

Aplicaciones y Servicios en Redes

Tema 03. Aplicaciones y Servicios Web



Alberto Eloy García Gutiérrez

Luis Sánchez González

DPTO. DE INGENIERÍA DE COMUNICACIONES

Este tema se publica bajo Licencia:

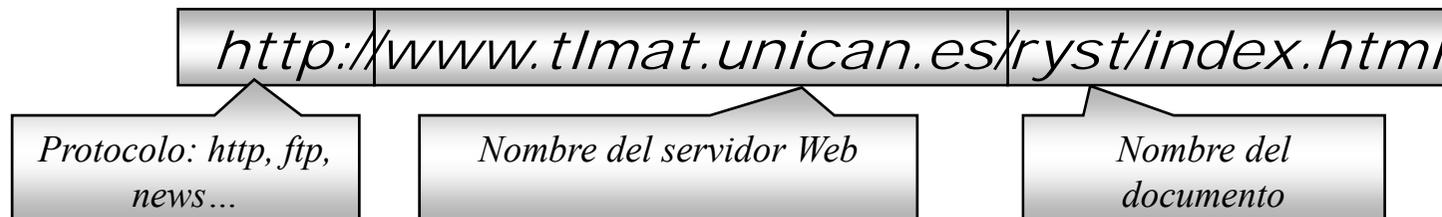
[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

World Wide Web

- Aplicación para acceder a información multimedia distribuida
- El cliente de este servicio se conoce como **navegador**
- Se basa en el modelo del hipertexto, pero los enlaces pueden referenciar documentos en cualquier servidor
- El cliente establece conexión (TCP) con el servidor en el puerto 80
- En la comunicación entre ambos se utiliza el protocolo HTTP (*HyperText Transfer Protocol*)
- (HTTP1.0 RFC 1945; HTTP1.1 RFC 2068)

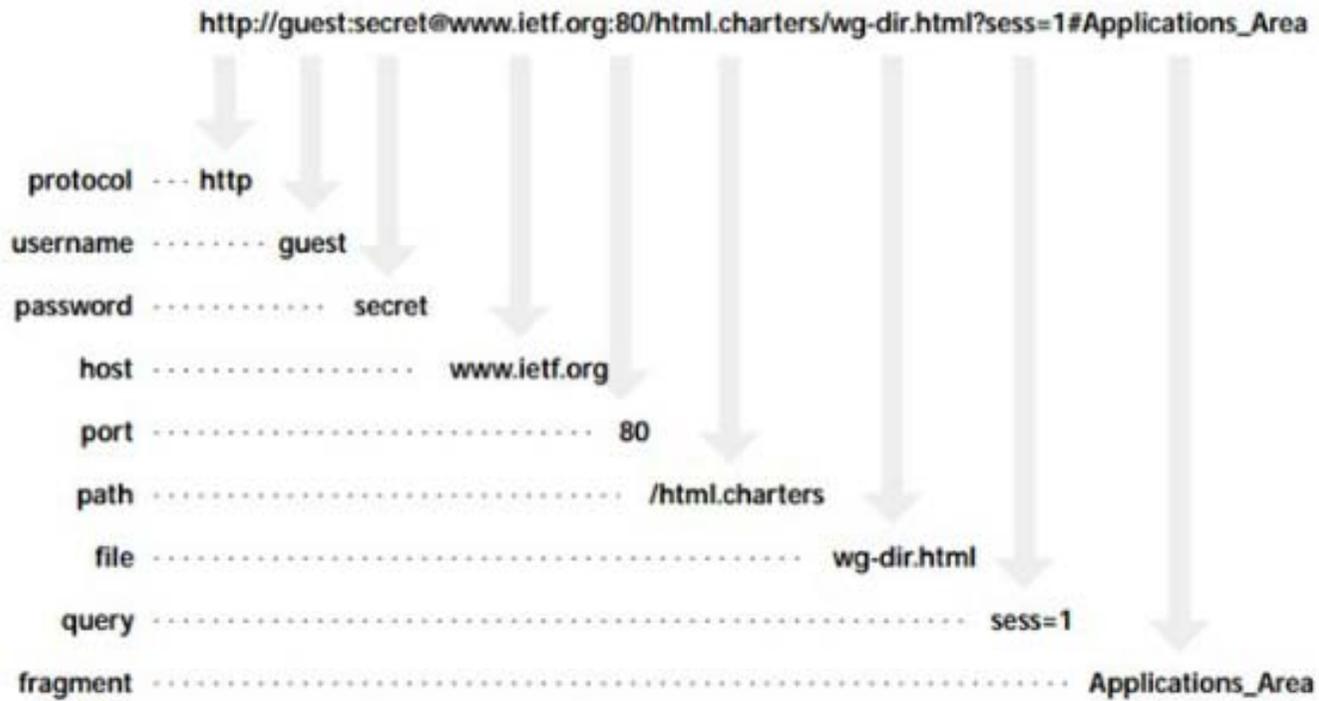
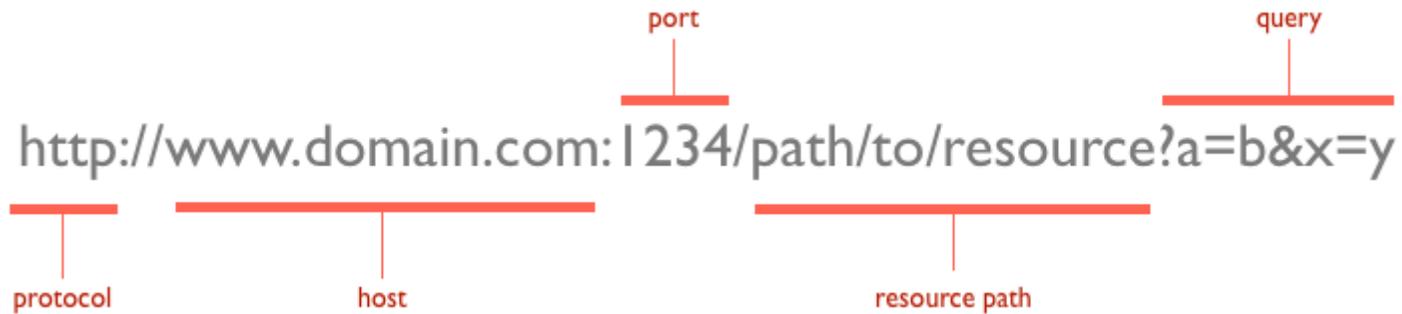
WWW: Formato URL (Uniform Remote Location)

- Esta notación expresa de manera uniforme los distintos recursos que podemos acceder con el cliente Web (RFC 1738 y 1808).
- Consta de varios campos:



- Esta es una **URL completa**
- En ocasiones es suficiente con la **URL parcial**

URL: Ejemplos

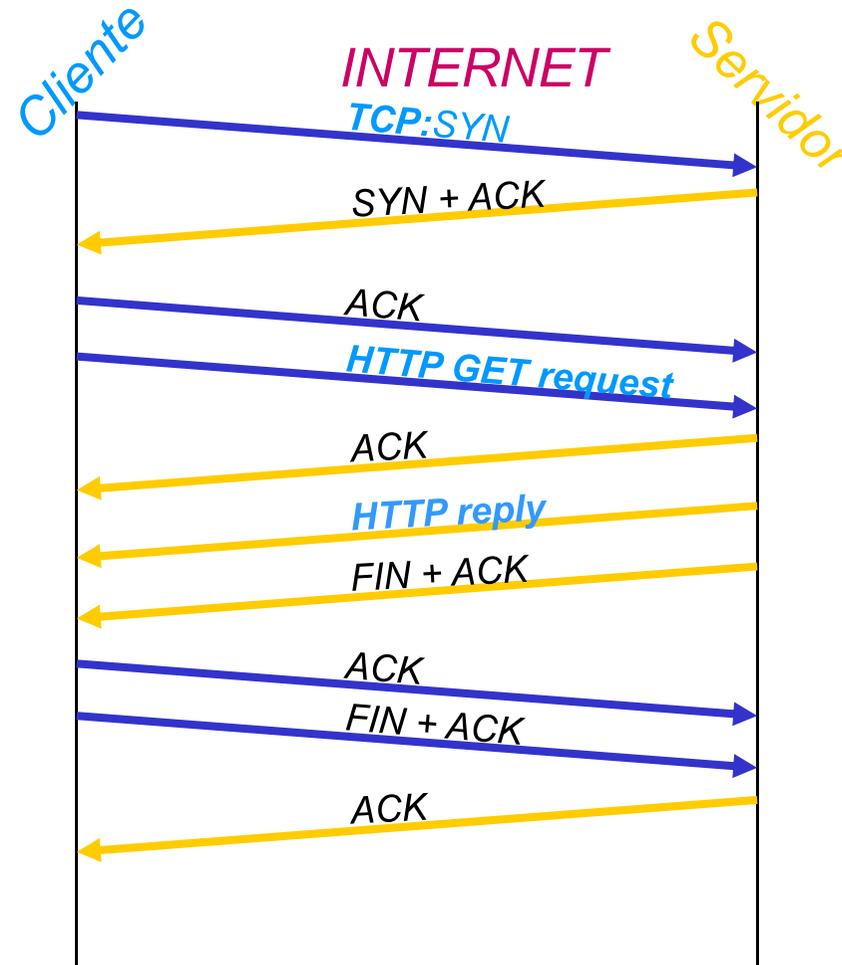


WWW: HTTP 1.0

- Establecida la conexión con el servidor, el cliente envía un comando:
 - **GET /ryst/index.html HTTP/1.0**
<CR LF> <CR LF>

(Si la conexión es directa al servidor se usa la URL parcial)

- El servidor envía el documento HTML y cancela la conexión
- Para conseguir cada una de las imágenes (iconos, textos, dibujos, etc.) de una página el cliente debe establecer una nueva conexión TCP



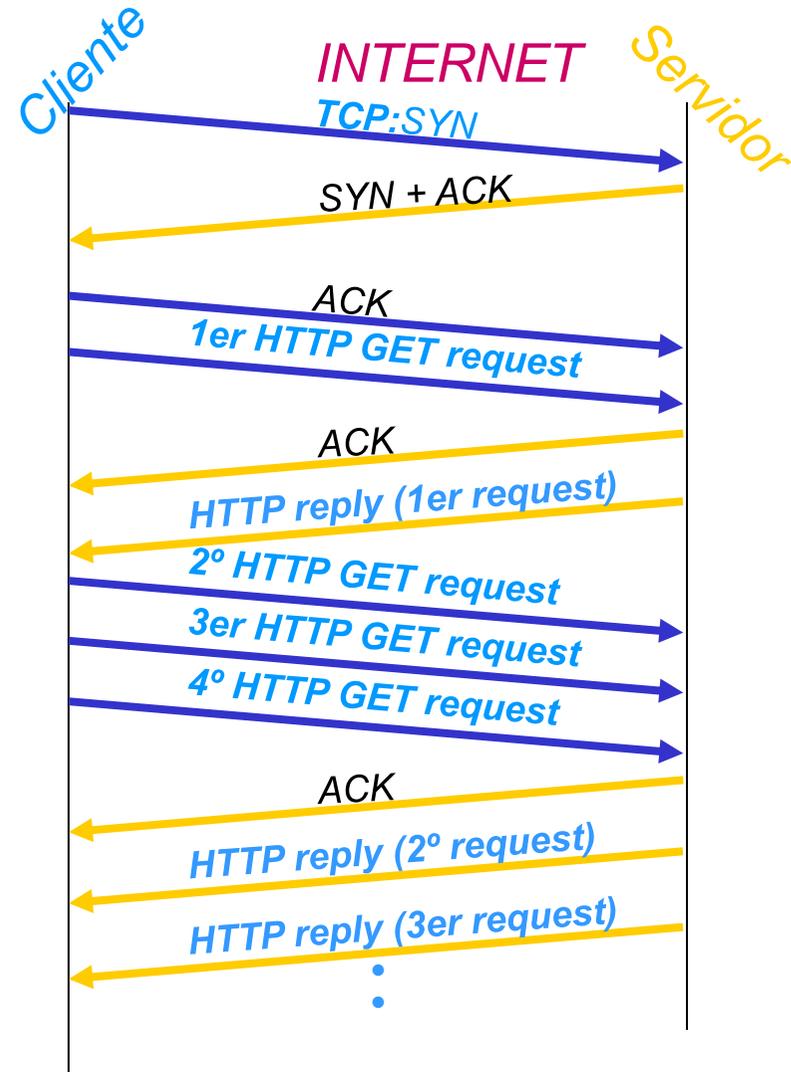
WWW: HTTP 1.1

- En HTTP 1.0 se abre una nueva conexión para cada petición
- Problema:
 - Protocolo a 3 bandas para abrir y cerrar las conexiones TCP
 - Slow-start en TCP
- Solución:
 - En HTTP 1.1 la conexión se mantiene abierta hasta que servidor o cliente decidan cerrarla
 - Por ejemplo: después de un tiempo de inactividad

CONEXIONES PERSISTENTES

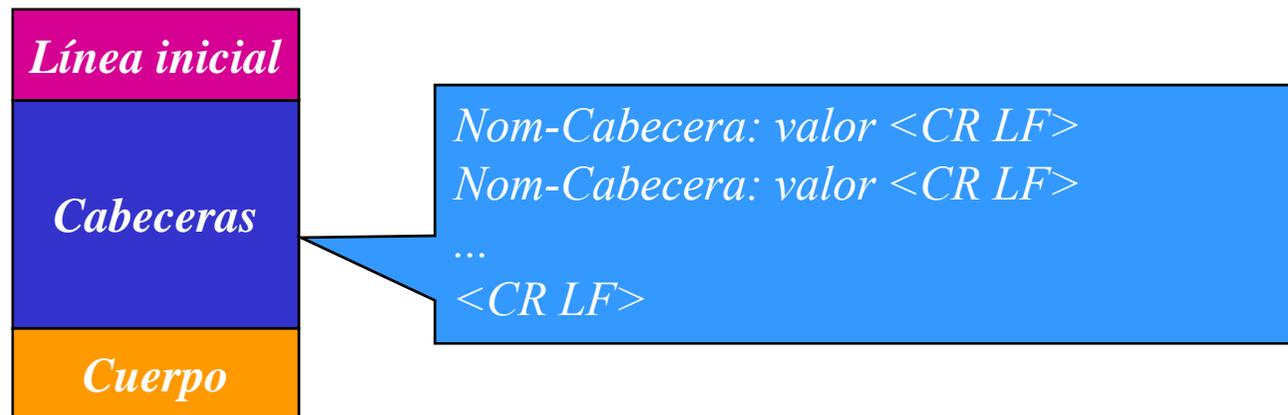
HTTP: Pipelining

- Aún se puede mejorar la utilización de la conexión:
 - El cliente envía la petición y espera la respuesta
 - El servidor envía la respuesta y espera la siguiente petición
- HTTP 1.1 permite “**pipelining**”
= enviar las peticiones sin esperar que se haya recibido la respuesta



