

Tema 6 – Nivel de aplicación

Redes de Computadores

Grado en Ingeniería Informática en Tecnologías de
la Información

Índice

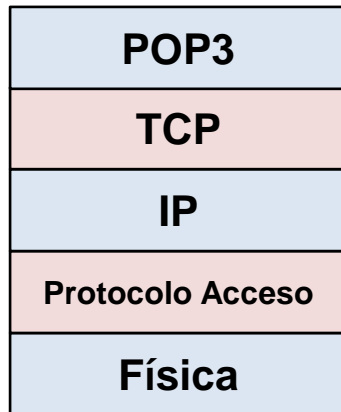
- Introducción
- Ejemplos de protocolos de nivel de aplicación
- El modelo cliente/servidor
- Arquitectura centralizada – Servidores y nube
- Arquitectura distribuida – El modelo P2P

Introducción

- Está formado por un conjunto de protocolos:
 - Cada uno de ellos se utiliza para un propósito específico
 - Cada uno de los protocolos es independiente
 - Pueden convivir varios dentro de una red y dispositivo
 - Son utilizados por aplicaciones a las que se denomina **servicios**
 - Utilizan servicios extremo a extremo del nivel de transporte

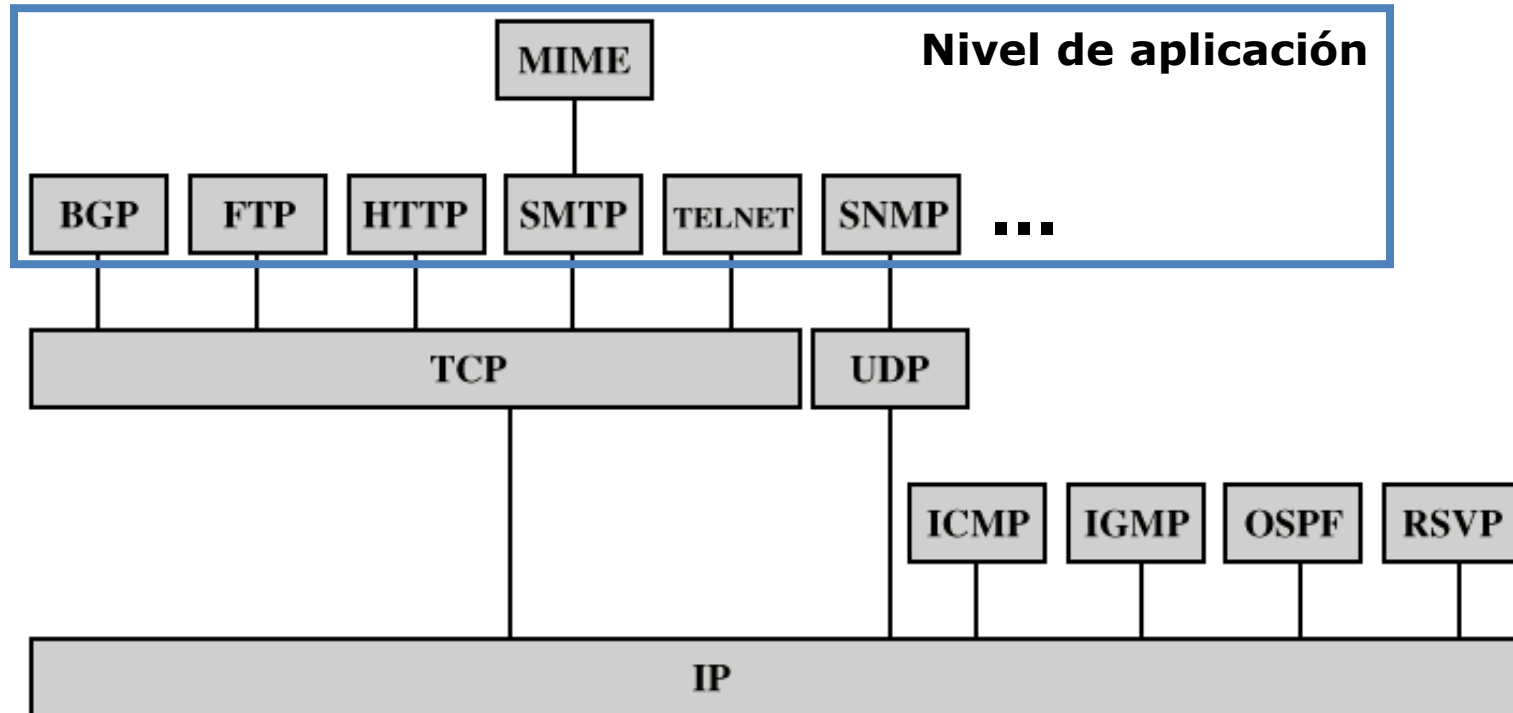
Introducción

- Ejemplo de protocolo de nivel de aplicación



```
Servidor: +OK Hello there.
Cliente: USER alumno
Servidor: +OK Password required.
Cliente: PASS Pass_Alumno
Servidor: +OK logged in.
Cliente: STAT
Servidor: +OK 1 15216
Cliente: LIST
Servidor: +OK POP3 clients that break here, they violate STD53.
      1 15216
.
Cliente: RETR 1
Servidor: +OK 15216 octets follow.
      Date: Mon, 04 Dec 2006 20:00:57 +0100
      From: Profesor <profesor@uniovi.es>
      Subject: Prueba e-mail
      To: alumno@uniovi.es
      ...
.
Cliente: DELE 1
Servidor: +OK Deleted.
Cliente: QUIT
Servidor: +OK Bye-bye.
```

Introducción



BGP = Border Gateway Protocol

FTP = File Transfer Protocol

HTTP = Hypertext Transfer Protocol

ICMP = Internet Control Message Protocol

IGMP = Internet Group Management Protocol

IP = Internet Protocol

MIME = Multi-Purpose Internet Mail Extension

OSPF = Open Shortest Path First

RSVP = Resource ReSerVation Protocol

SMTP = Simple Mail Transfer Protocol

SNMP = Simple Network Management Protocol

TCP = Transmission Control Protocol

UDP = User Datagram Protocol

Introducción

- Protocolos de **servicios orientados al usuario**

- HTTP
- SMTP, POP3, IMAP
- TELNET, SSH
- RFB, ICA, RDP
- FTP
- RTSP
- SIP
- ...

