



Sistemas Distribuidos I (75.74)

Multicomputing y Comunicaciones

Multiprocessors, Multicomputing, TCP/UDP

Docentes

- Pablo D. Roca
- Ezequiel Torres Feyuk
- Guido Albarello
- Ana Czarnitzki
- Cristian Raña

Agenda



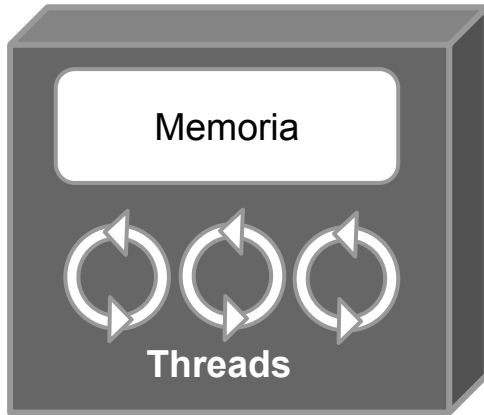
- **Multi-Processors & Multi-Computing**

- Comunicaciones

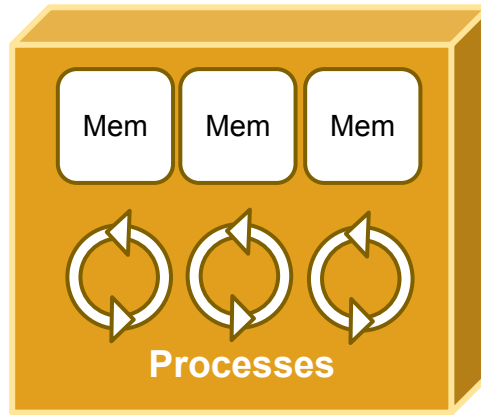


Modelos de Multiprogramación

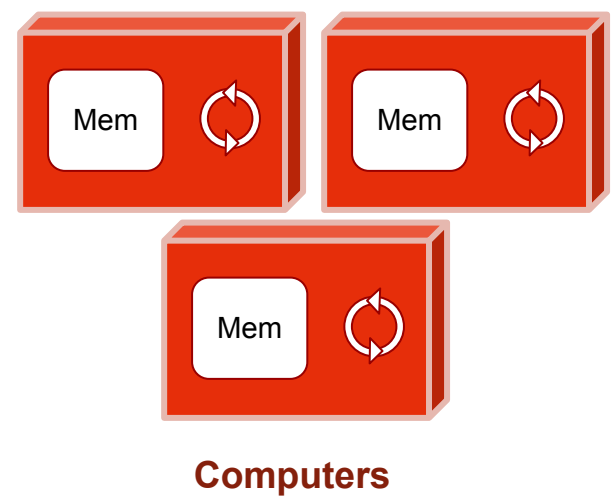