HTTP: Formato de transacciones

- El mismo formato básico para peticiones y respuestas:
 - La línea inicial es obligatoria (diferente contenido según sea petición o respuesta)
 - Las cabeceras son opcionales. Puede haber cero o más
 - El cuerpo es necesario en algunos casos (peticiones y respuestas)



HTTP: Líneas de Inicio (petición)

- El formato de la línea inicial de una petición es:

Método <URL> versión

Ejm:

GET /path/to/file/index.html HTTP/1.0

- GET
 - Es el método más común
- HEAD
 - Similar al GET, pero se solicita al servidor que conteste sólo con las cabeceras (sin el cuerpo del recurso)
 - Útil para conocer las características del recurso sin bajarlo
- POST
 - Se utiliza para enviar datos a un servidor:
 - Formularios
 - Datos para un programa que se ejecuta en el servidor (CGI)
 - Los datos se envían dentro del cuerpo

HTTP: Otros comandos

- **PUT:** actualiza un recurso existente, por lo que los datos pueden incluir la versión actualizada *del recurso*.
- **DELETE:** borra un recurso existente
- **TRACE:** utilizado para obtener el número de saltos hasta el servidor. Cada proxy o router introduce su dirección IP o nombre DNS en los datos de la cabecera VIA. Se utiliza para diagnóstico.
- **OPTIONS:** utilizado para obtener las capacidades del servidor y así poder rehacer la solicitud de forma acorde.

HTTP: Líneas de inicio (respuesta)

- También se conoce como línea de estado
- Ejemplos:
 - HTTP/1.0 200 OK
 - HTTP/1.0 404 Not Found
 - HTTP/1.0 500 Server Error
- Códigos de estado
 - 1xx indica una información al cliente
 - 2xx indica éxito en la operación
 - 3xx redirecciona al cliente a otra URL
 - 4xx indica un error por parte del cliente
 - 5xx indica un error por parte del servidor

HTTP: Códigos de estado

▪ 1xx: Mensajes de información

- Fueron introducidos provisionalmente en HTTP/1.1 and is purely provisional. El servidor puede enviar al cliente 100-continue message, indicándole que continúe enviando la petición, o ignorarlo si ya ha sido enviada. HTTP/1.0 lo ignora.

▪ 2xx: Successful

- Indica al cliente que la petición ha sido procesada:
 - **202** Petición aceptada pero el recurso podría no estar incluido en la respuesta.
 - **204** No content: No hay cuerpo en la respuesta.
 - **205** Reset Content: el cliente debe reiniciar la vista.
 - **206** Partial Content: La respuesta solo incluye parte del contenido.

▪ 3xx: Redirection

- El cliente tiene que rehacer la petición hacia otra URL.
 - **301** Moved Permanently: El recurso se encuentra en una nueva localización.
 - **303** See Other: El recurso está temporalmente en otra localización.
 - **304** Not Modified: El recurso no ha sido modificado y el cliente puede utilizar el contenido de la caché.

▪ 4xx: Client Error

- El servidor piensa que el cliente error está cometiendo algún fallo
 - **400 Bad Request:** Petición incorrecta
 - **401 Unauthorized:** El cliente tiene que rehacer la petición con la cabecera Authorization. En caso que ya existiera, las credenciales serían incorrectas.
 - **403 Forbidden:** El servidor deniega el acceso al recurso.
 - **404** Recurso inválido y no existe en el servidor.
 - **405 Method Not Allowed:** Operación no permitida o no soportada por el servidor.
 - **409 Conflict:** El servidor no puede completar la petición porque el cliente intenta modificar el recurso.

▪ 5xx: Server Error

- Indican fallo en el servidor:
 - **500 Internal Server Error.**
 - **501 Not Implemented:** Funcionalidad no implementada.
 - **503 Service Unavailable:** El servidor no es capaz de responder, posiblemente porque algún subsistema ha fallado.

HTTP: Cabeceras

- Dan información sobre la petición o la respuesta
- El formato de una cabecera es:

Nombre-Cabecera: valor <CR LF>

- HTTP 1.0 define 16 cabeceras, (ninguna obligatoria)
- HTTP 1.1 define 46 cabeceras, sólo una (Host:) es obligatoria en las peticiones
- Otras cabeceras habituales:
 - **User-Agent:** el navegador que usa el cliente
 - **Accept:** tipo de contenidos que son aceptados

HTTP: Ejemplo de cabeceras

- Ejemplo cabeceras en la petición:

GET /easy/http/ HTTP/1.0

If-Modified-Since: Wed, 13 Sep 2000 22:51:57 GMT; length=45531

Referer: http://www.google.com/search?q=http+tutorial

Connection: Keep-Alive

User-Agent: Mozilla/4.76 (Windows NT 5.0; U)

Pragma: no-cache

Host: www.jmarshall.com

Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg

Accept-Encoding: gzip

Accept-Charset: iso-8859-1,*,utf-8

Ejemplo cabeceras en la respuesta:

HTTP/1.1 304 Not Modified

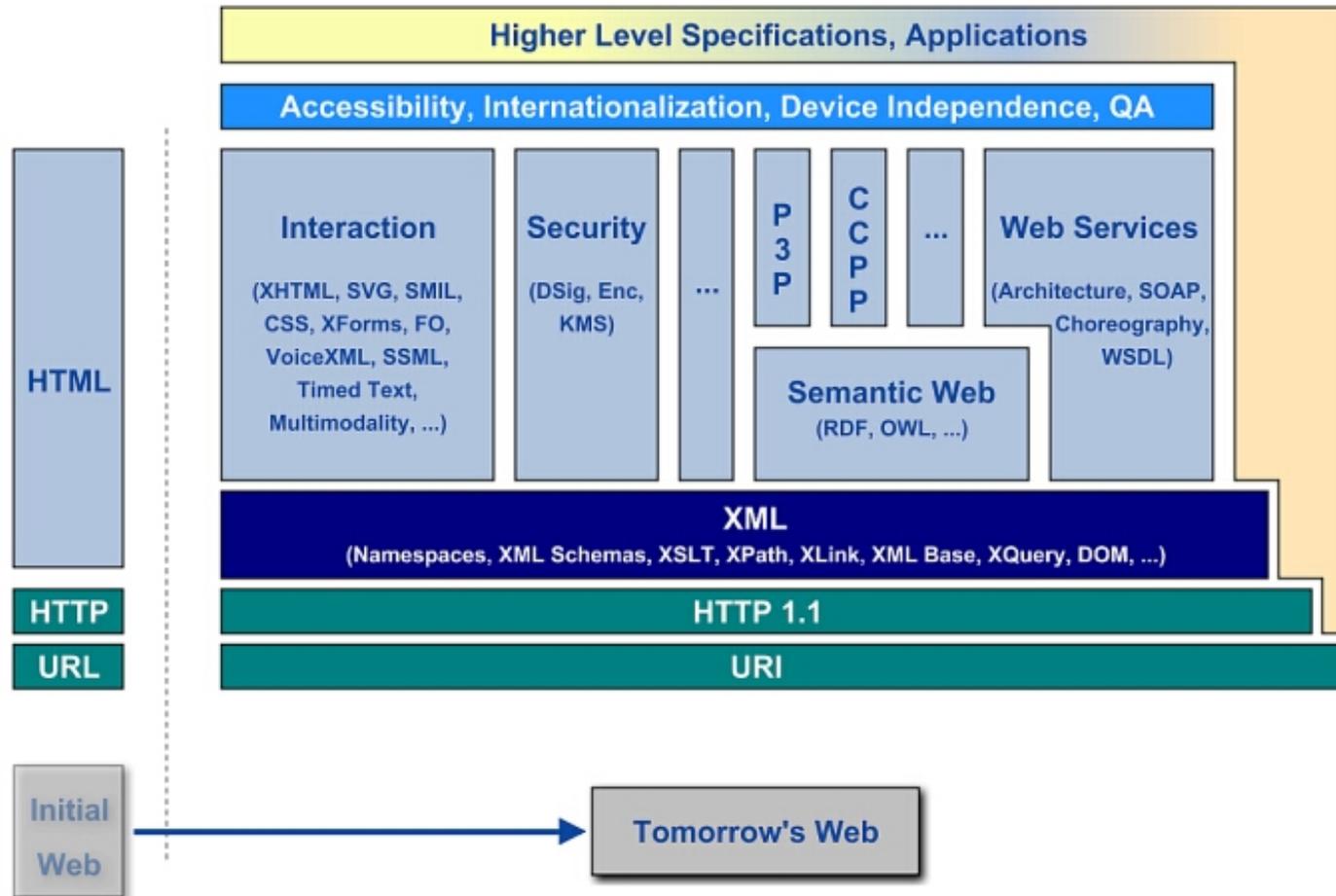
Date: Fri, 07 Dec 2001 11:29:00 GMT

Server: mod_jk FrontPage/4.0.4.3 Confluence
Apache/1.3.20 (Unix)

Connection: Keep-Alive

Keep-Alive: timeout=15, max=100

WWW: Protocolos



- **X)HTML(eXtended HyperText Markup Language):**
 - Versión 3.2 Enero 1997, 4.01 Diciembre 1999). Utiliza la sintaxis de XML desde Enero de 2000. Se basa en XHTML profiles, adaptados al tipo de dispositivos utilizados: Ejm. XHTML Basic, diseñado para clientes Web en teléfonos móviles, PDAs, “buscas” y settop boxes
- **CSS (Cascade Style Sheets):**
 - Distingue entre estructura y presentación. Define estilos para facilitar la gestión y accesibilidad. Permite el control de la presentación de formatos HTML y XML
- **XML 1.0 (eXtensible Markup Language):**
 - Considerado precursor de la NGWeb. Integra lenguajes diseñados a medida en una infraestructura común XML. XSLT permite transformar XML a XHTML or SVG. XLink y XML Base definen el vocabulario de hiperenlaces XML
- **DOM (Document Object Model):**
 - API para acceso a estructuras, estilos, acciones, etc. No está asociado a ningún lenguaje específico, permitiendo en diseño de Webs activas, manipulando XHTML y XML
- **MathML 2.0:**
 - Herramienta para representación matemática en XML
- **PNG (Portable Network Graphics):**
 - Formato de ficheros extensible para almacenamiento de imágenes raster

- SVG (Scalable Vector Graphics):
 - Soporte para gráficos vectoriales. Basado en XML, permite realizar animaciones. SVG Basic and Tiny permite soporte gráfico para PDA y móviles
- SMIL (Synchronized Multimedia Integration Language):
 - permite realizar presentaciones multimedia
- RDF (Resource Description Framework):
 - Define el modelo y la sintaxis para incluir metadatos en cualquier recurso Web
- XML-Signature y XML Encryption (Febrero y Diciembre 2002):
 - Define el marco para añadir firmas digitales y recursos encriptados a la Web
- P3P (Platform for Privacy Preferences):
 - otorga al usuario el control sobre su información personal a la hora de visitar recursos Web
- Web Accessibility Guidelines:
 - Promueve el acceso Web para discapacitados

WWW: HTML (*HyperText Mark-up Language* - RFC 1866)

- Lenguaje de descripción de páginas WWW
 - Describe cómo se visualizarán por pantalla los elementos de texto: párrafos, listas, tablas, etc.
- Permite incluir diferentes tipografías así como gráficos, sonidos, vídeos, etc
- Se basa en la inserción de "marcadores de control" junto con el texto.
Ejemplo:

<TITLE>Bienvenido a la Web </TITLE>

- Permite hacer uso de hiperenlaces:
 - Conexión a documentos relacionados
 - Los hiperenlaces apuntan a documentos en el mismo o en cualquier otro servidor de Web
- Los enlaces se consiguen enmarcando con la orden `<A>` al objeto (texto o gráfico) que representa el enlace
- Para especificar un nuevo documento en el enlace se emplea la notación URL

Pulse aquí para visitar la UC

WWW: Ejemplo transacción HTML

- *GET /index.html HTTP/1.0*

HTTP/1.0 200 OK

Date: Monday, 26-May-97 09:18:17 GMT

Server: NCSA/1.3

MIME-version: 1.0

Content-type: text/html

Last-modified: Monday, 12-May-97 07:49:06

Content-length: 2018

<html>

<head>

<title>TELEMATICA</title>

<meta name="GENERATOR" content="Vermeer FrontPage 1.0">

</head>

<body background="http://tmat.unican.es/fondo.jpg">

<bgsound src=<http://193.144.186.2/git/sounds/raiders.mid> loop="infinite">

<h1> Bienvenidos al GIT</h1>

<p>Esta es la página del Grupo de Ingeniería Telemática: </p>

....

</html>

Connection closed by foreign host.

