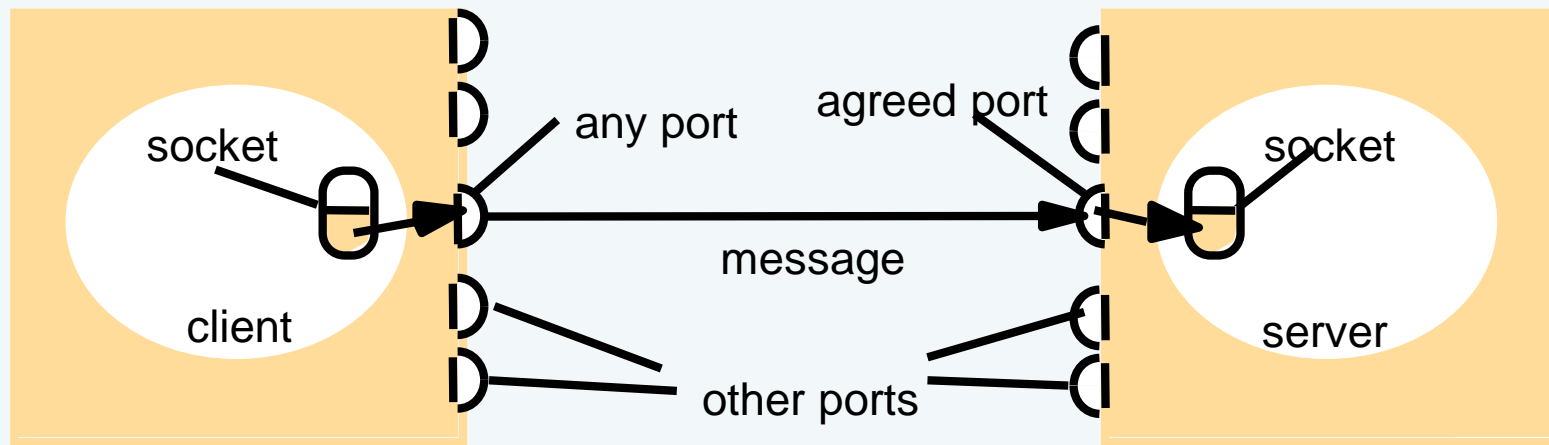
Underlying interprocess communication primitives:
Sockets, message passing, multicast support, overlay networks

UDP and TCP

Middleware
layers

This
chapter

Sockets y puertos



Internet address = 138.37.94.248

Internet address = 138.37.88.249

A faint, stylized map of Europe serves as the background for the top section of the slide. The map is rendered in a light blue tone, showing major landmasses and some geographical features like rivers and coastlines. It is partially obscured by the title text and the blue header bar.

Características de la Comunicación entre Procesos

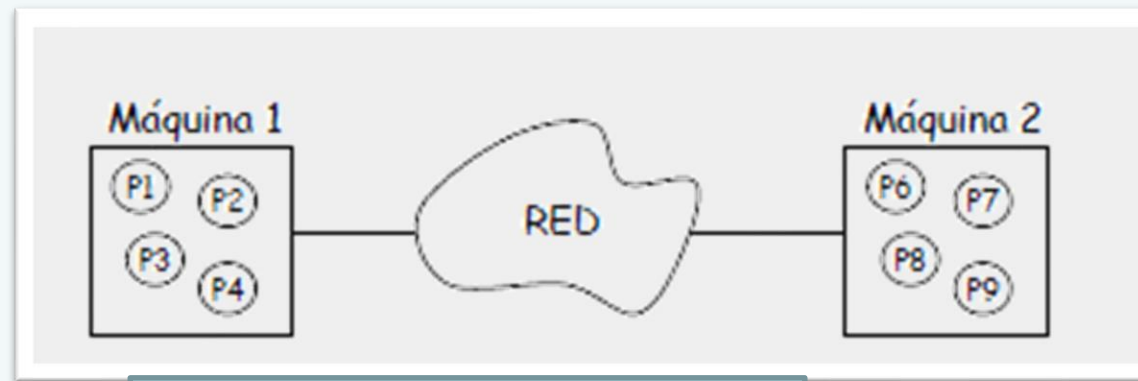
❖ Comunicación

- En grupo
- Punto a Punto

❖ Se tienen primitivas básicas

- `Send(destination, this_msg, msg_length)`
- `Receive(source, a_msg, &how_long)`

Los Modelos de Comunicación



Los procesos
están separados

Lógicamente

Físicamente

Se requiere una línea de comunicaciones
Para

Transferir Datos

Sincronizarse

¿Como se
comunican un grupo
de procesos?