Multi-threading



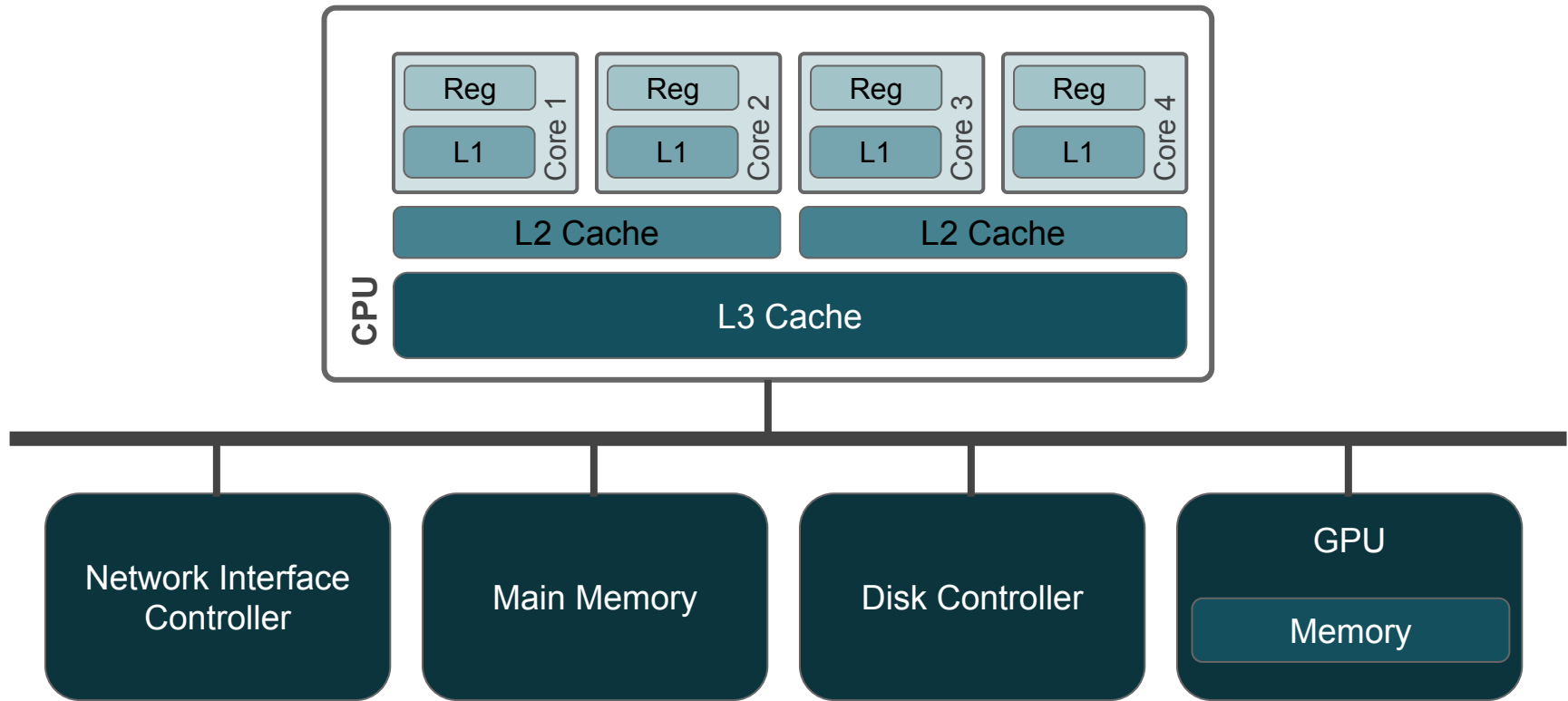
Multi-processing



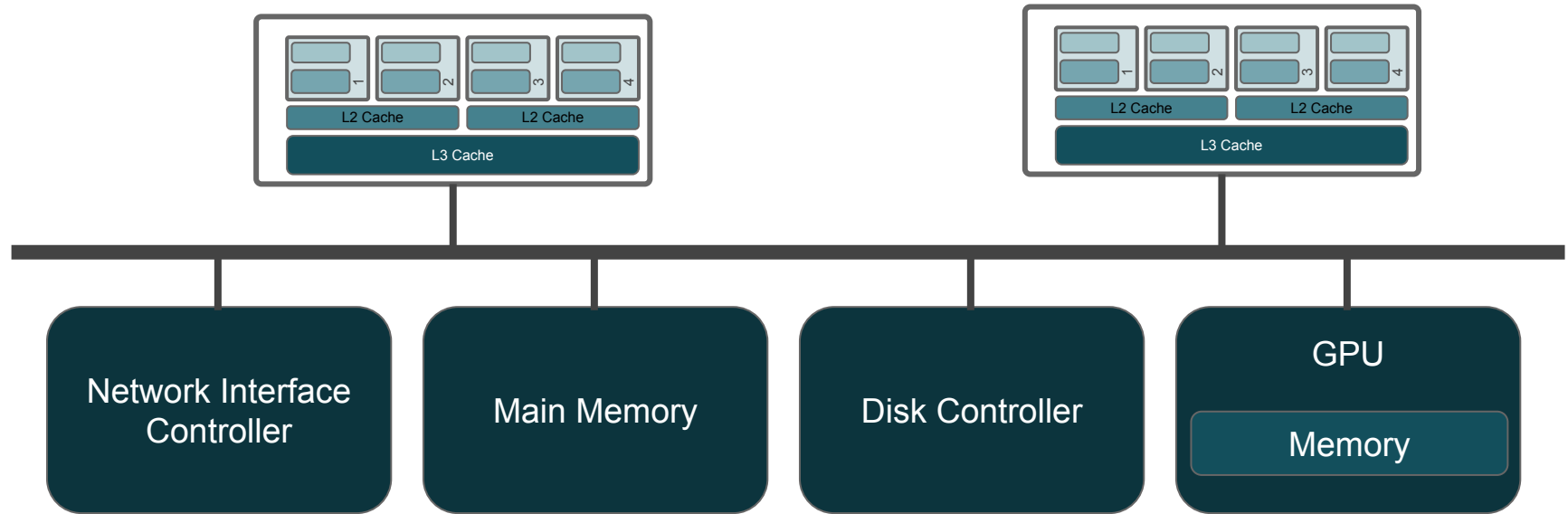
Multi-computing



Modelo Físico | Una CPU



Modelo Físico | Varias CPUs





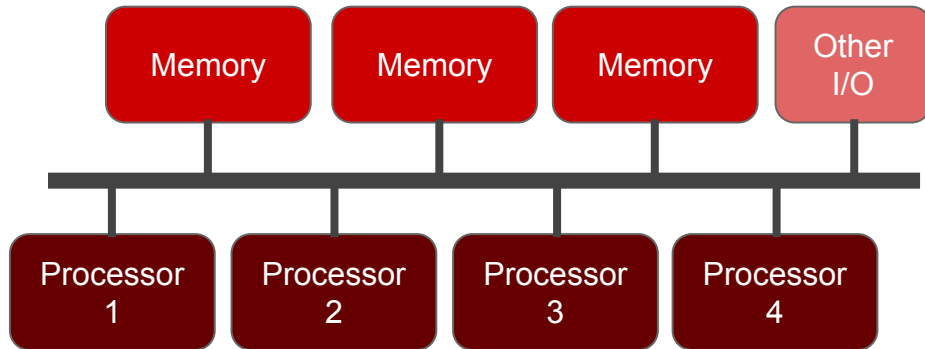
Clasificación de sistemas de acuerdo a cardinalidad de flujos de instrucciones (procesadores) y flujos de datos (memoria).

- **SISD** (*Single Instruction Single Data*): modelo estándar de un procesador sin paralelismo