Evolución de HTML

- 1991 HTML
- 1994 HTML 2
- 1996 CSS 1 + JavaScript
- 1997 HTML 4
- 1998 CSS 2
- 2000 XHTML 1
- 2002 Tableless Web Design
- 2005 AJAX
- 2009 HTML 5

Aplicaciones Web

- Además de la transferencia de código HTML, los Servidores web pueden entregar aplicaciones web. Éstas son porciones de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP:
 - **Aplicaciones en el lado del cliente:** el cliente web es el encargado de ejecutarlas en la máquina del usuario.
 - Java "applets" o Javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Otros lenguajes hacen uso de plugins.
 - **Aplicaciones en el lado del servidor:** el servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.
 - PHP 1995 Multiplataforma 5.3.5
 - ASP.Net 1998 Windows (Algunas versiones) 4.0
 - Perl 1987 Multiplataforma 5.12.3
 - Python 1991 Multiplataforma 3.2.0
 - Ruby 1995 Multiplataforma 1.9.3-p125

Servidores Web

- Algunos servidores web importantes son:
 - Nginx
 - Apache
 - Internet Information Services (IIS)
 - Cherokee
 - Tomcat
- Otros servidores, más simples pero más rápidos, son:
 - lighttpd
 - thttpd
- Existen numerosas aplicaciones que facilitan la instalación automática de servidores web Apache y aplicaciones adicionales como Mysql y PHP (entre otros), de forma conjunta, como XAMPP, JAMP o EasyPHP
 - LAMP en plataformas Linux
 - WAMP en sistemas Windows
 - MAMP en sistemas Apple Macintosh

Servicio Web

- Un servicio Web es un recurso programable y direccionable mediante una URL
- Permiten a las aplicaciones compartir información
- Permiten a las aplicaciones invocar funciones de otras aplicaciones

Página Web	Servicio Web
Puede ser visitada por cualquier usuario	Solo lo visitan aquellos programas que lo requieran

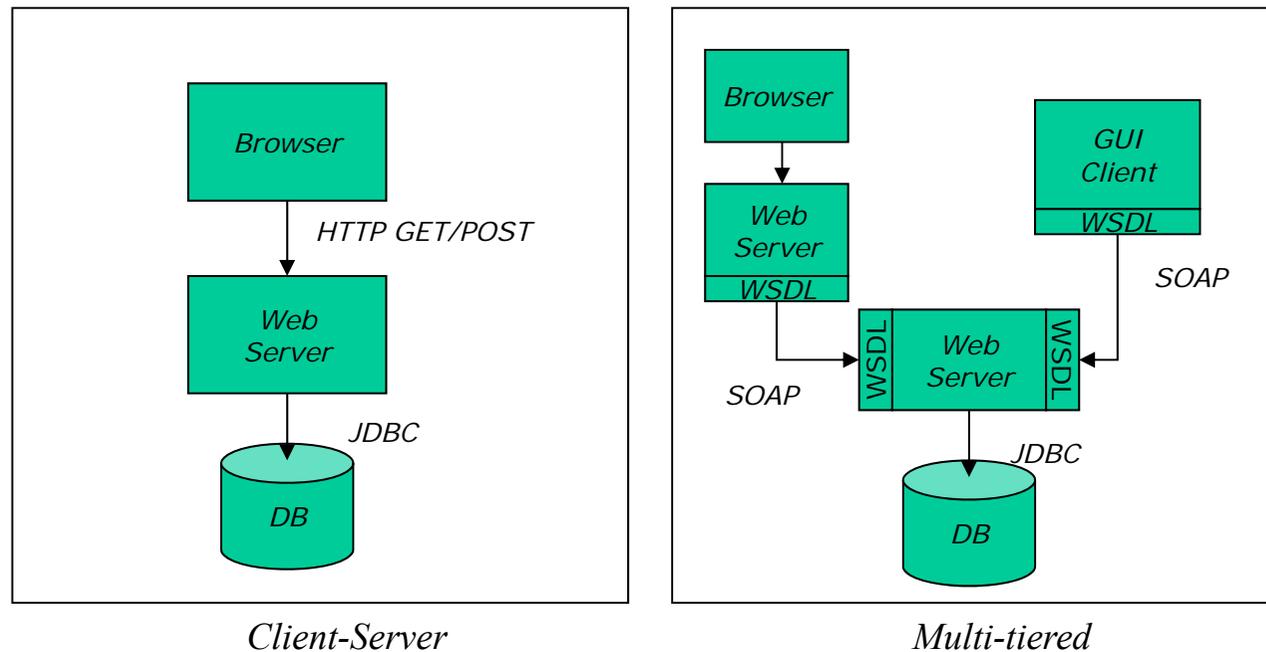
- El conjunto de Servicios Web es una WWW paralela, de carácter no humano (M2M)

Web Services

Web Services

- Hewlett-Packard crea e-Speak en 1999
- Microsoft introduce el nombre de “Web services” en 2000
- Todos los grandes fabricantes/operadores tienen líneas relacionadas

Servlets/CGI vs. Web Services



Web Services = XML (???)

- XML (extensible Markup Language) es un lenguaje de tags para describir datos estructurados.
- Estandarizado por <http://www.w3.org>
- Es extensible:
 - XML no impone un conjunto de tags.
 - Todos los documentos tienen un tag raíz.
 - Los tags se abren y se cierran y en medio pueden tener otros tags anidados.
 - Los tags pueden tener atributos
- Separa los datos y su propia definición (XSD, DTD) de la representación (XSL).
- Parser XML:
 - Está formado por un conjunto de clases que permite comprobar que el documento está bien formado y acceder a los datos.
 - Si además el parser es validante, también permite validar el documento (usando su DTD, esquema, etc.)

XML: Ejemplo

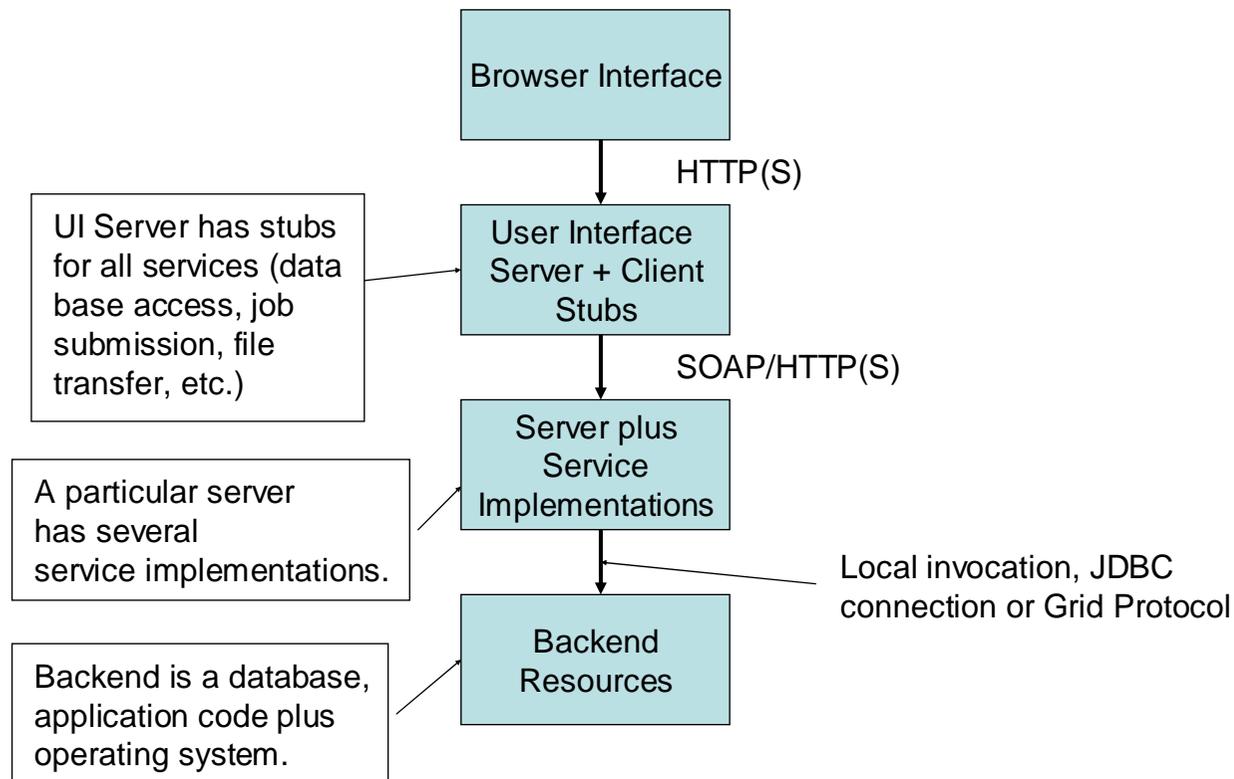
- Movies.xml

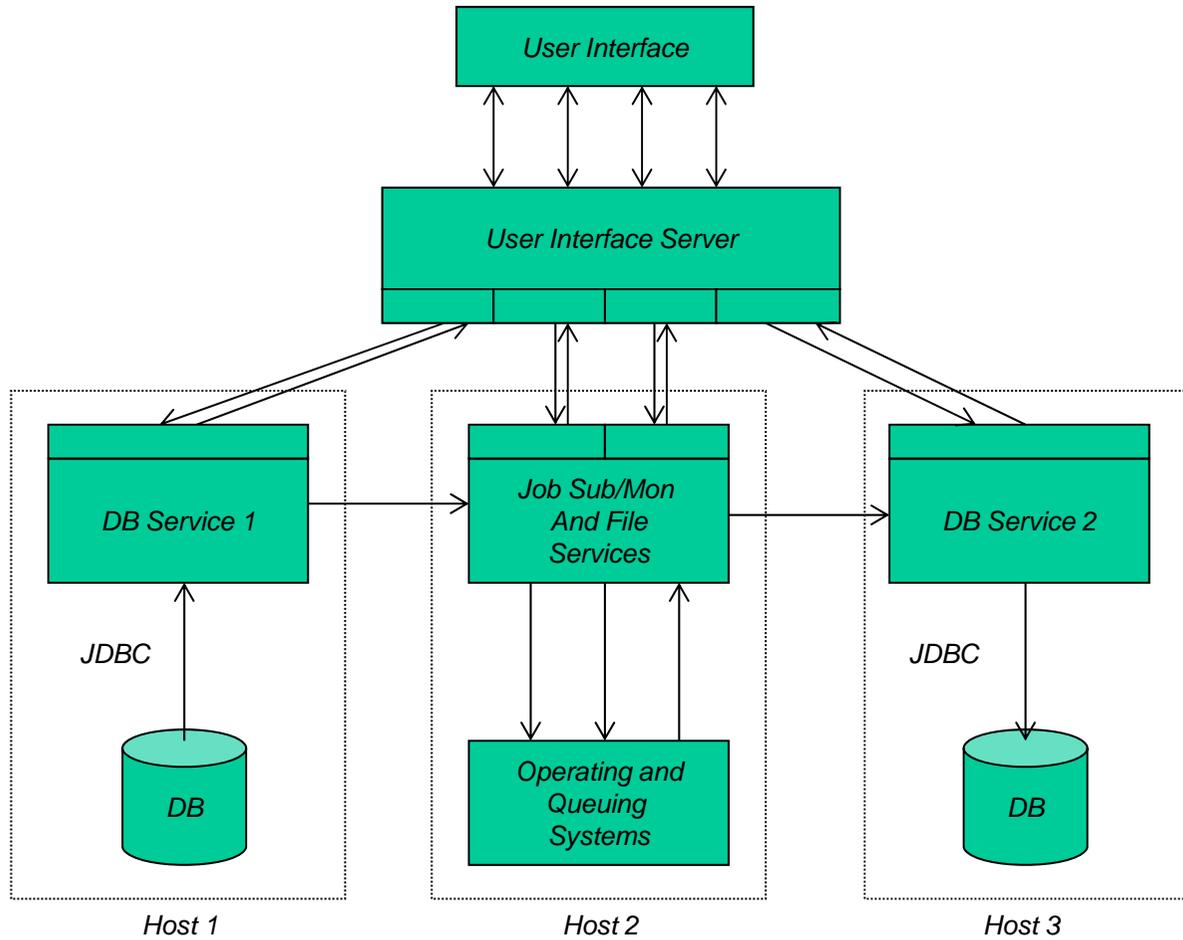
```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<movies xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://knopfler.gris.des.fi.udc.es/fb
  ellas/Movies.xsd">
  <!-- La Maldición del Escorpión de Jade. -->
  <movie>
    <identifíer>3</identifíer>
    <title>La Maldición del Escorpión de Jade</title>
    <runtime>103</runtime>
    <releaseDate day="19" month="10" year="2001"/>
    <director>Woody Allen</director>
    <actor>Woody Allen</actor>
    <actor>Helen Hunt</actor>
    <actor>Dan Aykroyd</actor>
    <actor>Elizabeth Berkley</actor>
    <actor>Charlize Theron</actor>
    <genre>COM</genre>.
    <synopsis>Woody Allen encarna a CW Briggs,el rey de los...</synopsis>
  </movie>.
</movies>
```

Web Services: Protocolos

- Hay un convenio generalizado que nos da a entender que los Servicios Web se invocan en Internet por medio de protocolos estándar basados en XML.
- Existen dos grandes tendencias: XML-RPC y SOAP, estandarizadas por el consorcio W3C.
- Pueden atravesar firewalls y routers, que piensan que es una comunicación HTTP
- XML-RPC está diseñado para ser sencillo.
- SOAP está creado con idea de dar un soporte completo y minucioso de todo tipo de servicios web.
- SOAP incluye una infraestructura a su alrededor. No es un mero protocolo de comunicación entre ordenadores, sino que además se rodea de términos como WSDL y UDDI.
 - WSDL - un método estándar de descripción de servicios web y sus capacidades específicas (XML)
 - UDDI - define las reglas basadas en XML para la creación de directorios en el que las empresas se anuncian y sus servicios web

Web Services: Arquitectura





SOAP (*Simple Object Access Protocol*)

- Permite la comunicación entre sistemas distribuidos
- Un mensaje SOAP tiene tres partes:
 - sobre - envuelve todo el mensaje y contiene la cabecera y el cuerpo
 - header - elemento opcional con información adicional, como la seguridad o encaminamiento
 - Body - específico de la aplicación y de los datos que se intercambien

```

<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd=http://www.w3.org/2001/XMLSchema
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:execLocalCommand
      soapenv:encodingStyle
        ="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1
        ="http://.../GCWS/services/Submitjob/GCWS/services/Submitjob">
      <in0 xsi:type="xsd:string">/usr/bin/csh /tmp/job.script</in0>
    </ns1:execLocalCommand>
  </soapenv:Body>
</soapenv:Envelope>