Utilizados por servicios a los que el usuario accede directamente

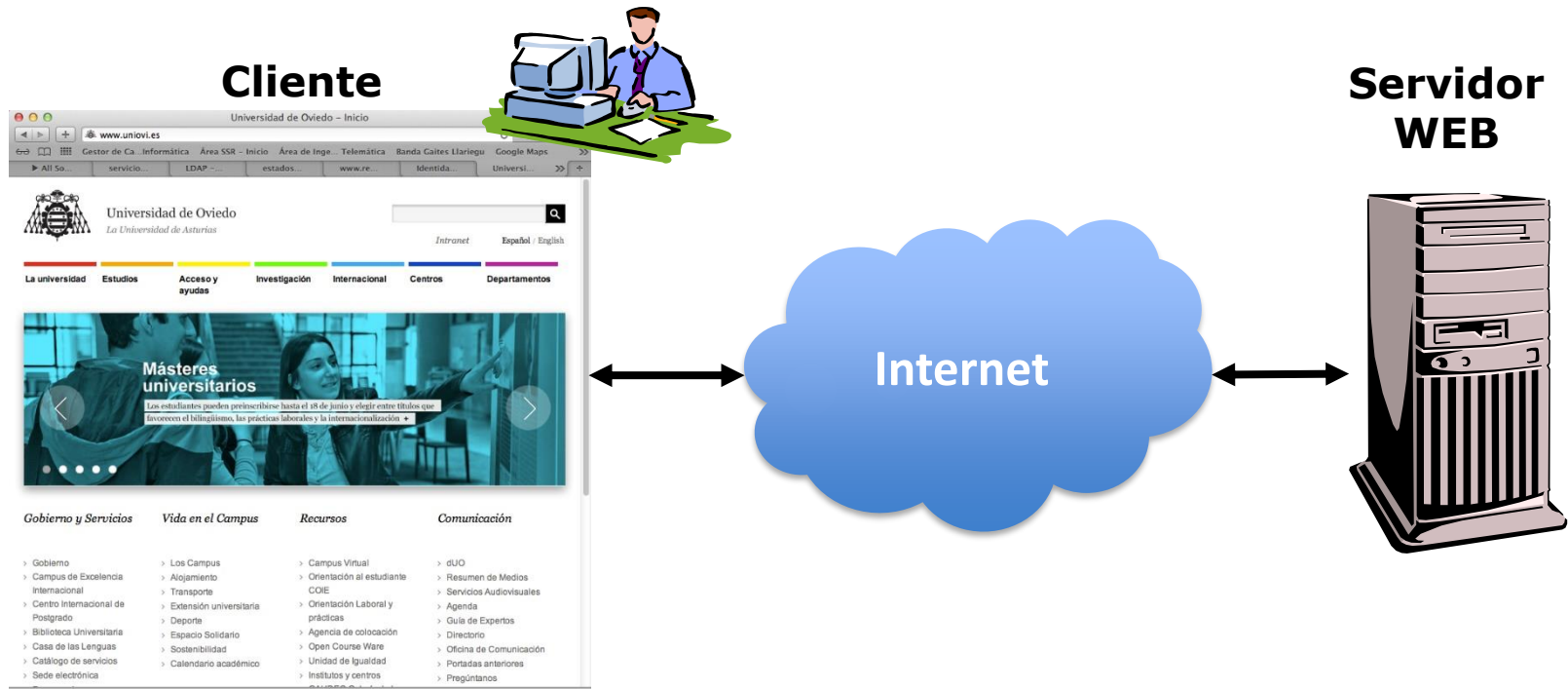
- Protocolos de **servicios básicos**

- DNS
- DHCP
- NTP
- LDAP
- SNMP
- ...

Utilizados por servicios base para el funcionamiento de la red o de otros servicios