- **SIMD**: *array processors*. Ej: GPU
- **MISD**: no son usuales (procesar múltiples veces los mismos datos de forma determinística retorna el mismo resultado)
- **MIMD**: se divide en dos modelos:
 - *Multiprocessors* (?): con memoria y/o *clock* compartidos
 - *Multicomputers*: sin memoria ni *clock* compartidos

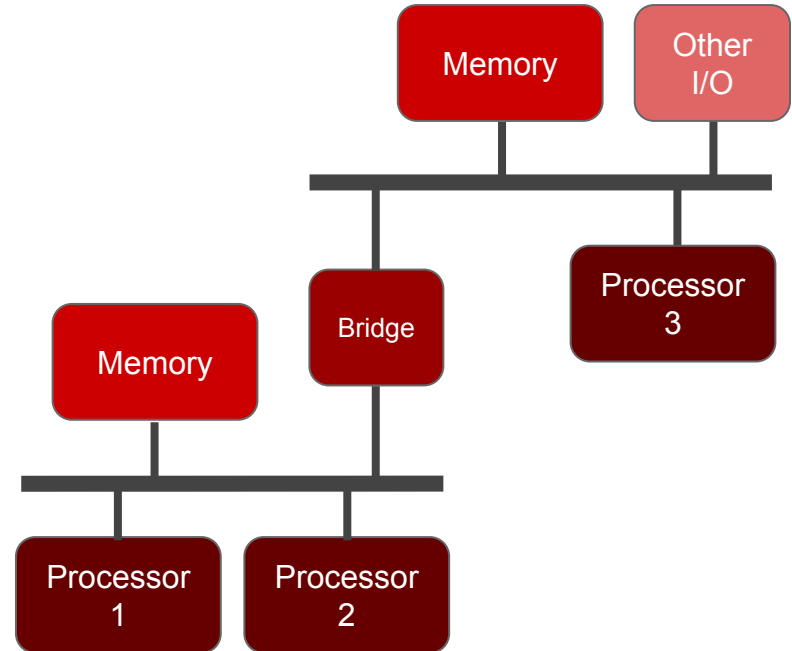
MIMD | *Multiprocessors* (Memoria Compartida)