```

soapenv:Envelope

```
xmlns:soapenv=http://schemas.xmlsoap.org/soap/envelope/  
xmlns:xsd=http://www.w3.org/2001/XMLSchema  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

<soapenv:Body>

<ns1:execLocalCommandResponse

```
soapenv:encodingStyle=
```

```
http://schemas.xmlsoap.org/soap/encoding/
```

```
xmlns:ns1="http://../services/Submitjob">
```

```
<execLocalCommandReturn xsi:type="soapenc:Array">
```

```
soapenc:arrayType="xsd:string[2]"
```

```
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
```

```
<item></item> <item></item>
```

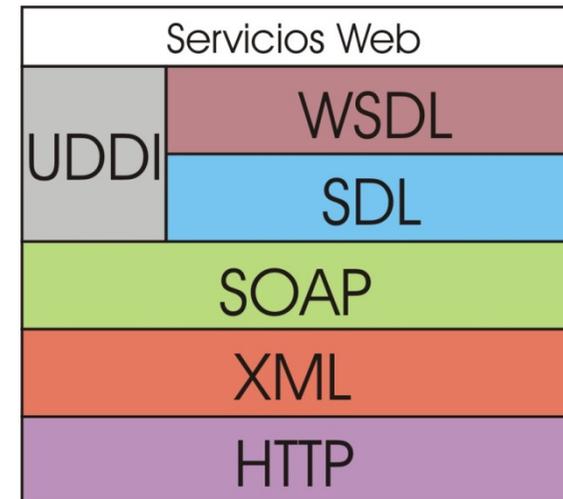
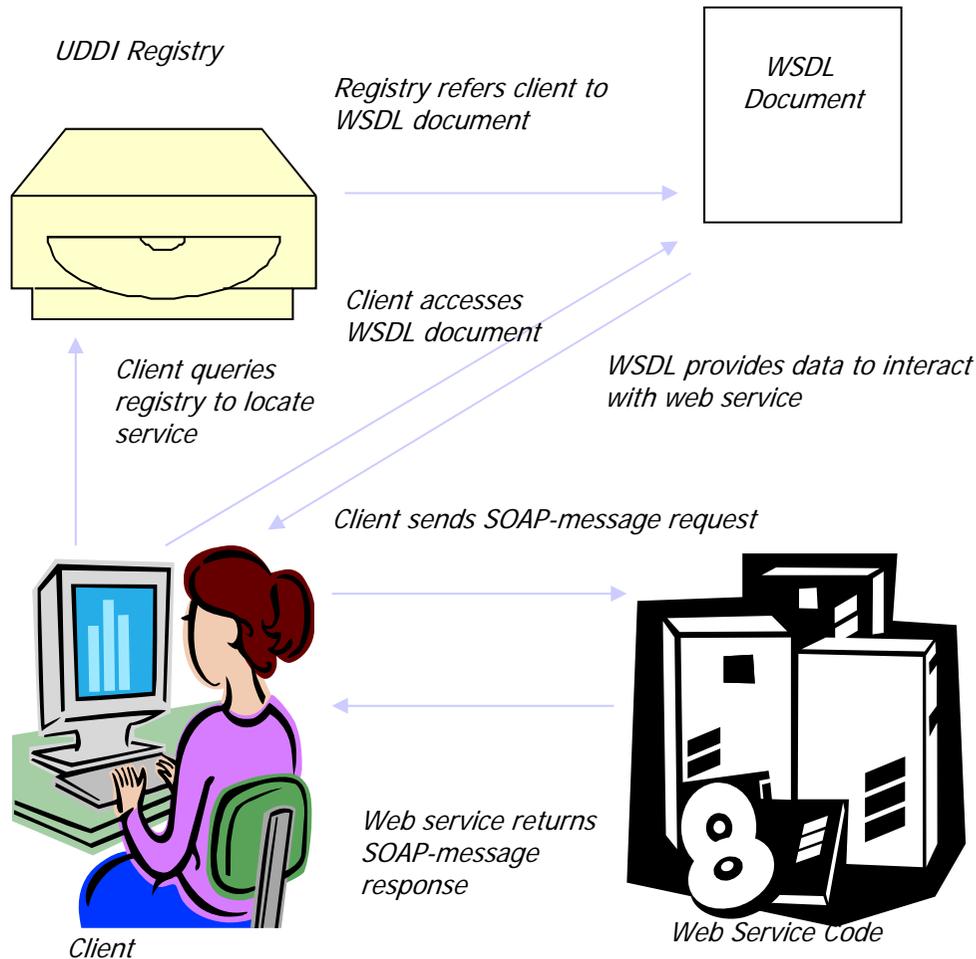
```
</execLocalCommandReturn>
```