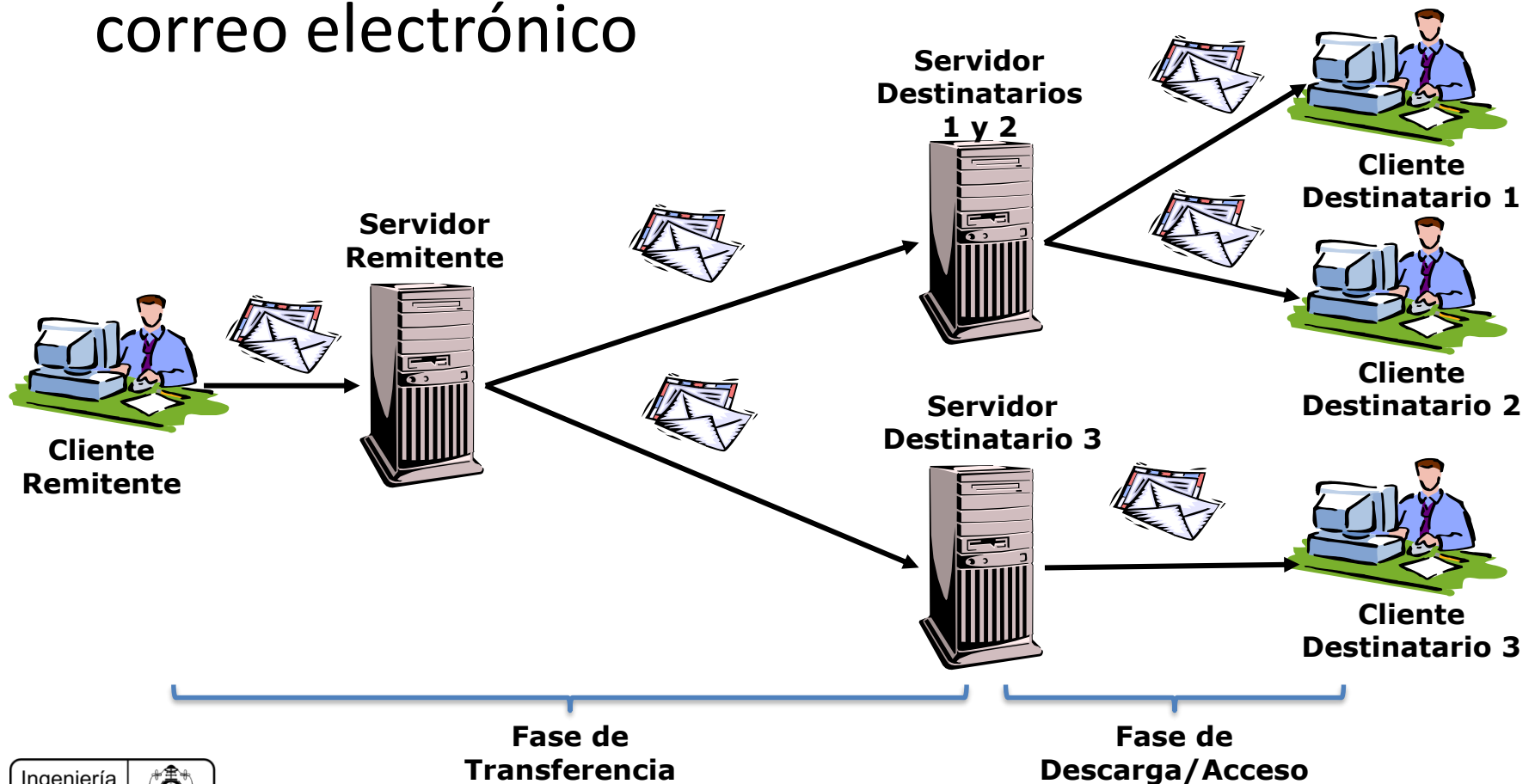
El protocolo HTTP

- Protocolo para la transferencia de ficheros de hipertexto
- Base para los servicios Web
- Sobre una capa de cifrado se le conoce como HTTPS



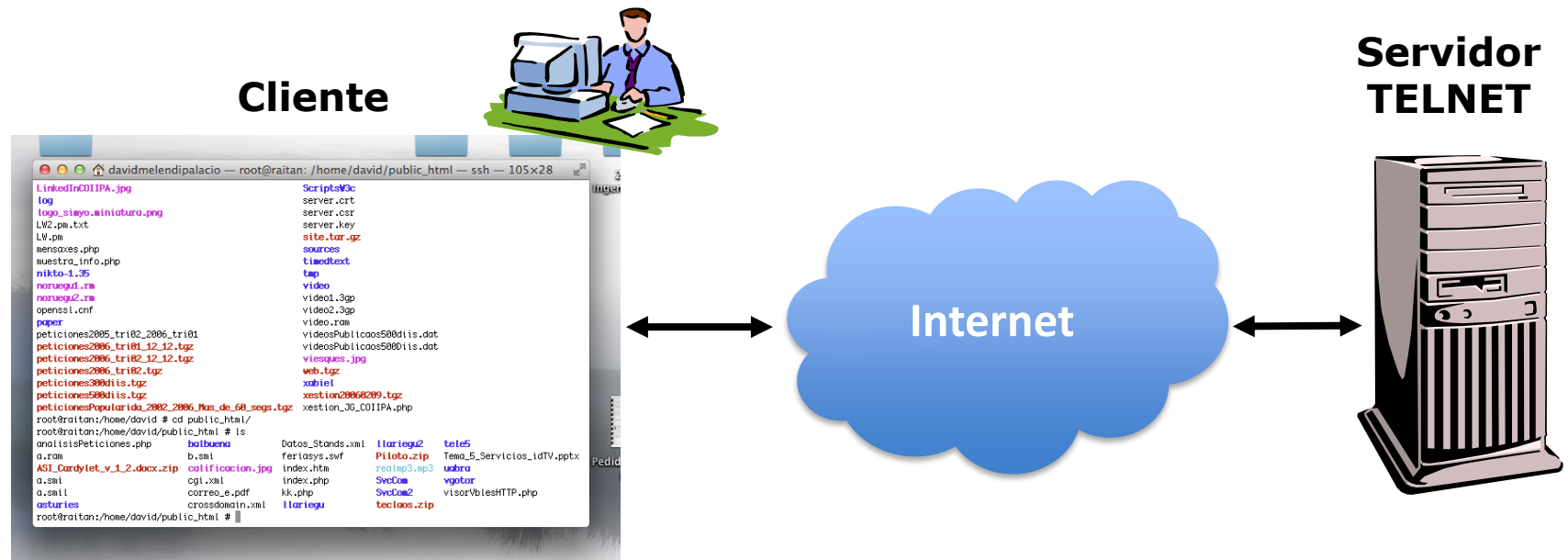
Protocolos SMTP, POP3, IMAP

- Protocolos para el envío de mensajes de correo electrónico



Protocolos TELNET y SSH

- Protocolos para el trabajo mediante terminal remota
- Permiten trabajar desde una localización remota con la consola de un computador