Symmetric Multiprocessing



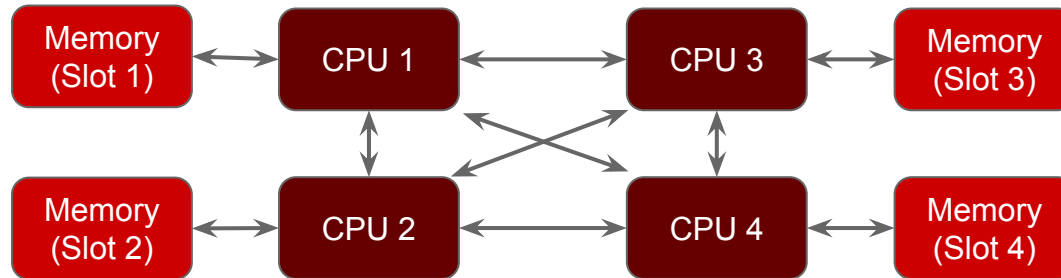
Asymmetric Multiprocessing





MIMD | Multiprocessors | UMA vs NUMA

- Uniform Memory Access (**UMA / non-NUMA**)
 - Tiempo de acceso a la memoria es idéntico para todos los procesadores.
- Non Uniform Memory Access (**NUMA**):
 - Ideado en SGI, ahora presente en Linux kernel y MS Servers
 - Cada CPU controla un bloque de memoria y se transforma en su 'home agent'





MIMD | *Multicomputers* (sin memoria compartida)

- Cada computadora tiene su propia memoria local
- Cada computadora puede fallar de forma independiente
- No poseen un reloj central de ejecución de instrucciones
- Requieren comunicación entre computadoras:
 - Networking: LAN, MAN, WAN



Recursos Compartidos

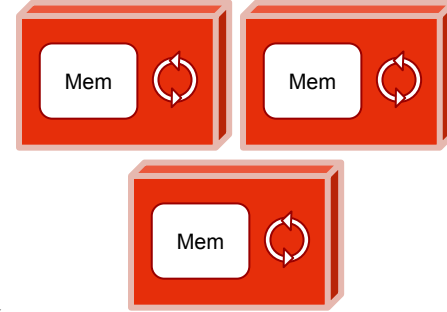
- Ninguno

Sincronización

- Mensajes Ad-Hoc entre Computadoras
=> necesidad de implementar mecanismos de sincronización.

Características clave

- Comunicación de red => problemas por limitaciones de ancho de banda, latencia y pérdida de mensajes.
- Comunicación entre procesos: compleja y central al diseño del sistema.
- Alta escalabilidad y tolerantes a fallos.