```
</ns1:execLocalCommandResponse>
```

```
</soapenv:Body>
```

```
</soapenv:Envelope>
```

WSDL + UDDI



Web services: Implementación

- Dentro de las tecnologías de computación distribuida:
 - RMI (Java)
 - CORBA
 - DCOM (Microsoft)
 - Web Services

- Los servicios web pueden ser desarrollados con:
 - PHP, PYTHON, PERL, JSP, RUBY, TCL, C++,...
 - Visual Studio.NET (la gran apuesta de Microsoft).
 - APACHE Axis

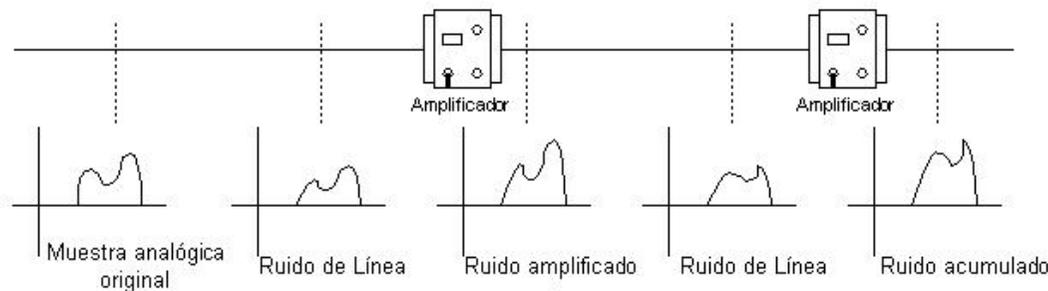
Los “nuevos” servicios Web

VoIP + IPTV + P2P

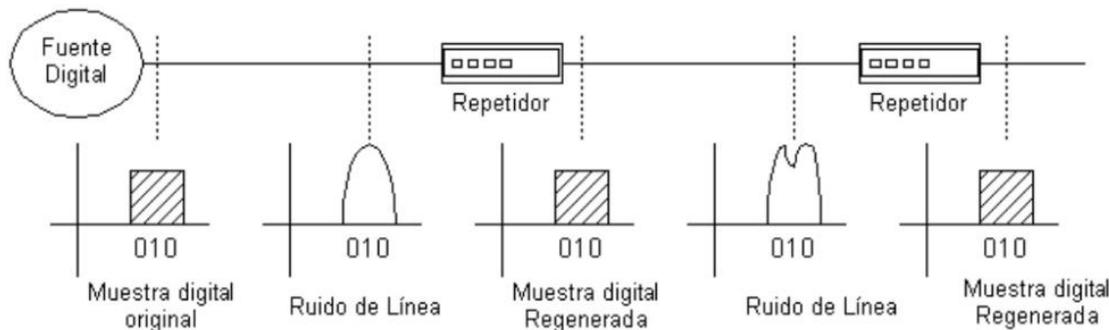
Antecedentes de VoIP: PSTN

Señalización Analógica / Digital

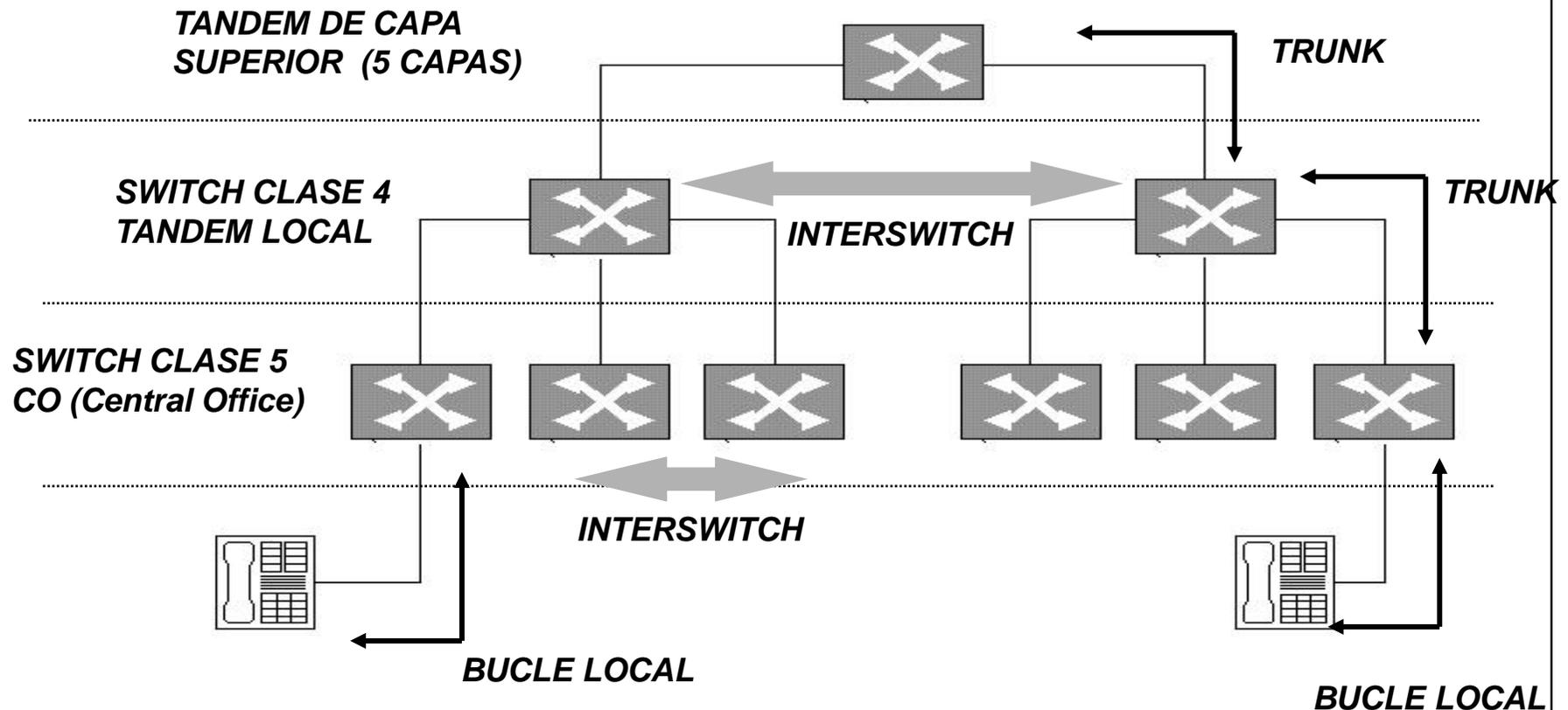
- **Señal Analógica (Voz) ⇒ Ruido de Línea**



- **Señal Digital ⇒ Regen. “Libre de errores”**



Jerarquía PSTN



PSTN: Señalización y planes de numeración

USUARIO A RED

Dentro de Banda  DTMF (Marcación Multifrecuencia)

Fuera de Banda  RDSI

RED A RED

Dentro de Banda  MF (Marcación Multifrecuencia)
RBS (Robbed Bit Signalling)

Fuera de Banda  SS7 / C7

PLANES DE NUMERACIÓN

EEUU: NANP (NA Numbering Plan)

NPA-NXX-XXX

- NPA: Numbering Plan Area
- NXX: Central Office Code
- XXXX: Station Number

ITUT: Rec. E-164

CC (3)-NDC (15)-SN (15)

- CC: Código de País
- NDC: Cód. Destino Nacional
- SN: Número de Abonado

PSTN: Inconvenientes

- *El tráfico de datos supera a la voz en redes diseñadas para voz*

Ancho de Banda elevado y variable

FUTURO: Voz sobre redes para Datos

- *El desarrollo de nuevos servicios y aplicaciones es rígido y limitado*

- *No hay soporte para la mejora de calidad*

- *Cada llamada de voz acapara 64 kbps*

SOLUCIÓN: Tratar la Voz como datos
NETWORKING DE VOZ BASADA EN PAQUETES

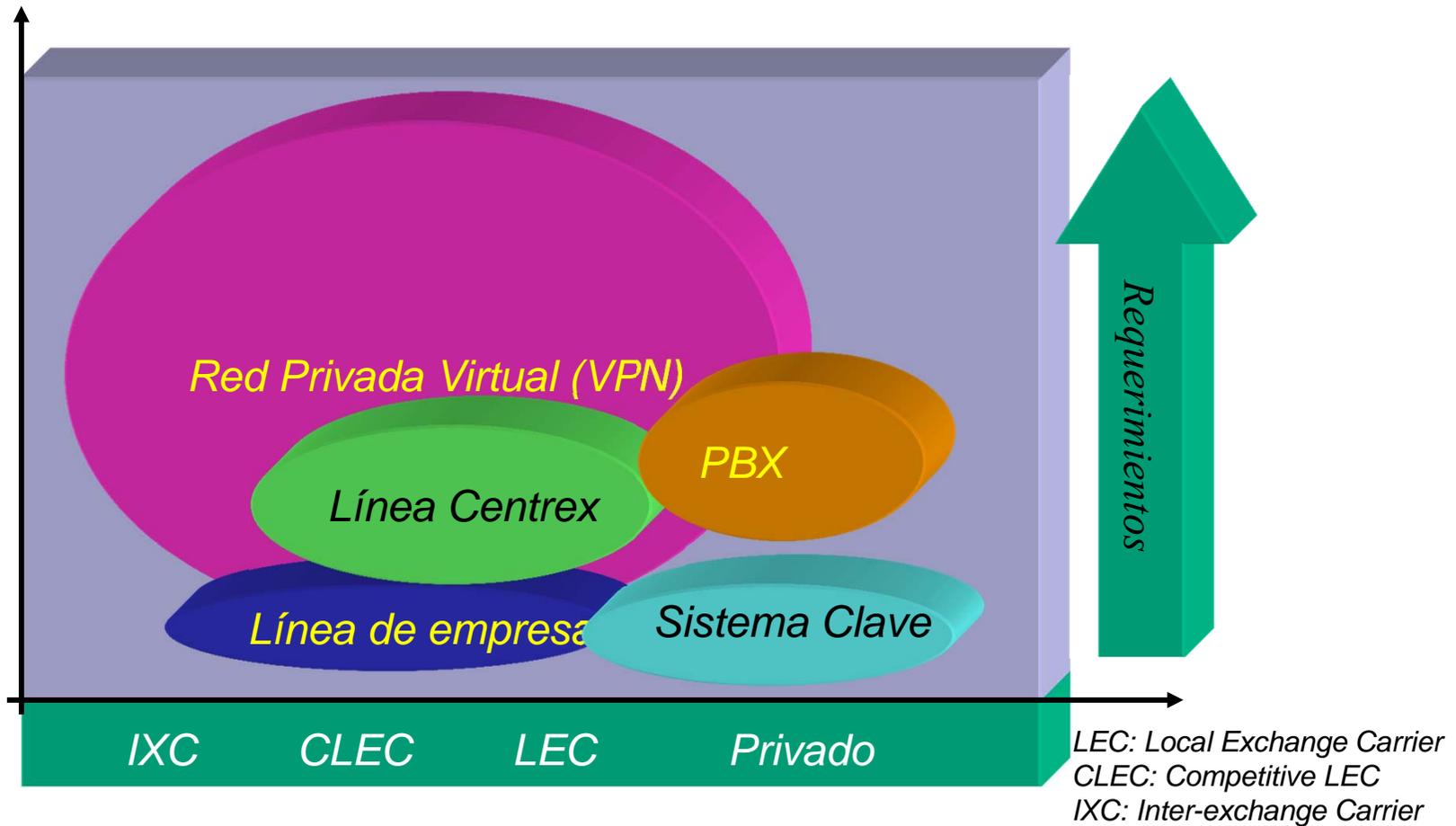
Enterprise Telephony (ET)

- *Conmutación de circuitos a 64kbps*
- *Modelo de infraestructura común*
- *Bucle local*
- *Servicios ofrecidos*
- *Sistemas de señalización dependientes de los fabricantes:*
 - ✓ *CAS (Señalización asociada al Canal)*
 - ✓ *PRI*
 - ✓ *QSIG*
 - ✓ *DPNSS (Dig. Private Network Sign. System)*
- *Funciones avanzadas:*
 - ✓ *Centros de llamadas (inbound/outbound)*
 - ✓ *Telefonía de empresa financiera (hoot-n-holler)*

Escaleta de conmutación desde
5 líneas
(PBX Private Branch Exchange)

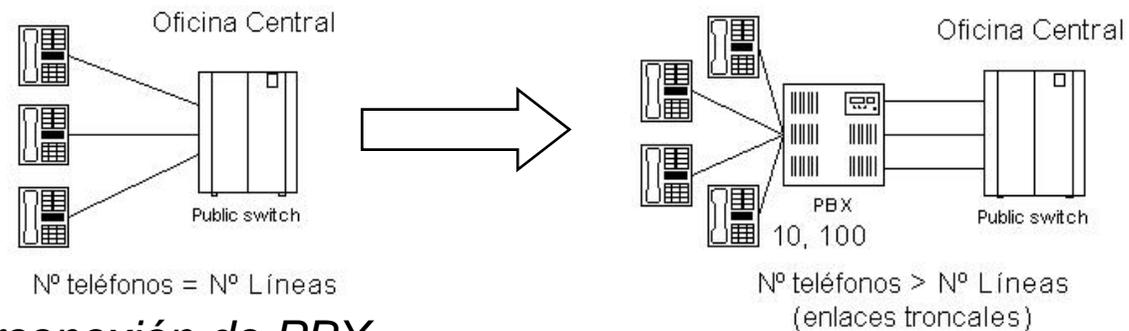
Similar a PSTN

Interworking ET/PSTN

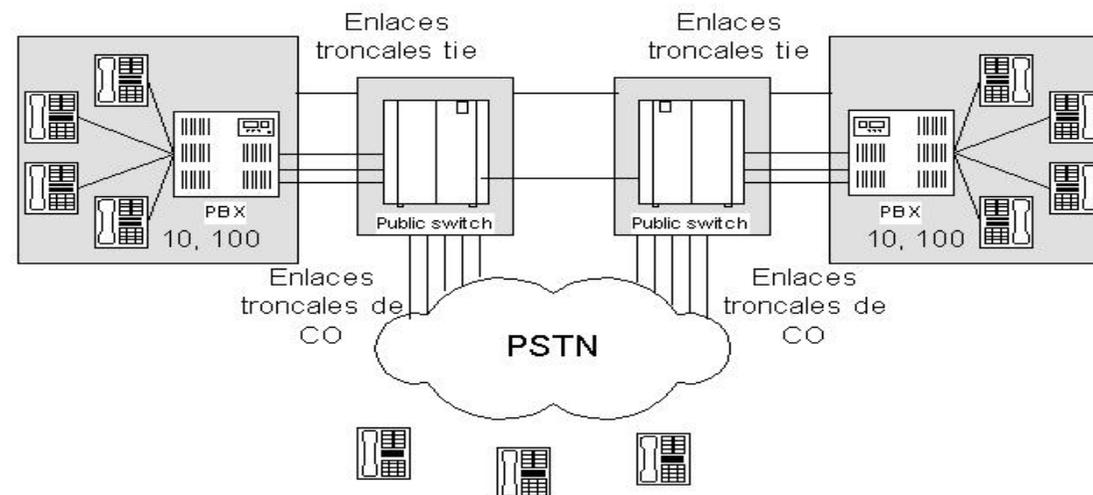


Redes ET Privadas

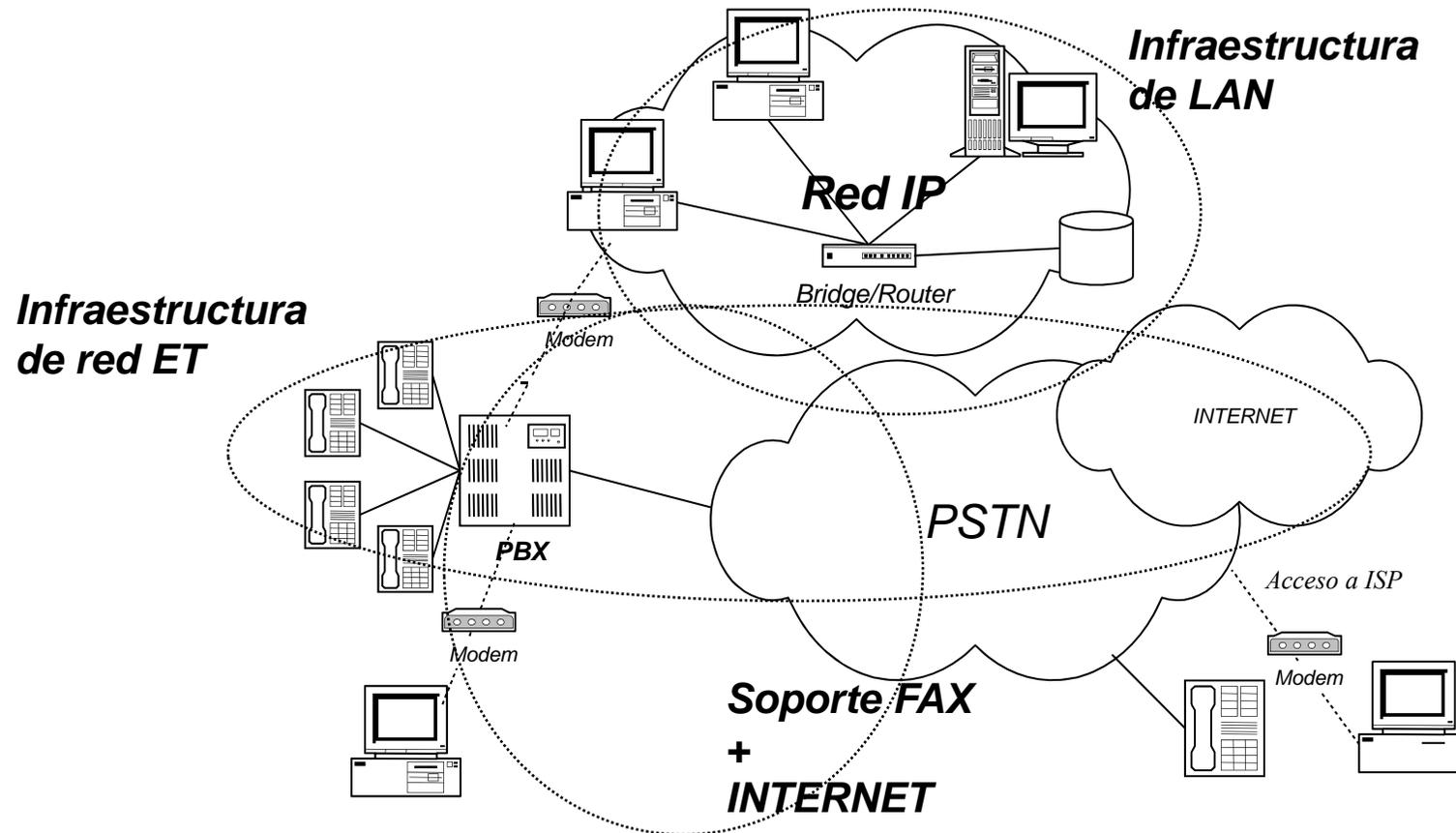
- *Redes PBX*



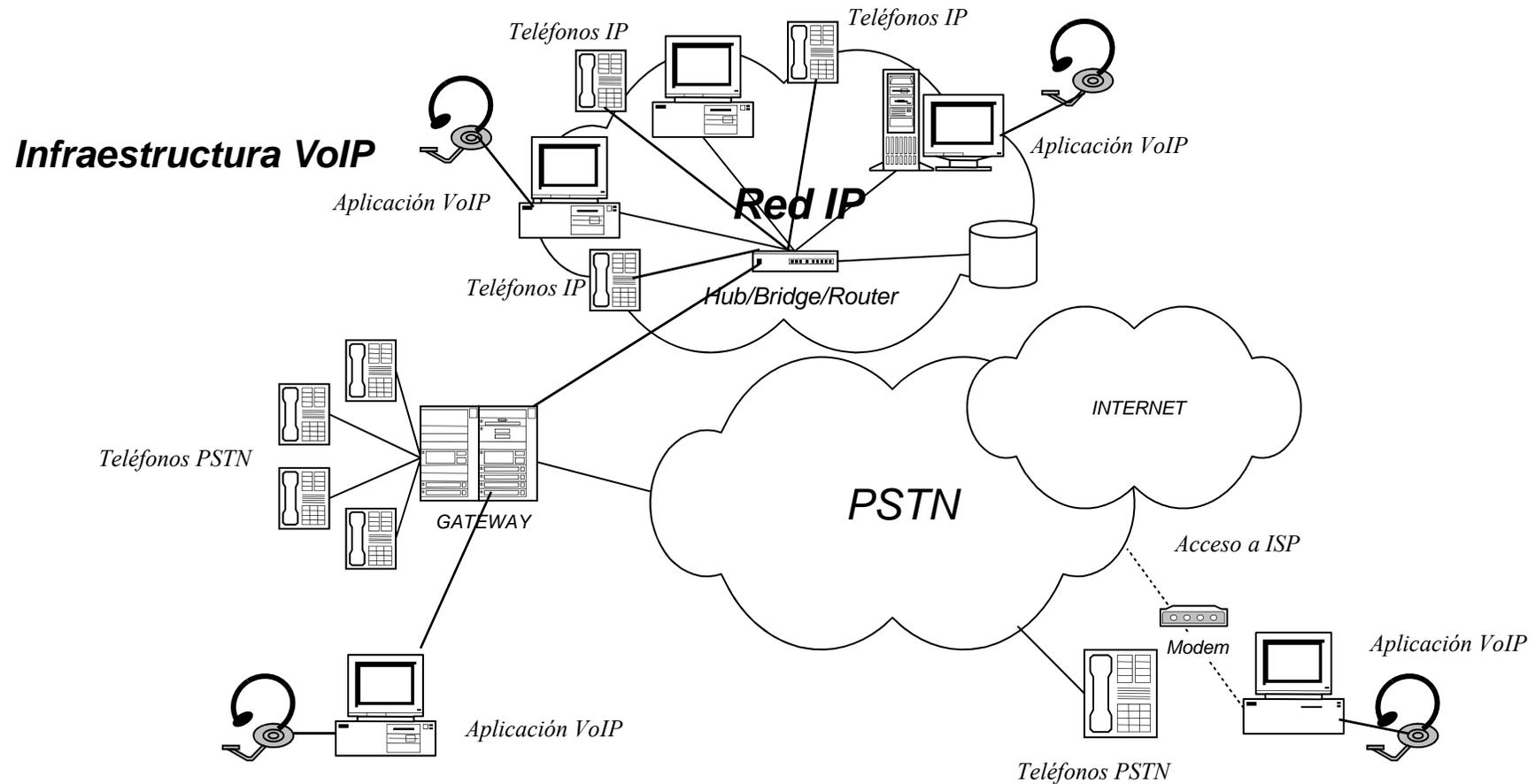
- *Líneas "tie" para interconexión de PBX*



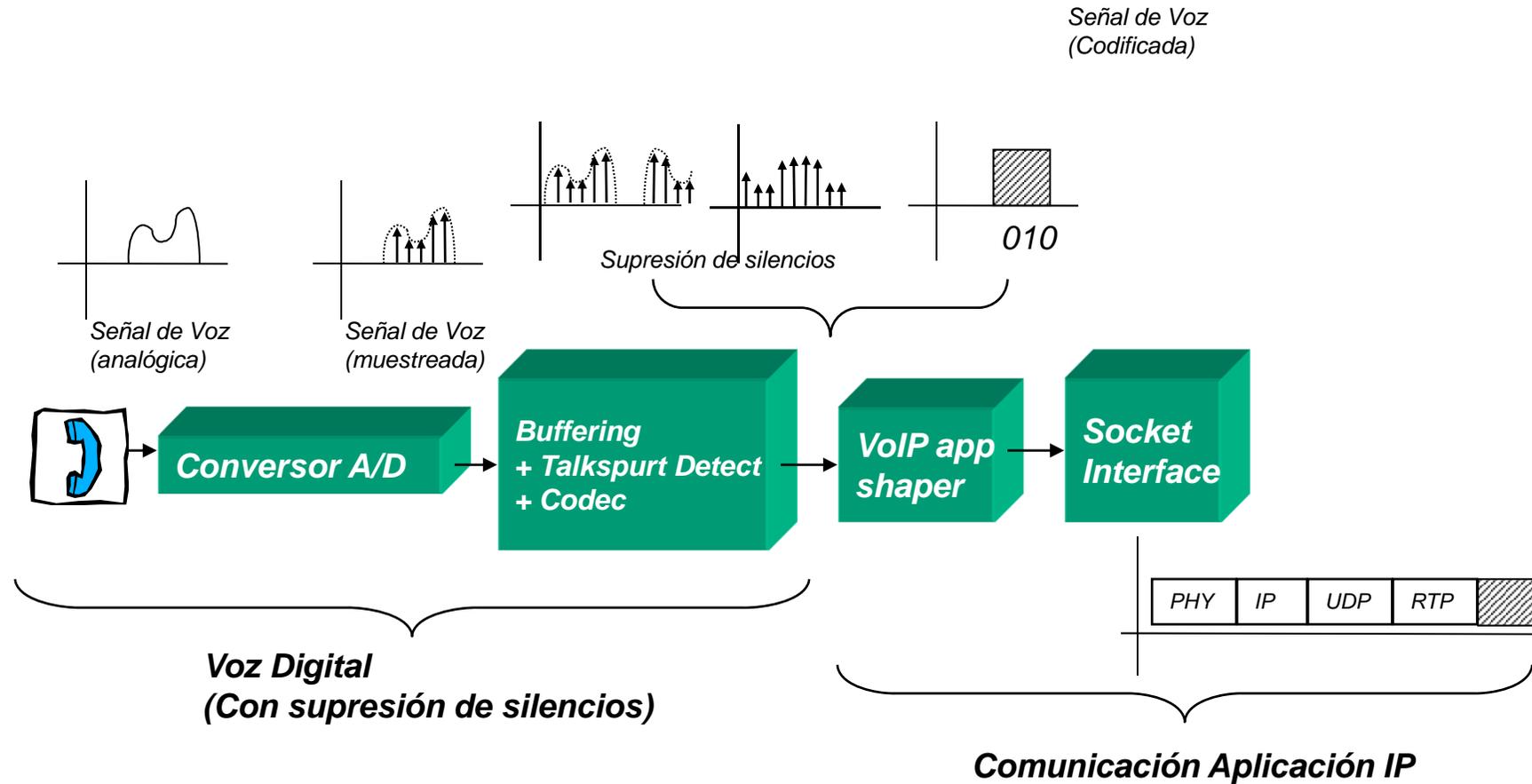