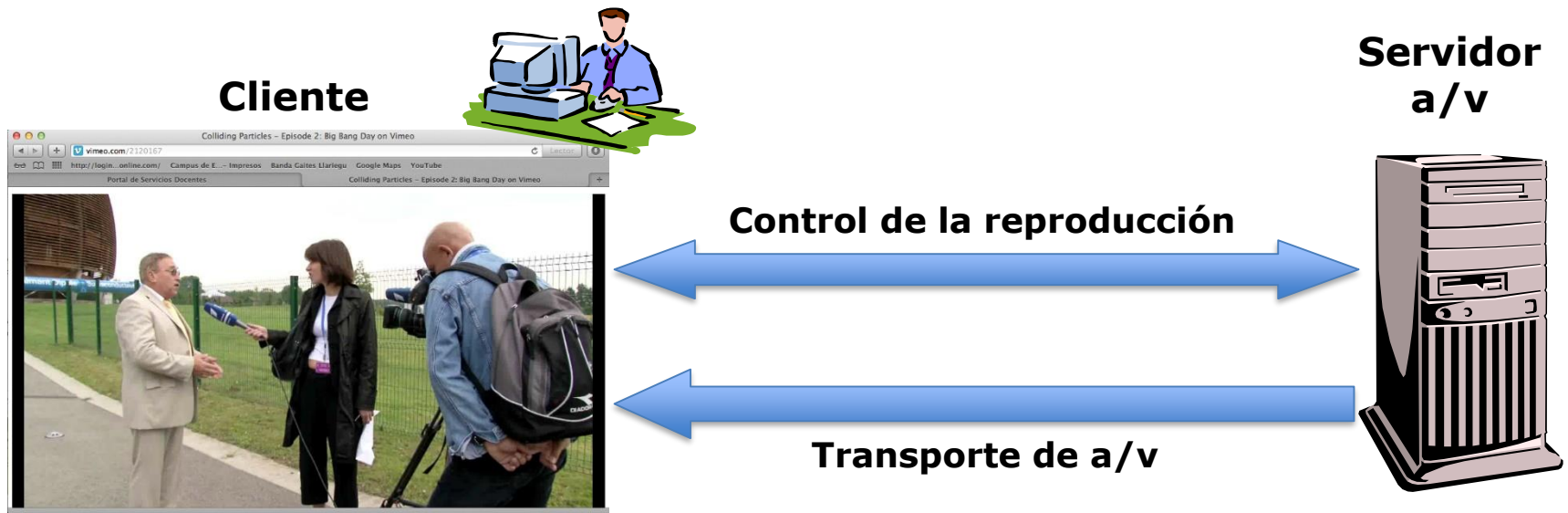
El protocolo FTP

- Protocolo utilizado para la transferencia de archivos entre máquinas remotas



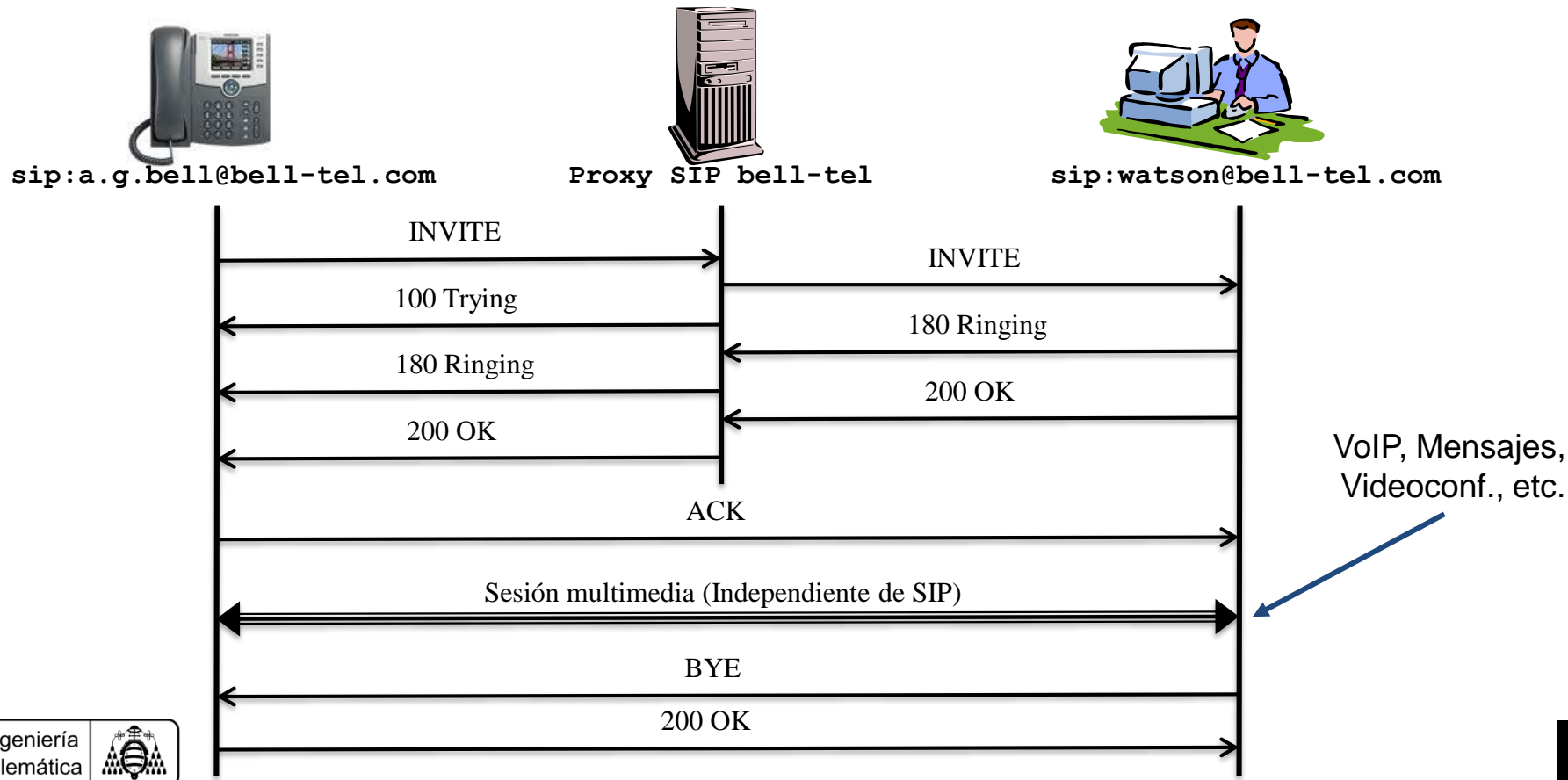
El protocolo RTSP

- Protocolo para el control de servicio multimedia basados en tecnología de streaming
- Únicamente realiza el control del sistema
- Utiliza otros protocolos para el transporte de la información