Agenda



○ Multi-Processors & Multi-Computing

● **Comunicaciones**

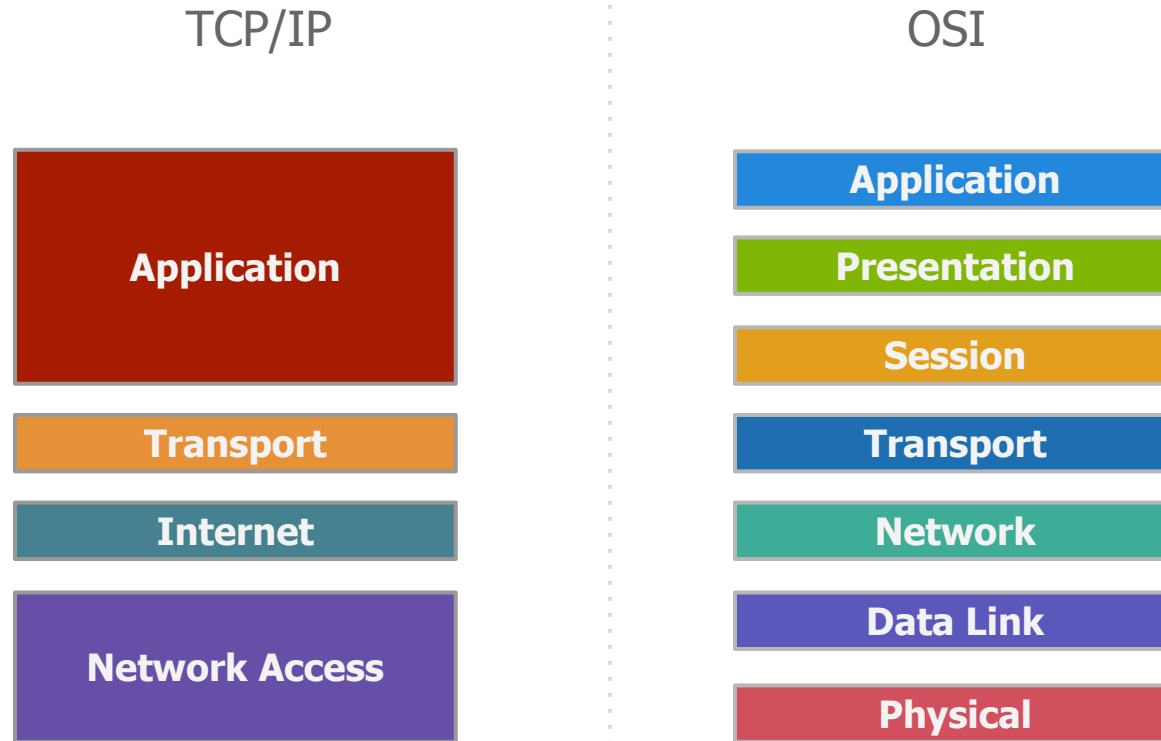


Application	Aplicaciones de usuario, representación de datos
Transport	Comunicación punto a punto
Internet	Lógica de transmisión de datos sobre la red
Network Access	Transferencia física confiable, libre de errores.



Application	Aplicaciones de usuario
Presentation	Representación de datos
Session	Manejo de conexiones y sesión
Transport	Transferencia confiable, libre de errores
Network	Establecer, mantener y terminar conexiones. Transmisión
Data Link	Sincronización, control de errores y envío de frames
Physical	Manejo del medio físico para transmitir bits

Comunicaciones | Relaciones entre OSI y TCP/IP



Estructura de Paquetes | IP



Ver.	IHL	Type of Service	Total Length	
Identification			Flags	Frag. Offset
TTL	Protocol		Header Checksum	
Source Address				
Dest. Address				
Options				
Data (variable)				

32 bits



Estructura de Paquetes | TCP y UDP

TCP

- Orientado a conexión
- Asegura entrega y orden

Source Port		Dest. Port	
Sequence Number			
Ack. Number			
Data Offset	Reserved	Flags	Window (sliding win.)
Checksum		Urgent Pointer	
Options			Padding
Data (variable)			

32 bits

UDP

- Orientado a datos
- Sin garantías: *best effort*

Source Port	Dest. Port
Length	Checksum
Data (variable)	