Redes de empresa: situación inicial



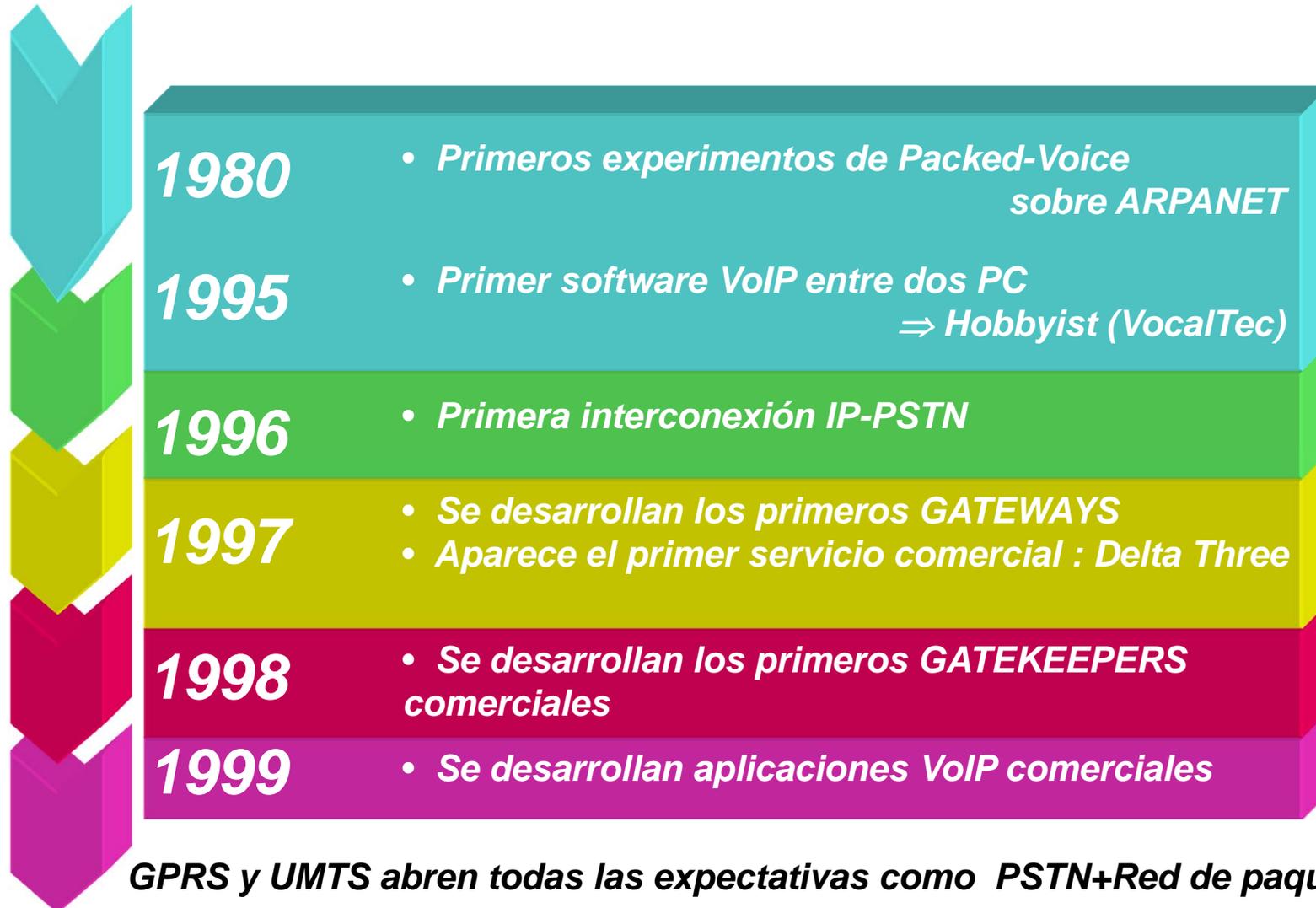
Red de empresa: Situación final



VoIP: Fundamentos



VoIP: Evolución



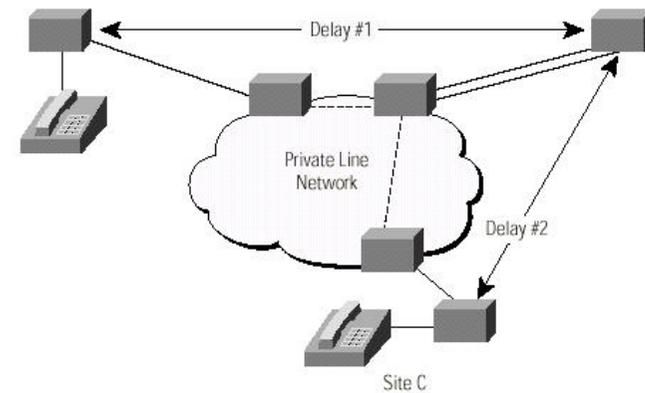
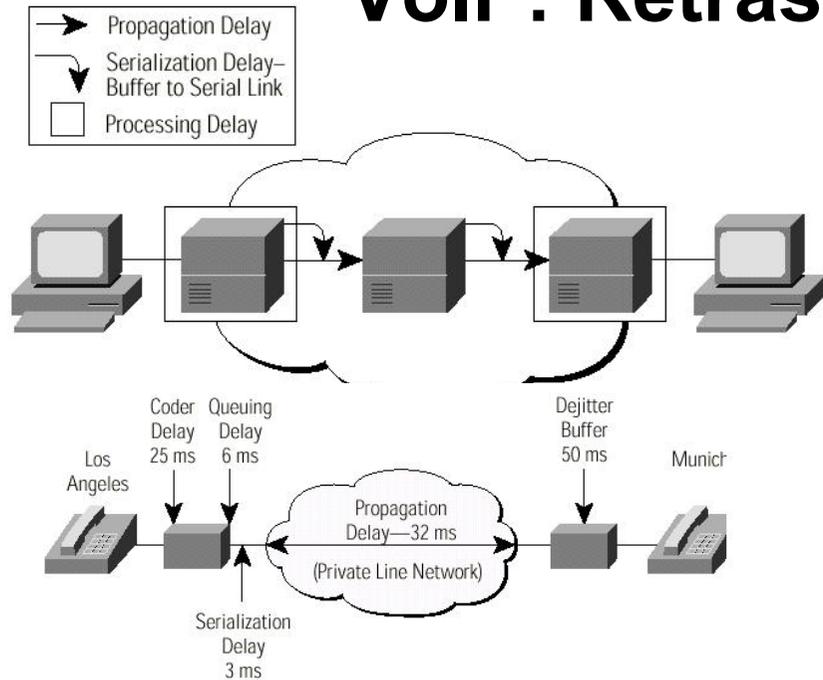
VoIP vs. PSTN

	<i>PSTN</i>	<i>VoIP</i>
<i>Conmutación</i>	Estática, basada en numeración geográfica	Dinámica, independiente de la localización
<i>Inteligencia</i>	Centralizada a través de las CO	Distribuida entre los distintos elementos de interconexión
<i>Ancho de Banda</i>	Fijo para cada comunicación de voz	Variable, función de la calidad y disponibilidad de la red
<i>Calidad</i>	Controlada	Dependiente de la red. Requiere de métodos adicionales (Ejm.:RSVP)
<i>Compatibilidad</i>	PSTN básica: Totalmente compatible ET: Dependiente de los fabricantes	A nivel de aplicación no hay estándares únicos definidos
<i>Interconexión</i>	Depende de los fabricantes (solo en el caso de ET)	Existen estándares abiertos, dependientes de los fabricantes

VoIP: La importancia de los Parámetros de Servicio

- *Retraso / Latencia*
- *Jitter*
- *Muestreo digital*
- *Compresión de voz*
- *Eco*
- *Pérdida de paquetes*
- *Detección de actividad de voz*
- *Conversión Digital / Analógico*
- *Codificación Tandem*

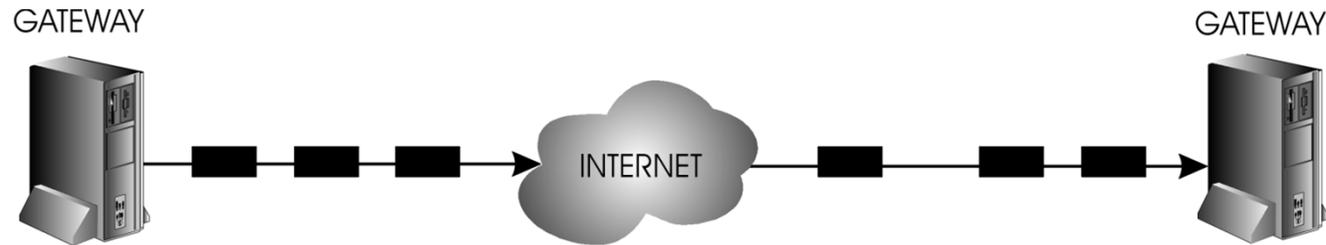
VoIP: Retraso y Latencia



	Fixed Delay	Variable Delay
Coder Delay G.729 (5 ms Look-ahead)	5 ms	
Coder Delay G.729 (10 ms per frame)	20 ms	
Packetization Delay—Included in Coder Delay		
Queuing Delay 64 kbps Trunk		6 ms
Serialization Delay 64 kbps Trunk	3 ms	
Propagation Delay (Private Lines)	32 ms	
Network Delay (e.g., Public Frame Relay Svc)		
Dejitter Buffer	50 ms	
Total	110 ms	

	Fixed Delay	Variable Delay
DELAY #1 Total	110 ms	
DELAY #2		
Coder Delay G.729	25 ms	
Packetization Delay (Included in Coder Delay)		
Queuing Delay 2 Mbps Trunk		0.2 ms
Serialization Delay 2 Mbps Trunk	0.1 ms	
Propagation Delay (Private Lines)	5 ms	
Dejitter Buffer	50 ms	
Delay #2 Total	80 ms	
Total Delay	190 ms	

VoIP: Jitter



- *En emisión el intervalo entre paquetes es fijo*
- *En recepción el intervalo entre paquetes es variable*
- *El decodificador requiere recuperar los intervalos originales*

Solución: *BUFFER en Recepción*



Incrementa la Latencia

VoIP: Compresión y codificación de voz

Type	Model	Rate (kbps)	Algorithmic Delay (ms)	Comments
G.711	Pulse code modulation (PCM) of voice frequencies	64, 56 or 48	0	Uncompressed 64 kbps channel
G.722	SB-ADPCM (Sub-Band Adaptive Differential Pulse Code Modulation)	64, 56 or 48	0.125	
G.723	ADPCM (Adaptive Delta Pulse Code Modulation)	24 or 40	0.125	