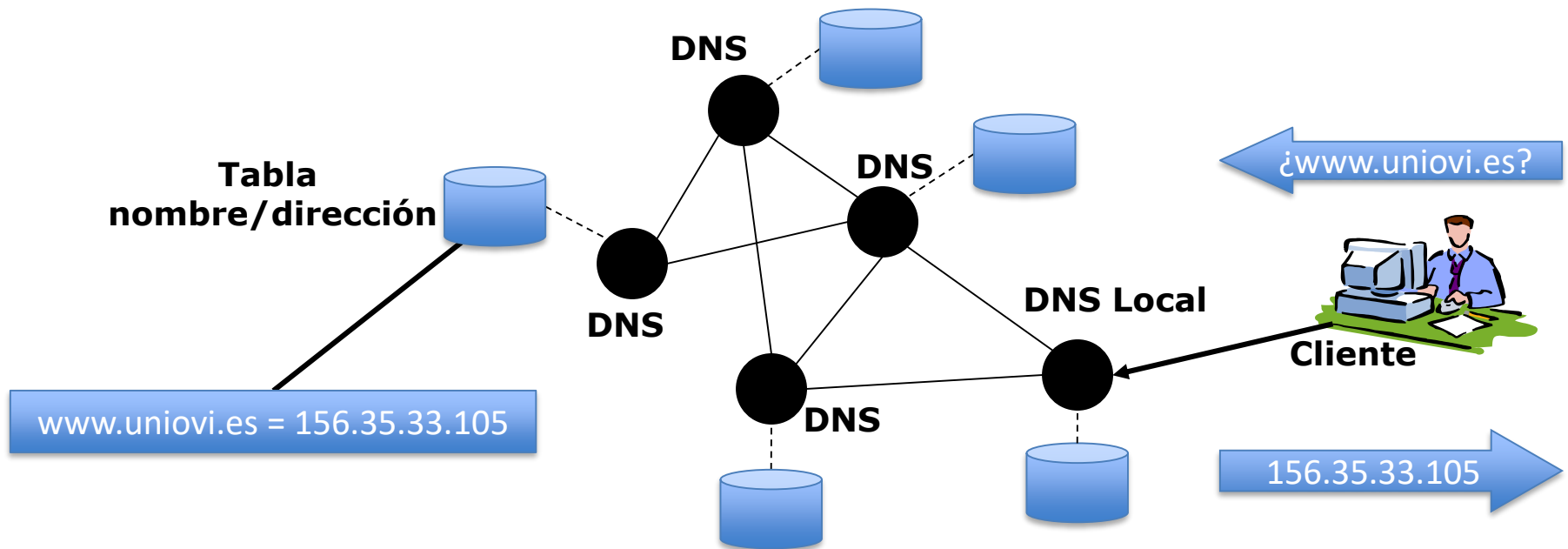
El protocolo SIP

- Protocolo de control y señalización para la creación, modificación y finalización de sesiones de uno o más participantes
- Sesión: llamada de VoIP, mensajería, videoconferencia, ...



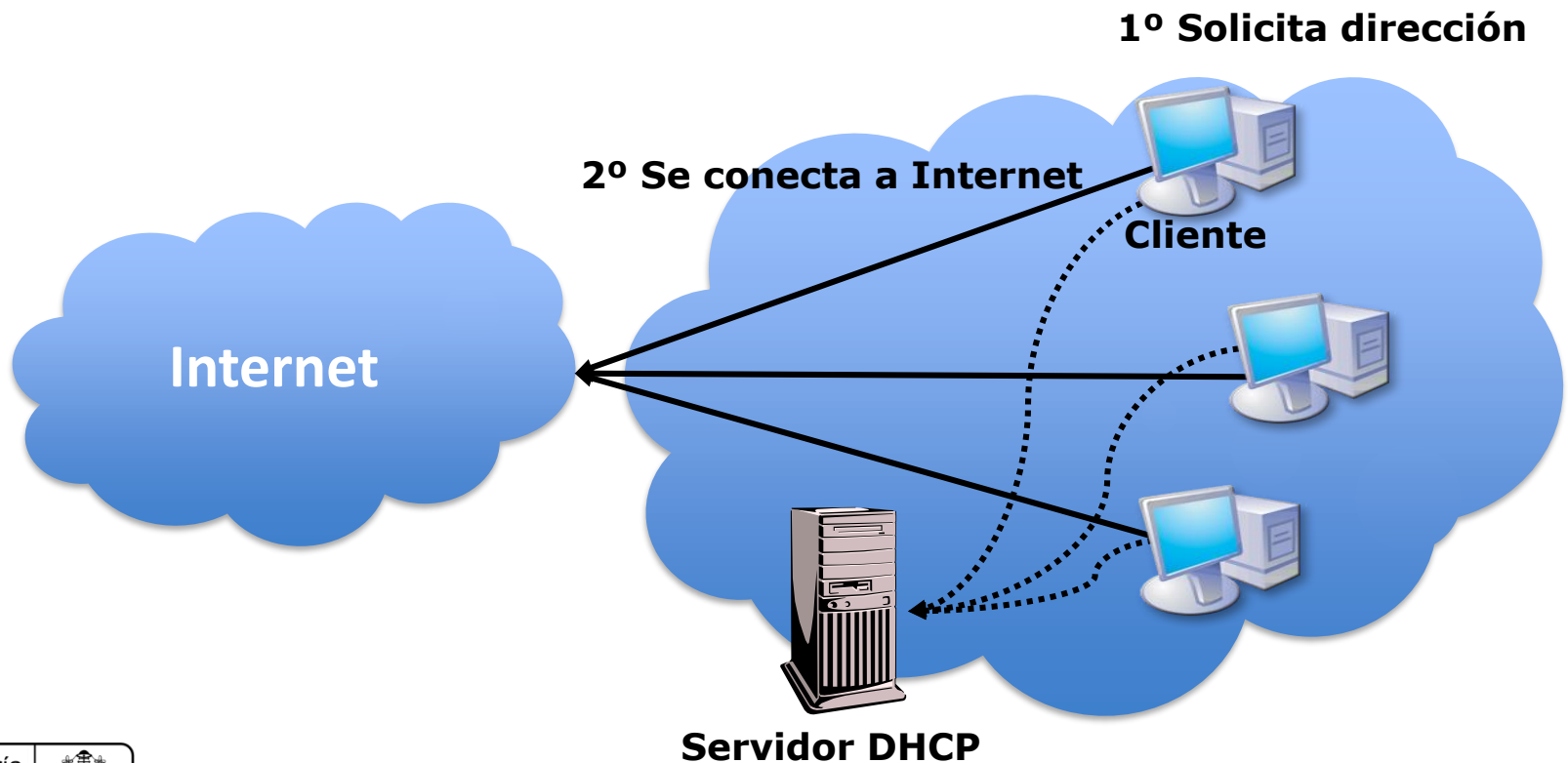
El protocolo DNS

- Protocolo para la resolución de nombres
- A partir del nombre lógico de una máquina resuelve su dirección IP