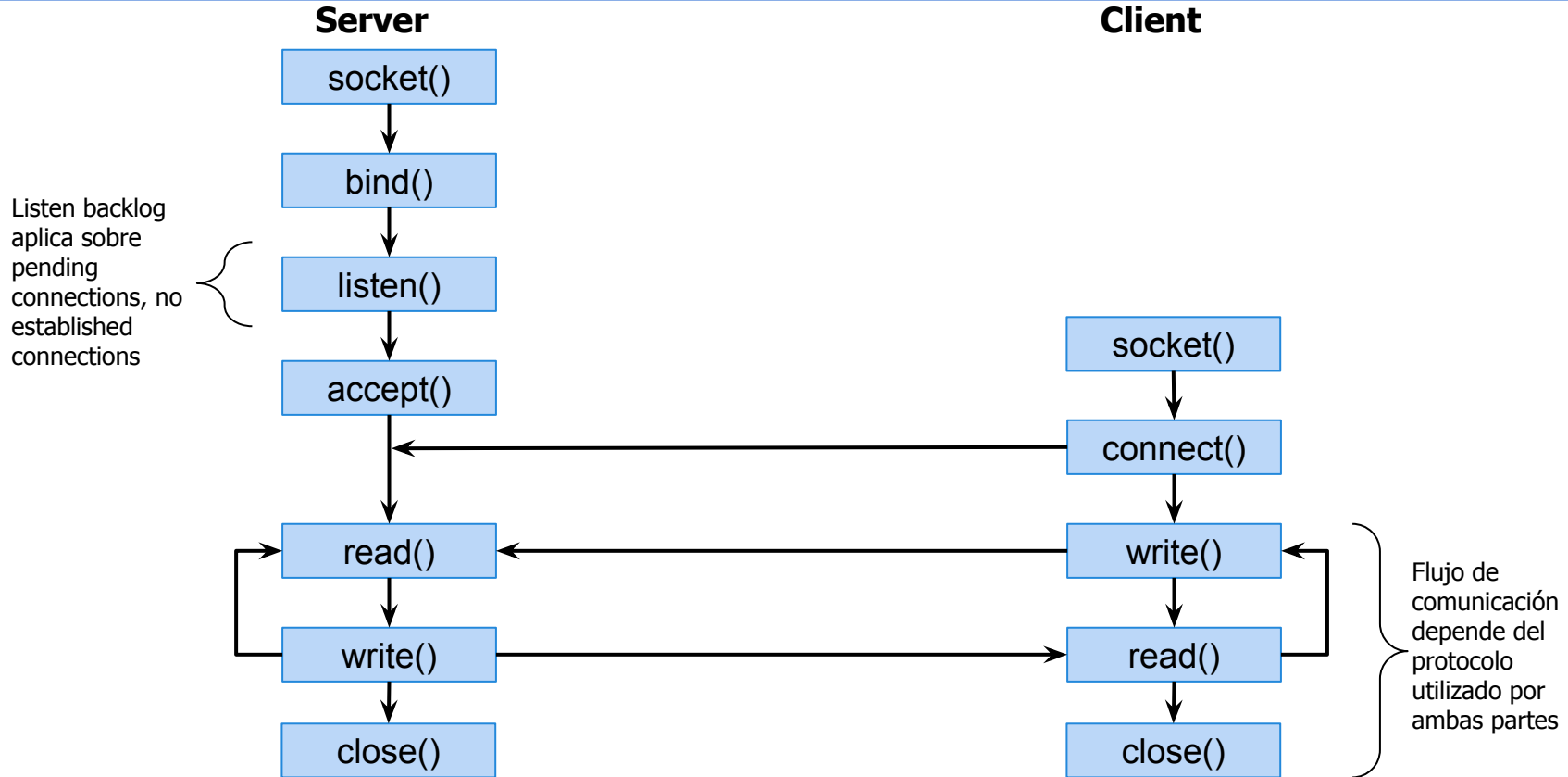
32 bits



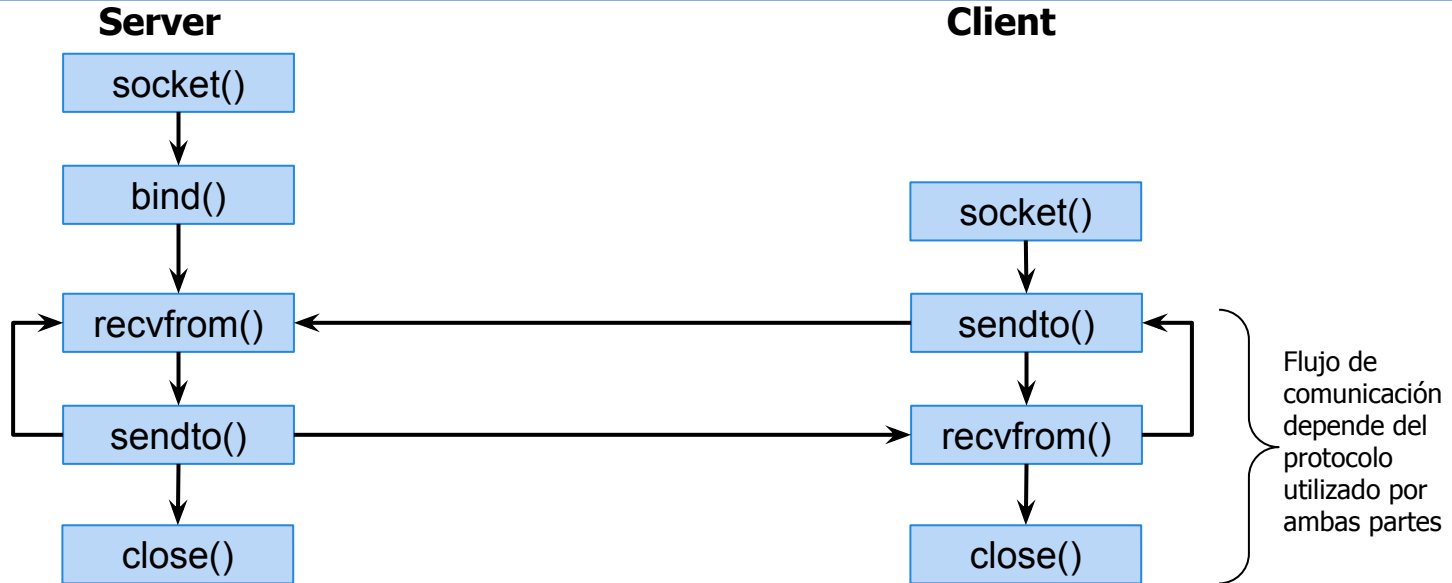
Sockets | Primitivas

- **socket**(domain: **int**, type: **int**, protocol: **int**): **int**
- **bind**(fd: **int**, addr: struct sockaddr*, addrlen: **int**): **int**
- **listen**(fd: **int**, backlog: **int**): **int**
- **accept**(fd: **int**, addr: struct sockaddr*, addrlen: **int**): **int**
- **connect**(fd: **int**, addr: struct sockaddr*, addrlen: **int**): **int**
- **send**(fd: **int**, buf: **void***, buflen: **int**, flags: **int**): **int**
- **receive**(fd: **int**, buf: **void***, buflen: **int**, flags: **int**): **int**
- **close**(fd: **int**): **int**