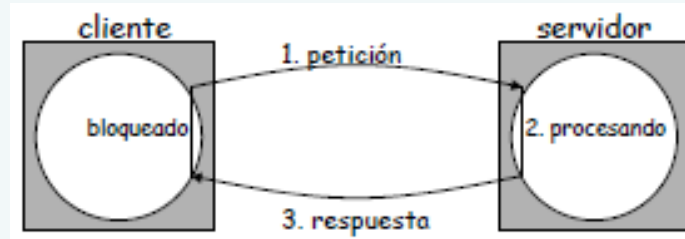
Una Pareja

Modelo Cliente-Servidor

A Un Grupo

Modelo Multicast

Modelo Cliente - Servidor



1. Cliente: Envió \rightarrow bloqueado
2. Servidor: Recibe \rightarrow procesa \rightarrow contesta
3. Cliente : Recibe respuesta \rightarrow continua

RPC

Imprimir(Fichero, Impresora, ok)

Averiguar_Nodo_Impresor
Enviar (Fichero,Nodo_Impresor)
Recibir(Respuesta,Nodo_Impresor=

¿Como se conocen los
clientes y los servidores?

Los servidores son dinámicos

Los servidores deben registrarse con un
nombre de servicio predefinido

Los servidores no conocen a los clientes

Los clientes deben incluir su id. en la petición

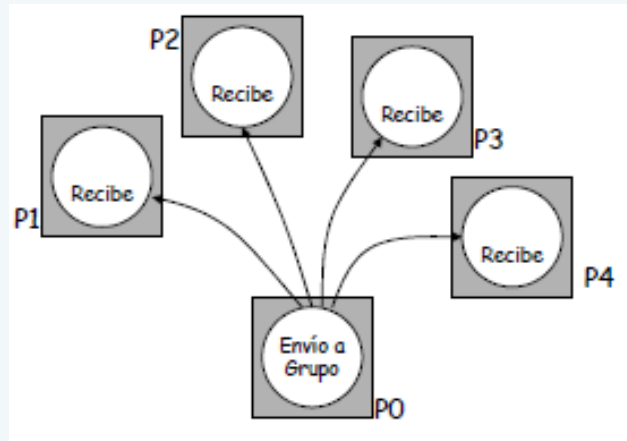
Clientes y
Servidores

No son maquinas
SON PROCESOS
ambivalentes

Modelo Multicast

MULTICAST

Envío de múltiples copias de un mismo mensaje desde un proceso origen a múltiples procesos de destino



Aplicaciones de Multicast

Búsqueda de un recurso
Tolerancia a fallos
Actualizaciones múltiples

La velocidad del envío multicast depende del algoritmo utilizado y del soporte hardware disponible

Rendimiento
General del Sistema

A faint, stylized map of Mexico serves as the background for the top portion of the slide. It shows major geographical features like the Gulf of Mexico, the Yucatan Peninsula, and various state boundaries. The map is rendered in a light blue and white color scheme, blending into the dark blue header bar.