G.726	ADPCM (Adaptive Delta Pulse Code Modulation)	16, 24, 32 or 40	0.125	
G.728	D-CELP (Low-Delay Code-Excited Linear Prediction)	16	2.5	
G.729	CS-ACELP (Conjugate-Structure Code-Excited Linear-Prediction)	8	10	Quality similar to 32 kbps ADPCM. Default for VoFR.
G.729A	CS-ACELP (Conjugate-Structure Algebraic-Code-Excited Linear-Prediction)	8	15	Reduced complexity version of the G.729 codec.
G.723.1	Multi Rate Coder	5.3 or 6.3	37.5	MP-MLQ for the high rate coder ACELP for the low rate coder

Eco / Pérdida de paquetes / Detección de Actividad de Voz

- **Cancelación de eco**

- ✓ *Cancelación Software (codecs de velocidad baja)*
- ✓ *Cancelación en el DSP (echo tails – 16, 24 y 32 ms)*

- **Pérdida de paquetes**

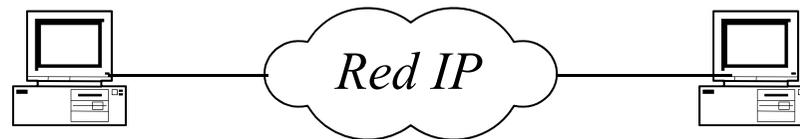
- ✓ *Estrategias de Ocultación: repiten el ultimo paquete*
- ✓ *Máxima pérdida \leq longitud del buffer en recepción (Jitter)*
Ejm.: G.729 tolera el 5% de pérdidas

- **VAD**

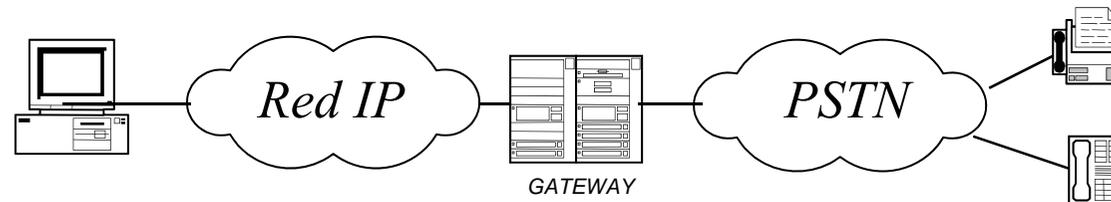
- ✓ *Se considera silencio después de 200 ms (hangover)*
- ✓ *En entornos ruidosos se limita el “umbral de señal a ruido*
- ✓ *El recorte de voz frontal no es apreciable (front-end speech clipping)*

VoIP: Escenarios de Interconexión

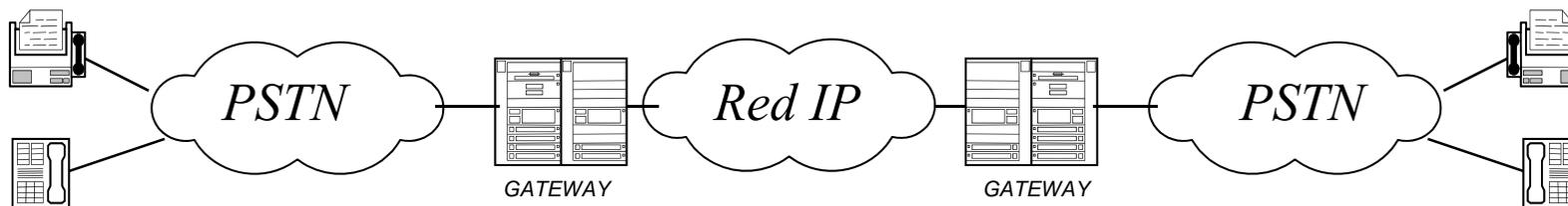
- **Terminal IP - terminal IP**



- **Terminal IP – Teléfono/Fax**



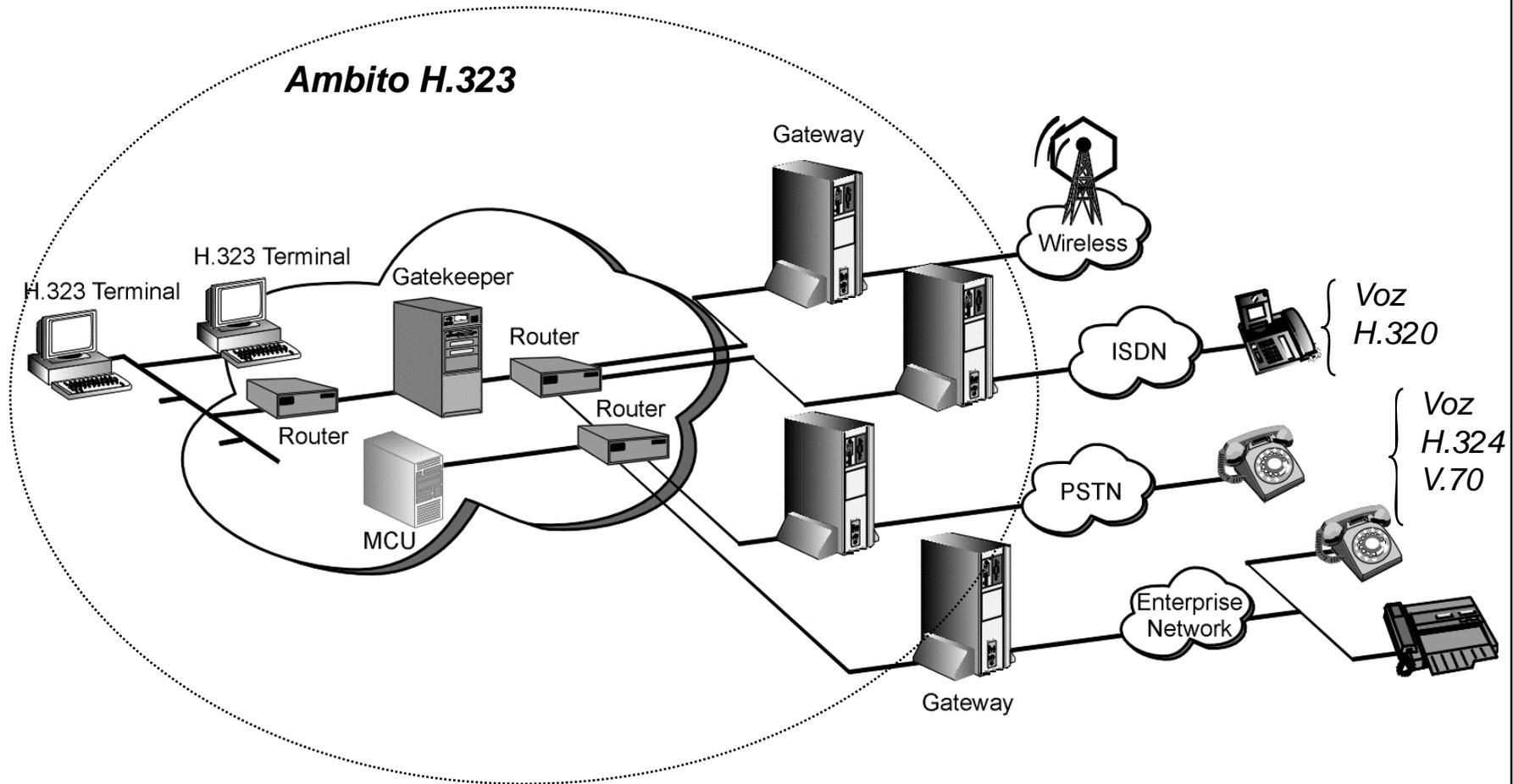
- **Teléfono/Fax – Teléfono/Fax**



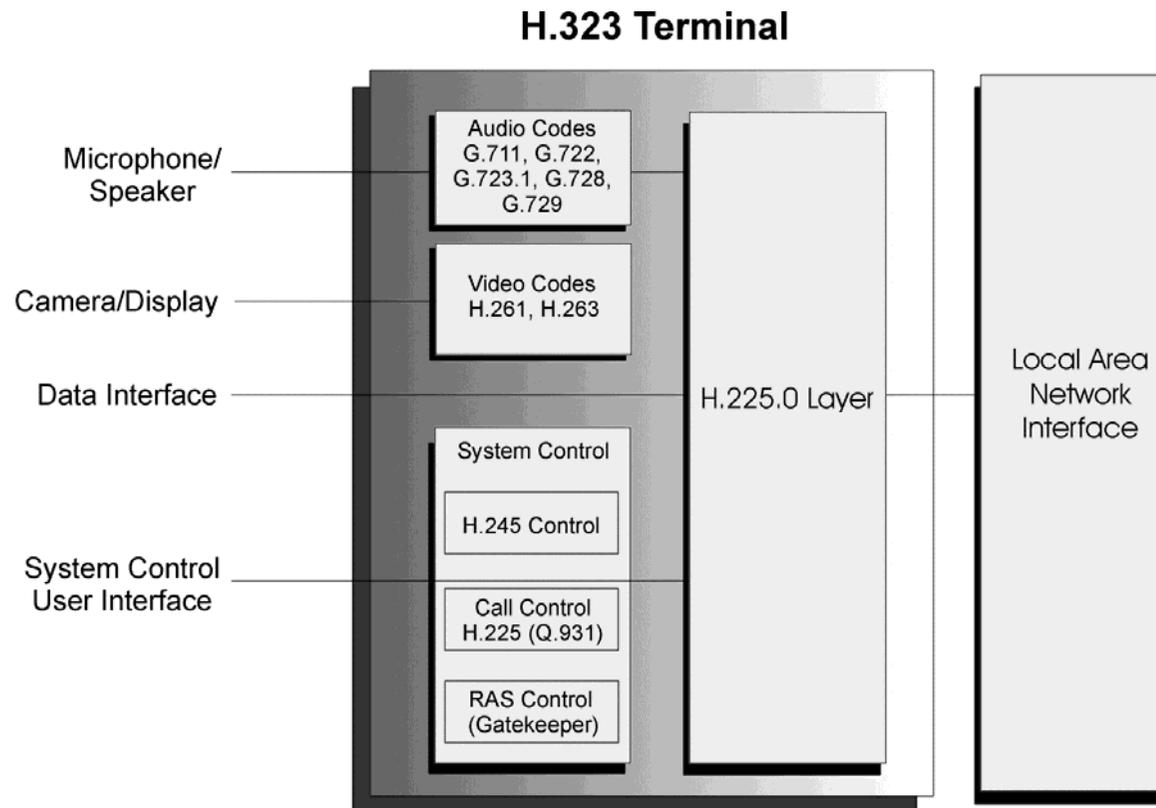
VoIP: Arquitecturas

- H.323
 - Es el estándar de señalización para transmisión de voz sobre IP
 - Es considerado el marco de referencia de VoIP
- SIP (Protocolo de Inicio de Sesión)
 - Es el estándar IETF
 - Puede coexistir con H.323

H.323: Escenario



H.323: Terminales



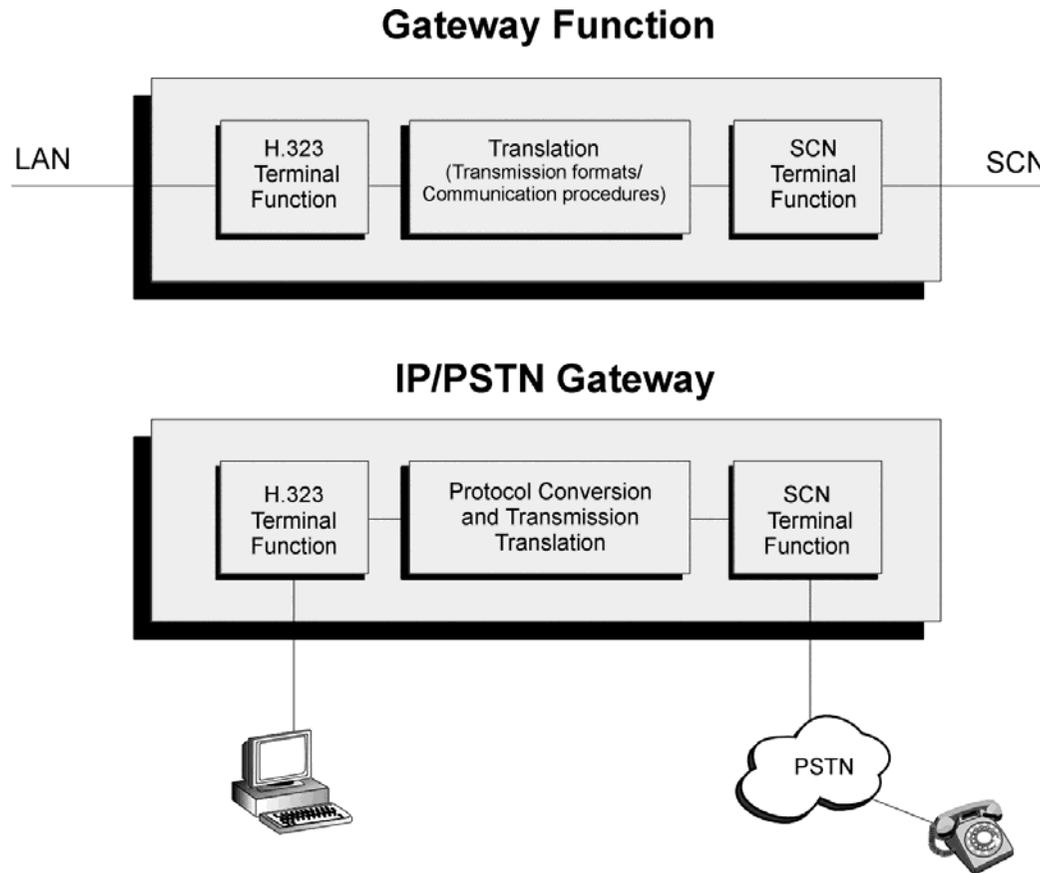
- **Teléfonos IP**

- ✓ Cisco 30 VIP
- ✓ Selsius IP EthPhone
- ✓ Nokia IPCourier

- **Software Phones**

- ✓ VocalTec IPhone
- ✓ Netscape's CoolTalk
- ✓ WhitePine's CU-SeeMe
- ✓ Microsoft NetMeeting

H.323: Gateways



• Funciones

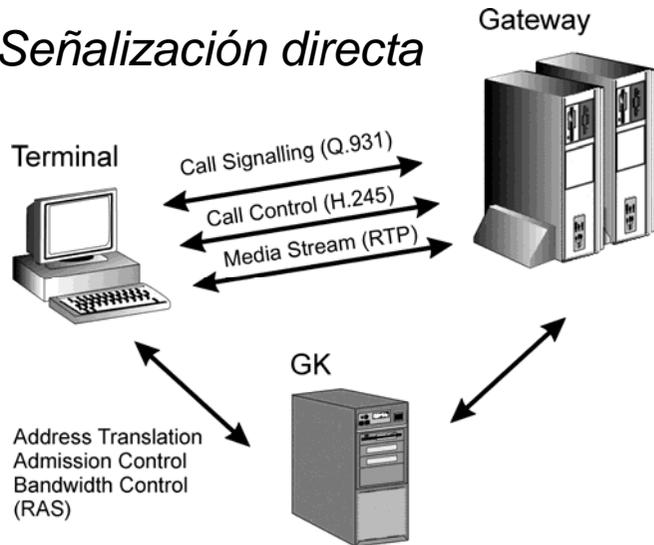
- ✓ Búsqueda
- ✓ Conexión
- ✓ Digitalización
- ✓ Demodulación
- ✓ Compresión
- ✓ Decompresión

• Productos

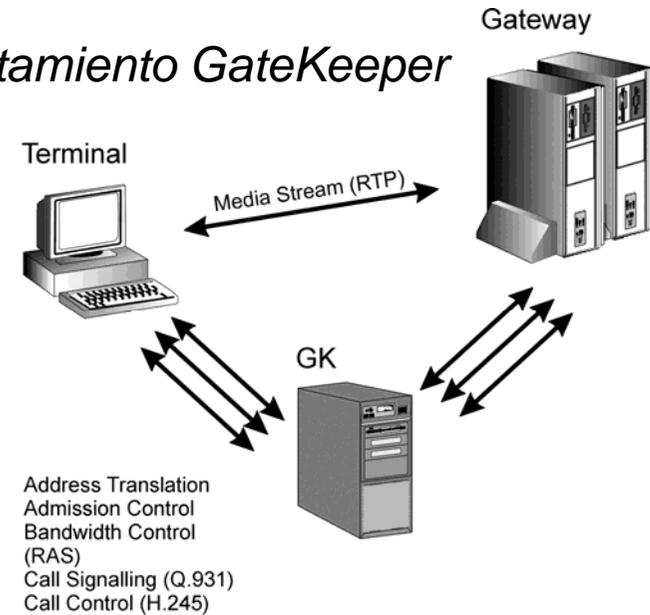
- ✓ MICOM V/IP Gateway
- ✓ Nortel Networks CVX SST GW
- ✓ Lucent Pathstar Access Server
- ✓ CISCO DE-30+ GW
- ✓ 3Com GW
- ✓ VocalTec Series 2000 GW
- ✓ Nuera Sol. Access plus F200 IP

H.323: Gatekeepers

- *Señalización directa*



- *Enrutamiento GateKeeper*



- **Funciones**

- ✓ Traducción de direcciones
- ✓ Control de admisión