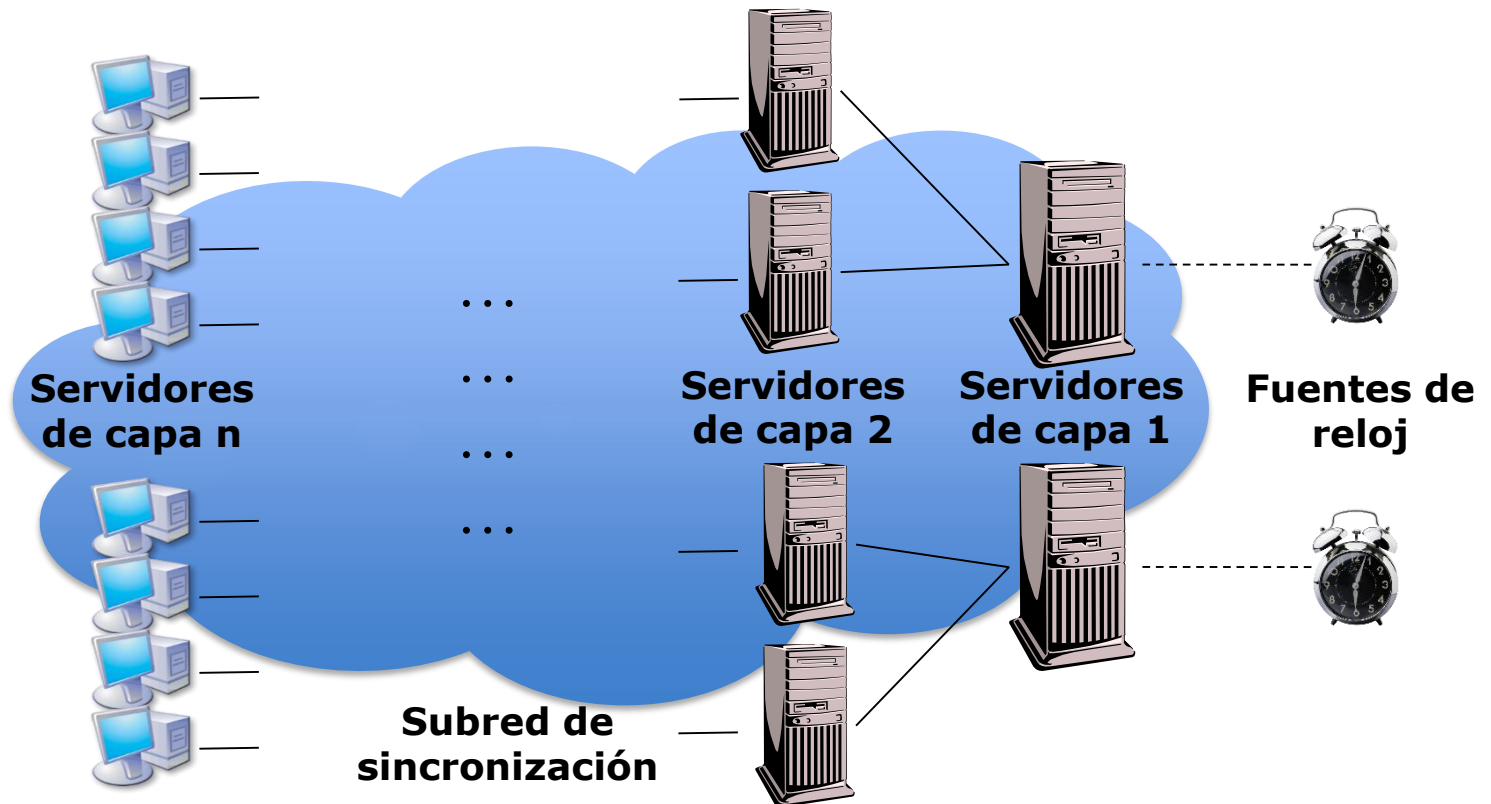
El protocolo DHCP

- Protocolo para el reparto de direcciones IP de forma dinámica



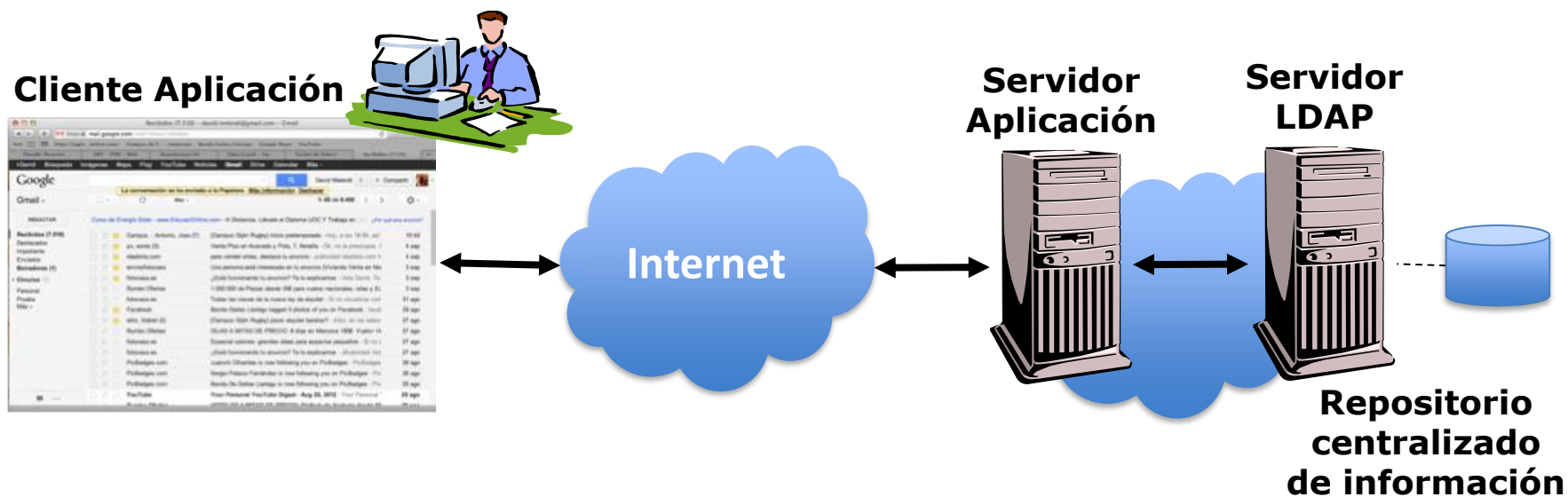
El protocolo NTP

- Permite el acceso y la distribución de señales de reloj precisas



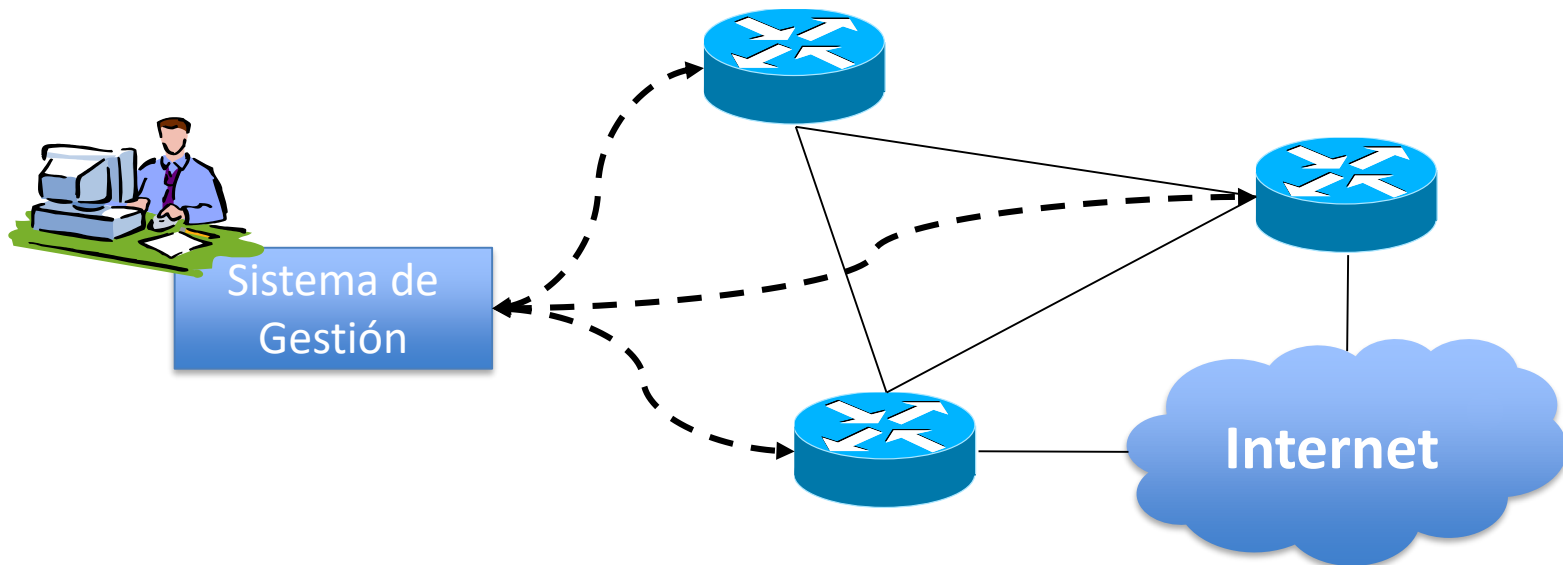
El protocolo LDAP

- Implementa un servicio de directorio
- Altamente optimizado para lectura de datos

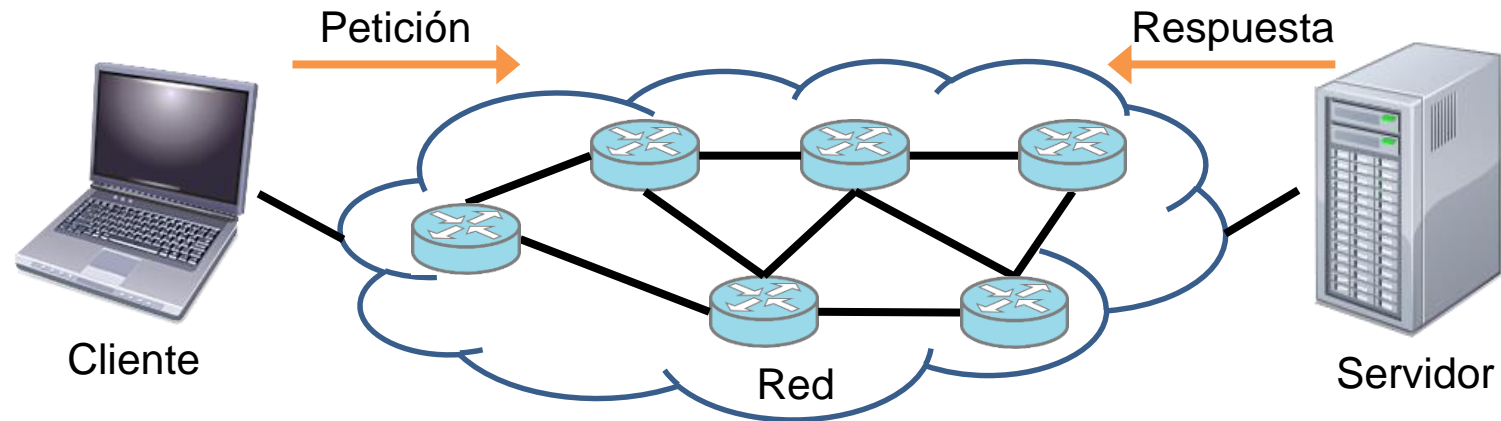


El protocolo SNMP

- Protocolo para la gestión de red
 - Monitorización y control



El modelo cliente/servidor



- **Programa Servidor:** ofrece un servicio y acepta peticiones de clientes
- **Programa Cliente:** se comunican con el servidor para solicitar un servicio

Cliente/servidor: Ejemplo simple

- Consideremos un servidor de eco basado en UDP
 - El servidor seguirá el siguiente proceso:
 - Negociará con el Sistema Operativo un puerto UDP
 - Entrará en un ciclo sin fin con las siguientes tareas:
 - Espera a que un datagrama llegue al puerto de eco
 - Invierte las direcciones de origen y destino
 - Devuelve el datagrama al emisor original
 - El cliente seguirá el siguiente proceso:
 - Envía un mensaje al programa servidor de eco
 - Espera la respuesta