Sockets | Flujo de uso TCP



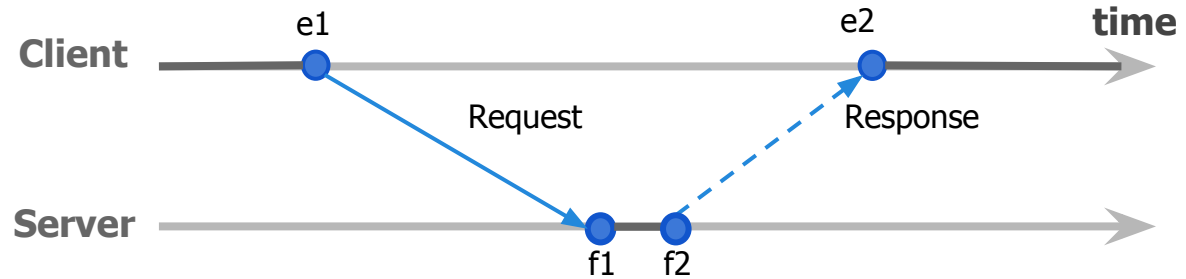
Sockets | Flujo de uso UDP



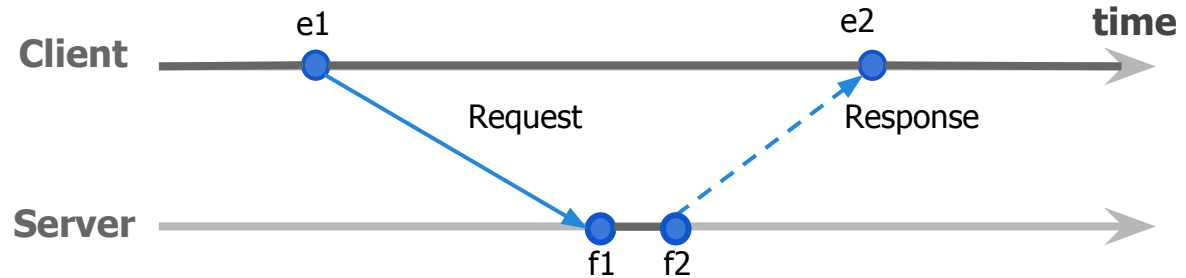


Mensajes Sincrónicos vs Asincrónicos

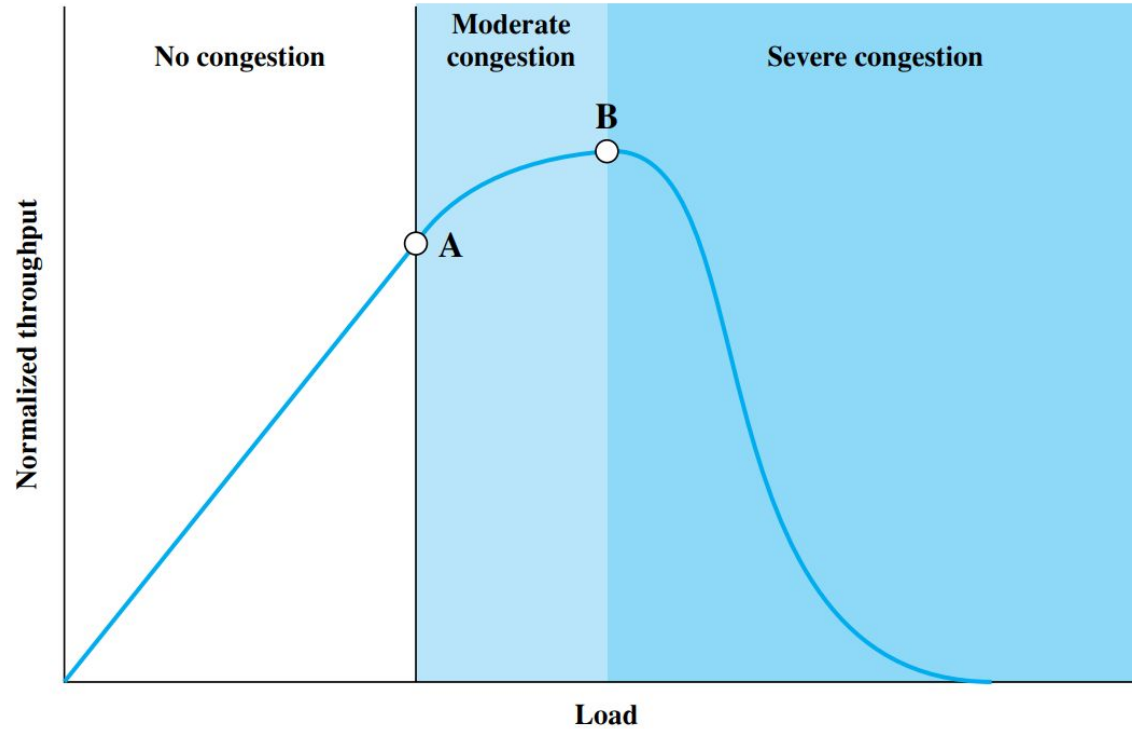
Sincrónico



Asincrónico



Congestión de Red | *Throughput*



Congestión de Red | *Delay*

