

# Nivel aplicación

## Interacción Cliente Servidor

ELO322: Redes de Computadores

Agustín J. González

# Introducción

- Recordar que todo lo visto hasta ahora sólo se justifica por las aplicaciones a nivel aplicación. Otros ejemplos, los sistemas operativos por si mismos no son útiles, ellos están estructurados para servir mejor a los programas que corren sobre ellos. Nos debemos preguntar ¿Cuál es el objetivo final de las redes?
- Las aplicaciones tiene a personas como usuarios; por lo tanto, ellas definen nombres simbólicos para identificar los recursos físicos y abstractos disponibles en una red. Ej. URL, `agv@elo.utfsm.cl`, mateo, lucas.

# Funcionalidad provista por una internet

- Las redes desde el nivel transporte hacia abajo proveen una infraestructura general de comunicación. No indican qué servicio será ofrecido para los usuarios, qué computador correrá qué servicio, cómo estos servicios serán conocidos remotamente, o cómo los servicios serán usados. Todos estos aspectos deben ser cubiertos por las aplicaciones.
- Un programa de aplicación intenta comunicarse con otro en otro computador, y éste responde a los requerimientos. Este modo (cliente-servidor) es uno de los modelos más usados. Es como en el teléfono.

# Modelo Cliente Servidor

- Para que la comunicación entre dos aplicaciones se lleve a cabo uno de los programas de aplicación debe estar esperando por requerimientos por parte del programa llamador.
- Este modelo, un programa espera pasivamente y el otro inicia la comunicación se conoce como el *paradigma de interacción cliente servidor*.
- La aplicación que espera pasivamente es llamada **SERVIDOR** y la que inicia el contacto es llamada **CLIENTE**.

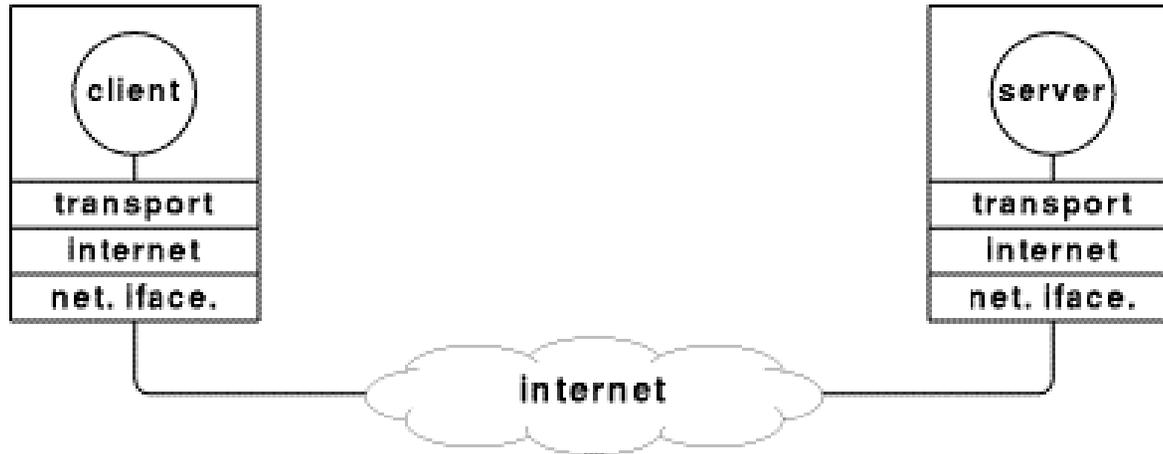
# Características de los Clientes y Servidores

- **Cliente:**
  - Es una aplicación normal que actúa como cliente cuando se requiere acceso remoto.
  - Es invocado directamente por el usuario y tiene una existencia dada por la duración de la sesión del usuario.
  - Corre localmente en el computador del usuario.
  - Inicia activamente el contacto con un servidor.
  - Ejemplo: navegador (netscape, Internet explorer, mozilla, etc)
- **Servidor:**
  - Corre en un computador compartido.
  - Espera pasivamente ser contactado por clientes remotos.
  - Acepta ser contactado por clientes diversos pero ofrece un servicio bien definido.
  - Ejemplo: servidor WEB (apache)