El modelo cliente/servidor

- Características del servidor
 - Host siempre disponible
 - Dirección IP y puerto bien conocido
 - Su ejecución debe comenzar antes que la ejecución de los clientes
 - Puede atender a varios clientes
- Características del cliente
 - Es el que inicia la comunicación con el servidor
 - Puede tener dirección IP dinámica y puerto aleatorio
 - No se comunica directamente con otro cliente

Cliente/servidor: Complejidad en los servidores

- Habitualmente los servidores no son tan sencillos como el del ejemplo:
 - Procesamiento de peticiones de forma concurrente
 - Aspectos relacionados con la seguridad
- En cuanto a la concurrencia, los servidores suelen tener dos partes:
 - Un proceso **maestro** sencillo, responsable de aceptar las nuevas peticiones
 - Varios **esclavos**, responsables de manejar cada una de las peticiones

Cliente/servidor: Complejidad en los servidores

- Técnicas de gestión de esclavos:
 - **Esclavos por petición**: cada vez que llega una petición se crea un esclavo para procesarla
 - **Esclavos por sesión**: cada vez que se inicia una sesión se crea un esclavo para gestionarla (una sesión contiene una o varias peticiones)
 - **Conjunto de esclavos**: el servidor tiene inicialmente un conjunto de esclavos activos inicialmente que va repartiendo según llegan las peticiones. Cuando estas terminan los esclavos se liberan (pero no se destruyen).
Subtipos:
 - Conjunto de esclavos con asignación por petición
 - Conjunto de esclavos con asignación por sesión

Cliente/servidor: Complejidad en los servidores

- Problemas de seguridad
 - Protección del sistema y de los recursos
 - Deben mantener reglas de autorización y protección
 - Restringir el acceso a ciertas zonas
 - Integridad
 - Deben protegerse contra peticiones formadas equivocadamente y contra peticiones que causen la interrupción del programa

Cliente/servidor: Complejidad en los servidores

- Las arquitecturas de los servicios pueden ser más complejas.
 - Reducir el **consumo de recursos**
 - Garantizar la **disponibilidad** del servicio
 - Garantizar la **escalabilidad** del servicio
 - Incrementar la **seguridad** del servicio
- Otros elementos:
 - Cachés
 - Proxies
 - Repartidores de carga
 - Firewalls, IDS e IPS

Arquitectura centralizada – Servidores y nube

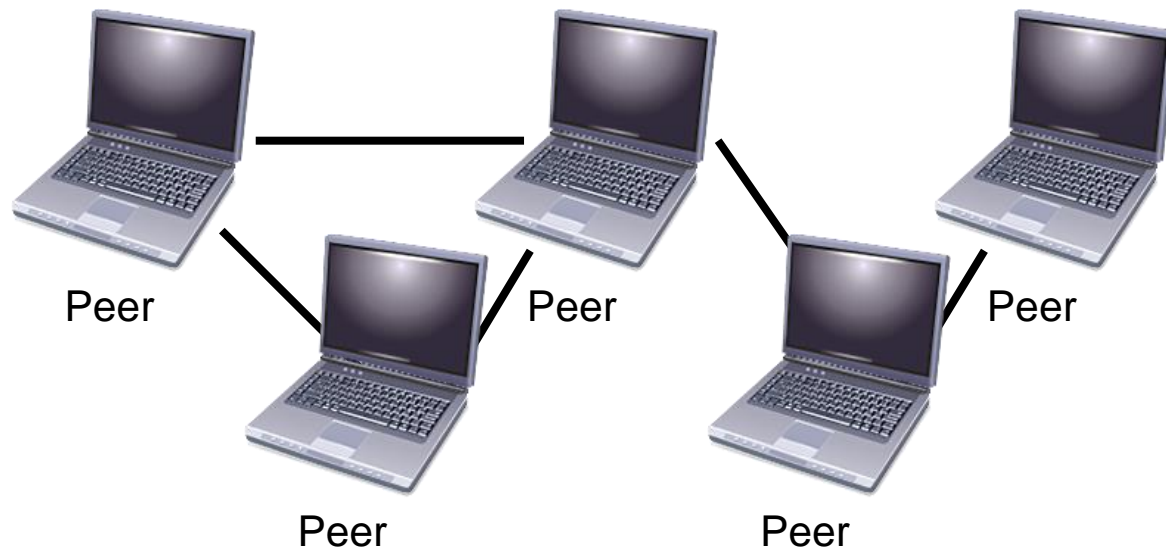
- El cliente puede delegar parte de sus funciones en el servidor
 - Videojuegos en *streaming*
 - Comandos de voz telefonía móvil
- La mayoría de servicios se trasladan al servidor
- Mayor necesidad de recursos y escalabilidad – Computación en la nube

Servicios en la nube

- Ventajas
 - Facilidad de escalado
 - Posible ahorro económico
 - Delegación de problemas técnicos: *PaaS (Platform as a Service)*
- Inconvenientes
 - No se tiene acceso físico a los servidores
 - Cesión la información del servidor a terceras empresas

Arquitectura distribuida – El modelo P2P

- Distribuye la información en vez de concentrarla en un servidor
- Se consideran todos los nodos iguales a la hora de compartir la información
 - Todos los nodos pueden dar y recibir
 - No existe un proveedor centralizado
 - Las comunicaciones son simétricas



Arquitectura distribuida – El modelo P2P

- Ventajas
 - **Escalabilidad**: Es muy fácil unir nuevos nodos
 - **Descentralización**: La información no se almacena únicamente en un servidor
 - **Coste**: El gasto se reparte entre los diferentes nodos
 - **Robustez**: No hay un único punto de fallo
- Inconvenientes
 - La información está distribuida entre múltiples nodos que a priori desconocemos
 - Un nodo malicioso puede causar grandes problemas a toda la red

P2P: Funcionamiento básico

- Funcionamiento básico
 - Se **localizan** otros pares que tengan la información deseada → proveedores
 - Herramientas de búsqueda, información en la web, servidores centralizados, *superpares*, ...
 - Se **descarga** la información
 - Si hay más de un proveedor la información se divide en porciones y se descarga cada porción de un par
 - Se **cede** la información tan pronto como se tenga