Comunicación Síncrona y Asíncrona

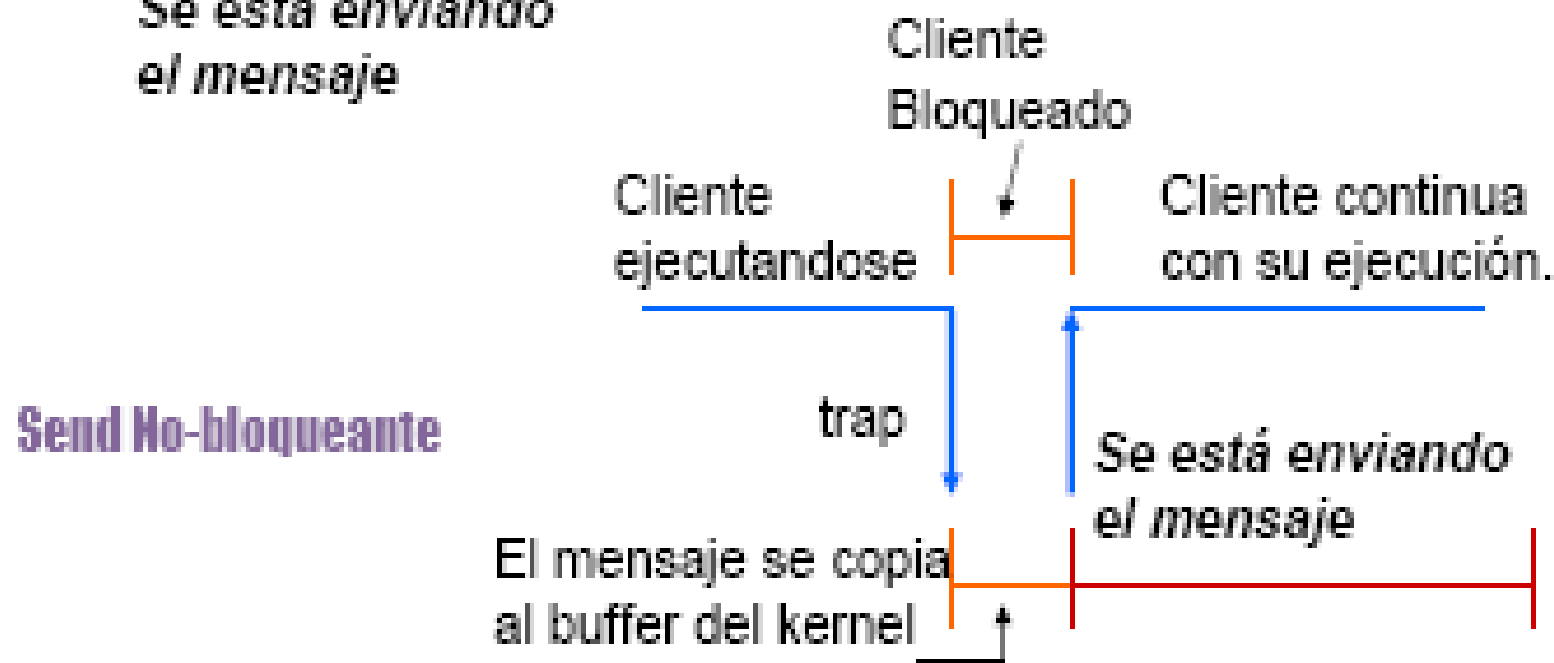
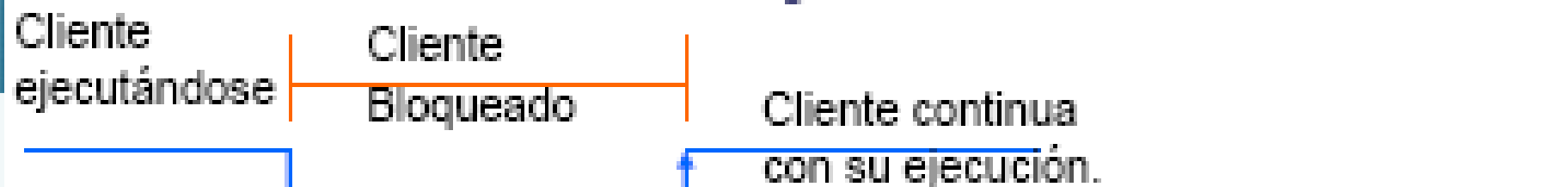
❖ Síncronas:

- El emisor (send) se suspende hasta tanto el mensaje no es recibido.
- Una llamada a receive no retorna hasta que el mensaje no ha sido colocado en el buffer que especifica el receptor.
- En algunos sistemas el receptor puede especificar de qué emisor desea recibir, en cuyo caso permanecerá bloqueado hasta que lleguen los mensajes que le interesan.

A background image showing a map of the Americas, with North and South America visible. The map is in a light blue and white color scheme, with a darker blue overlay on the right side.

❖ Asíncronas:

- Un send no bloqueante retorna el control al llamador inmediatamente antes que el mensaje sea enviado.
- Un receive no bloqueante le indica al kernel el buffer donde se dejará el mensaje y la llamada retorna inmediatamente.



A background image showing a map of the Americas, with North and South America visible. The map is in a light blue and white color scheme, with a darker blue overlay on the right side.

❖ Desventajas del send asíncrono:

- El emisor puede llegar a no tener idea de cuando efectivamente se ha realizado el envío.

❖ Solución: se interrumpe al emisor cuando el mensaje ha sido enviado (los programas son más difíciles de depurar y se vuelven complejos)



Tipos de Servicio

❖ Datagramas:

- Cada mensaje se envía en forma individual; algunos mensajes pueden perderse, otros se duplican y pueden llegar fuera de orden. Las aplicaciones que utilizan UDP dependen de sus propios “chequeos” para conseguir la calidad que necesitan respecto a la fiabilidad.

❖ Streams:

- Se establece una conexión lógica entre el emisor y el receptor. Se garantiza que los mensajes lleguen en secuencia, sin pérdidas o duplicados; tal y como fluyen las palabras en una conversación telefónica.

A background image showing a map of the Americas, with North and South America visible. The map is in shades of blue and green, with some white lines indicating borders or geographical features. It serves as a decorative header for the slide.

❖ Sockets:

- Los conectores deben estar asociados a un puerto local y a una dirección Internet .
- Los procesos pueden usar el mismo conector para leer y escribir mensajes.
- Cada computador permite 2¹⁶ puertos
- Cada proceso puede utilizar varios puertos para recibir mensajes, pero un proceso no puede compartir puertos con otros procesos del mismo computador.
- Cada conector se asocia con un protocolo concreto que puede ser TCP o UDP.