# Ejemplos:

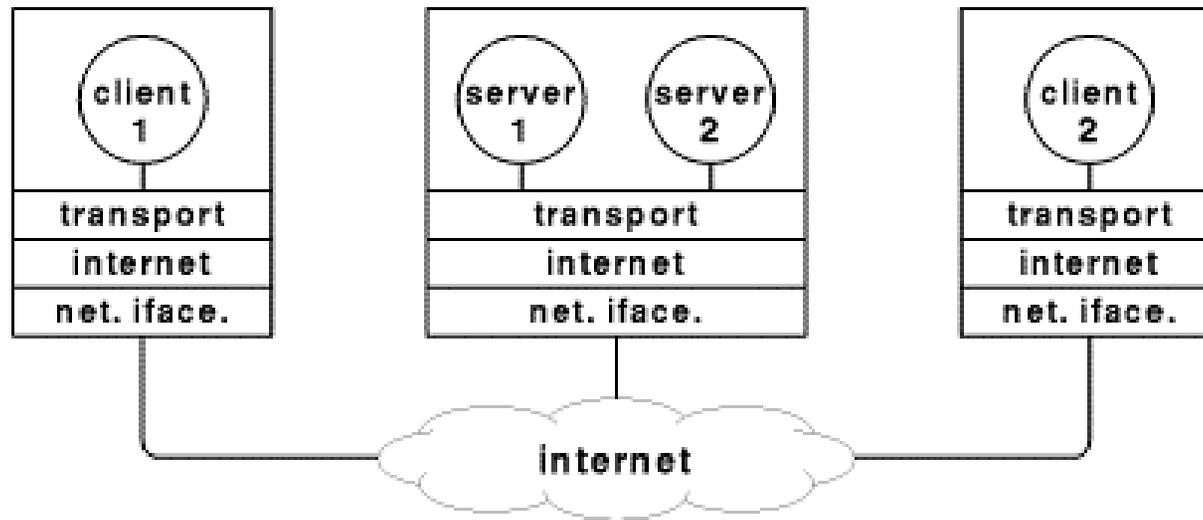
- Samba,
- WWW
- telnet
- ftp
- e-mail
- NFS (network File System)
- X (X-server, X-client)
- NetMeeting
- etc....

# Arquitectura y flujo de información



- El flujo de información viaja en ambas direcciones.
- Las interacciones son de extremo a extremo, es decir los nodos intermedios son transparentes.
- Se requiere un protocolo de transporte. Típicamente se usa TCP. En ocasiones se usa UDP (User Datagram Protocol).

# Servicios múltiples en un computador



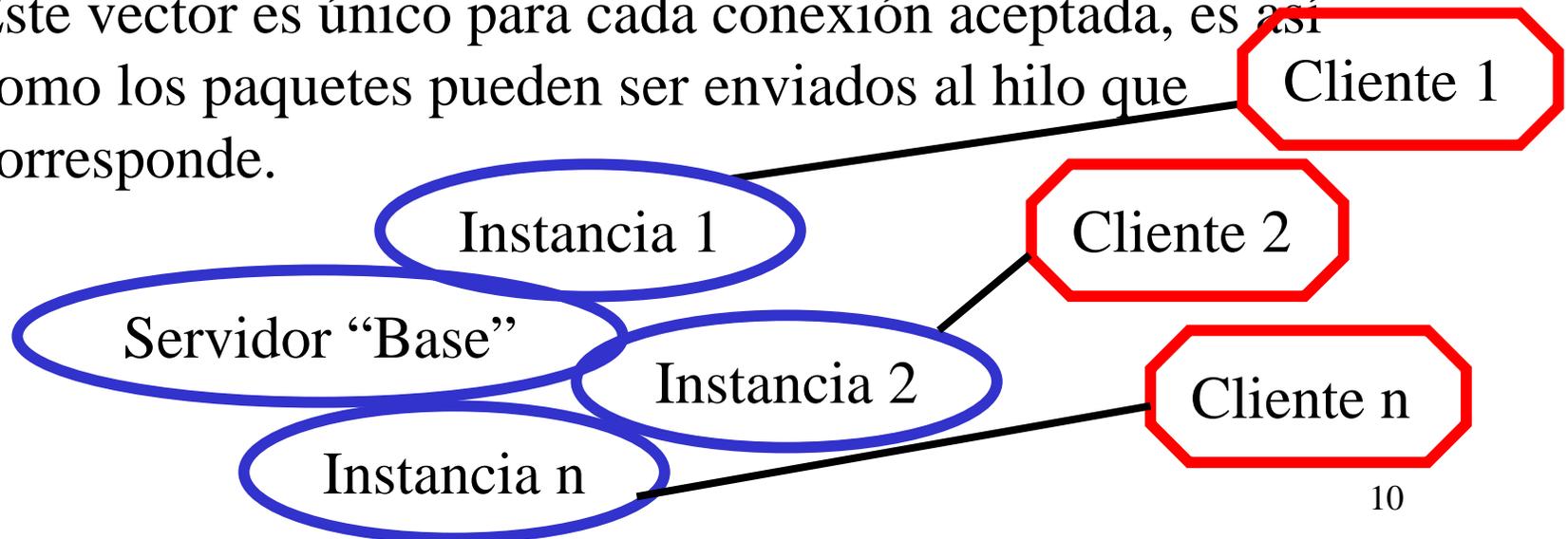
- Un computador normalmente puede correr varios servidores y clientes al mismo tiempo. Ej. Mateo tiene, entre otros, servidor WWW, servidor ssh, servidor ftp, de correo.
- Los computadores que corren servidores son a veces llamados servidores. Éste es un abuso de la terminología.

# Identificación de servicio

- Si hay muchos servidores corriendo y sólo una interfaz, ¿cómo se sabe a quién va dirigido un requerimiento?
- Los protocolos identifican a cada servicio con un identificador único. En TCP éste es el número de puerto. Éste es un número entero positivo de 16 bits. Los servicios estándares poseen un número previamente acordado. Por ejemplo WWW está en el puerto 80.
- `http://www.elo.utfsm.cl:80/~agv/` equivale a `http://www.elo.utfsm.cl/~agv/`

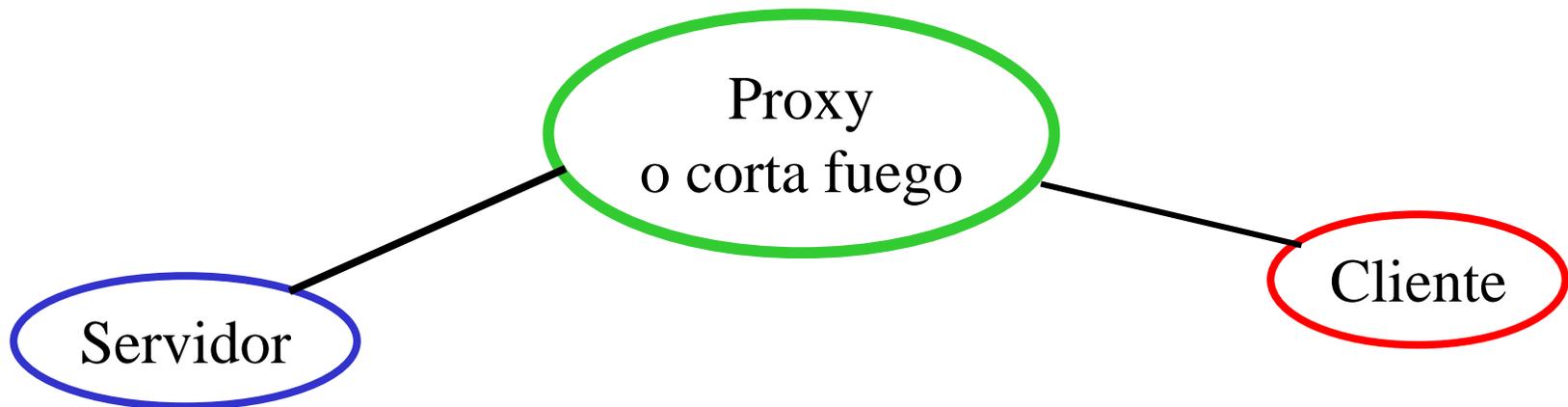
# Identificación no ambigua de comunicaciones

- Si hay varias conexiones hacia un servidor, ¿cómo este computador identifica el hilo que debe hacerse cargo de cada requerimiento? (todos van al mismo puerto y máquina)
- El TCP identifica una conexión por la 4-tupla pareja (**máquina local, puerta local, máquina remota, puerta remota**)
- Éste vector es único para cada conexión aceptada, es así como los paquetes pueden ser enviados al hilo que corresponde.



# Proxy y Corta Fuegos

- Un servidor puede transformarse en un cliente para satisfacer un requerimiento.
- Ejemplo una proxy, un corta fuego:



- Ojo: cuidado con los ciclos !!!