- ✓ Señalización de llamada
- ✓ Autorización
- ✓ Gestión de Ancho de Banda
- ✓ Gestión de llamadas

- **Productos**

- ❖ Ericsson H323 GK
- ❖ VocalTec GK
- ❖ Nortel Networks IPConnect
- ❖ Elemedia GK2000S

H.323: Multipoint Control Unit (MCU)

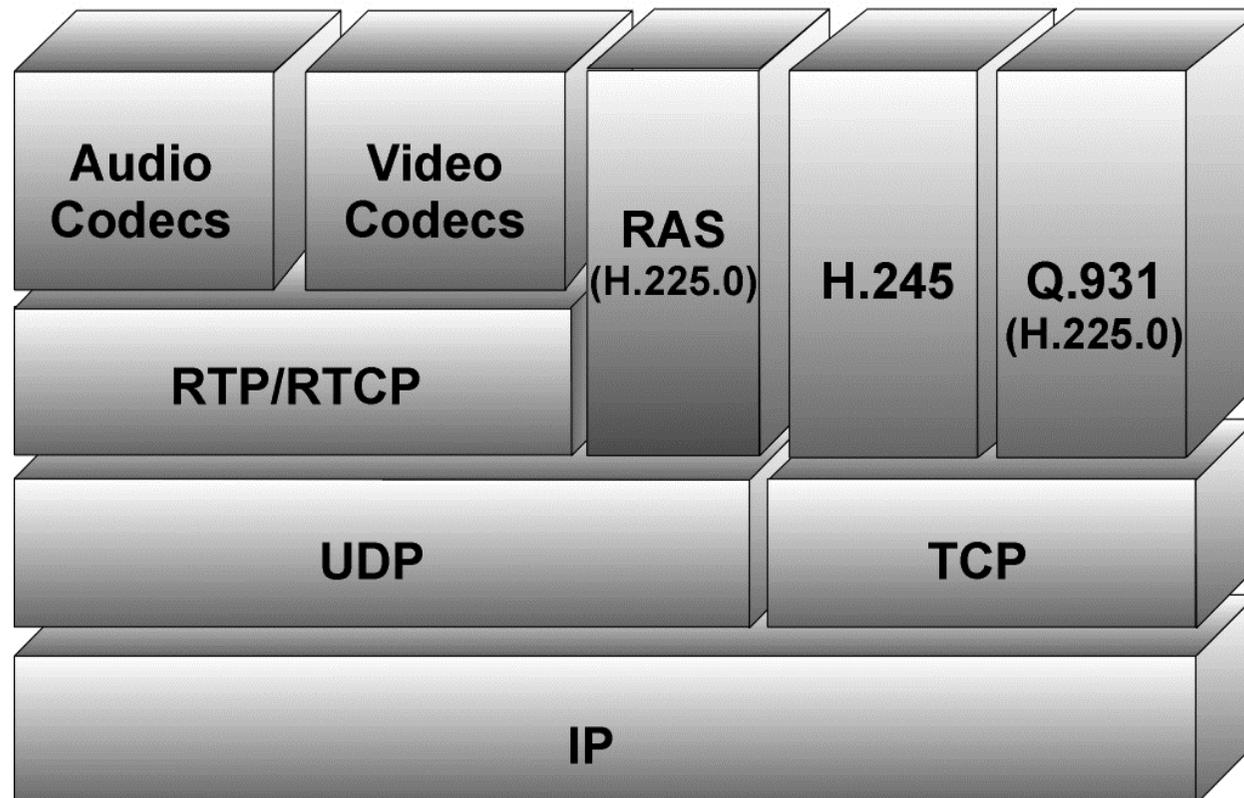
- *Implementa funcionalidades exclusivas para Multiconferencia*
- *Dos módulos:*
 - ✓ **Multipoint Controller (MC):** *Gestiona los mensajes de control y señalización.*
 - ✓ **Multipoint Processor (MP):** *Acepta streams, realizando su distribución.*
- *Una MCU normalmente soporta módulos MC*
- *Los extremos implementan los MP*
- *Una MCU dorsal presenta tanto MC como MP*

H.323: Proxi

- *Elemento de Capa de Aplicación, con acceso a los paquetes intercambiados por dos aplicaciones H.323*
- *Funciones*
 - ✓ **Negociación RSVP ó Precedencia IP (DiffServ)**
 - ✓ **Enrutamiento de tráfico H.323 : Application Specific Routing (ASR)**
 - ✓ **Direccionamiento privado**
 - ✓ **En entornos sin Firewall implementa seguridad H.323**
 - ✓ **En entornos seguros actúa como “nodo de confianza”**

H.323

Protocol Stack



H.323: Control y Señalización

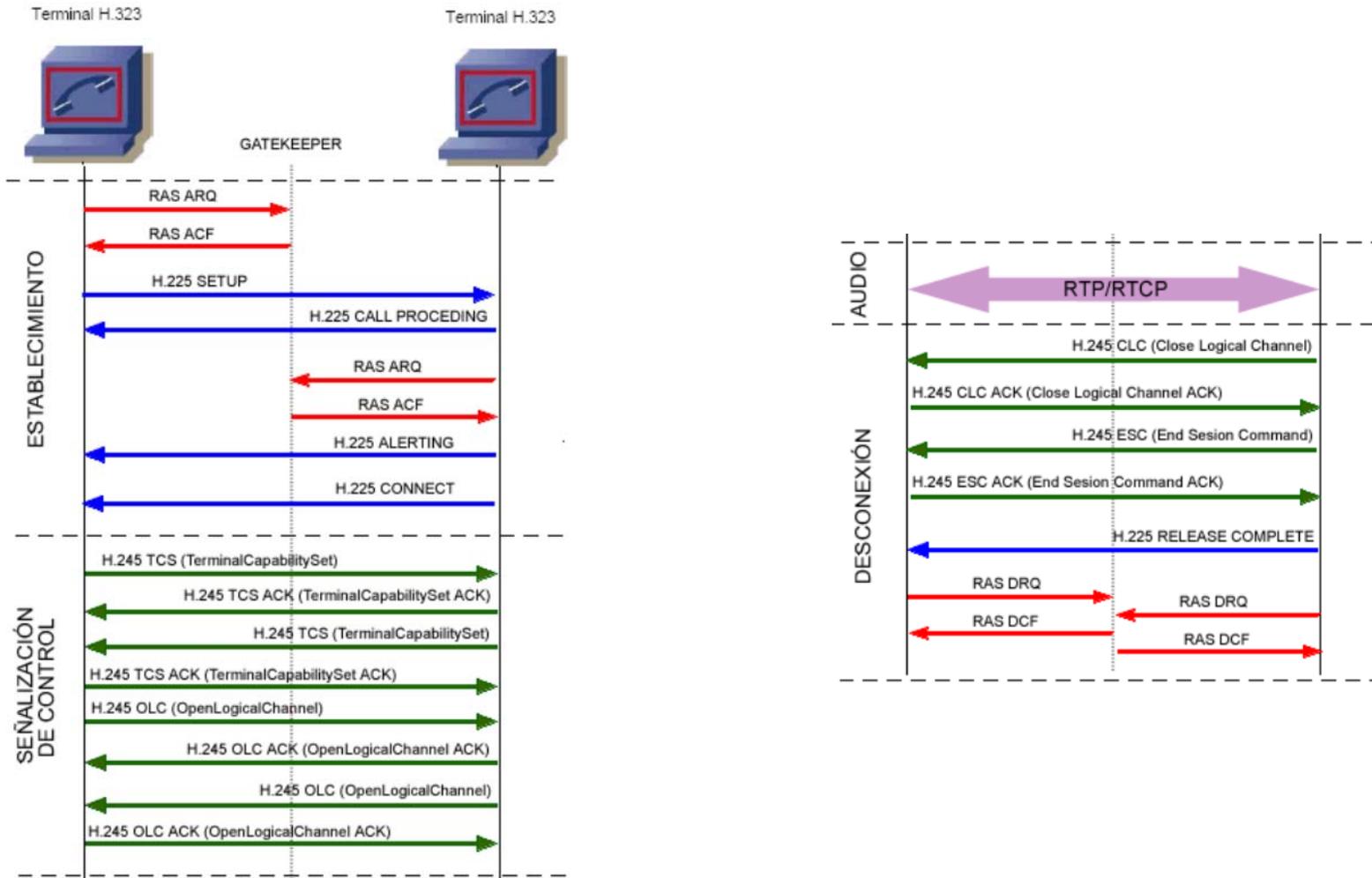
- *H.225.0 RAS*
 - *Comunicación entre puntos finales y el GateKeeper sobre UDP*
 - *Procedimientos*
 - ✓ *Descubrimiento del GateKeeper (GRQ, GCF, GRJ)*
 - ✓ *Registro del punto final (RRQ, RCF, RRJ, URQ, UCF)*
 - ✓ *Localización de punto final (LRQ, LCF, LRJ)*
 - ✓ *Admisión, Modificación del ancho de banda, Estado y Desenganche*

- *H.225.0/Q.931 Call Signalling*
 - *Canal de control de llamadas seguro (puerto 1720 TCP)*
 - *Q.931 : Mensajes de control de llamadas (conexión, mantenimiento y desconexión)*
 - *Q.932 : Servicios suplementarios*
 - *Enrutamiento del canal de señalización:*
 - ✓ *Señalización de llamada directa de punto final (Direct Endpoint Call Signalling)*
 - ✓ *Señalización de llamada de gatekeeper enrutado (Gatekeeper Routed Call Sig.)*

H.323: Control de medios (H.245)

- *Mensajes de control extremo a extremo entre entidades H.323*
- *Canales lógicos (audio, vídeo, datos y control)*
- *Procedimientos*
 - ✓ *Asignación de master-slave: busca el MC como control central*
 - ✓ *Intercambio de capacidades*
 - ✓ *Control del canal de medios*
 - ✓ *Control de conferencia*
- *Enrutamiento del canal (con GKRCs):*
 - ✓ **Direct H.245 Control:** *Directo entre puntos finales*
 - ✓ **Gatekeeper Routed H.245 Control:** *entre cada punto final y su gatekeeper*
- *Modo Tunnelling H.245:*
 - ✓ *Se pueden encapsular varios mensajes H.245 dentro de un mensaje H.225*

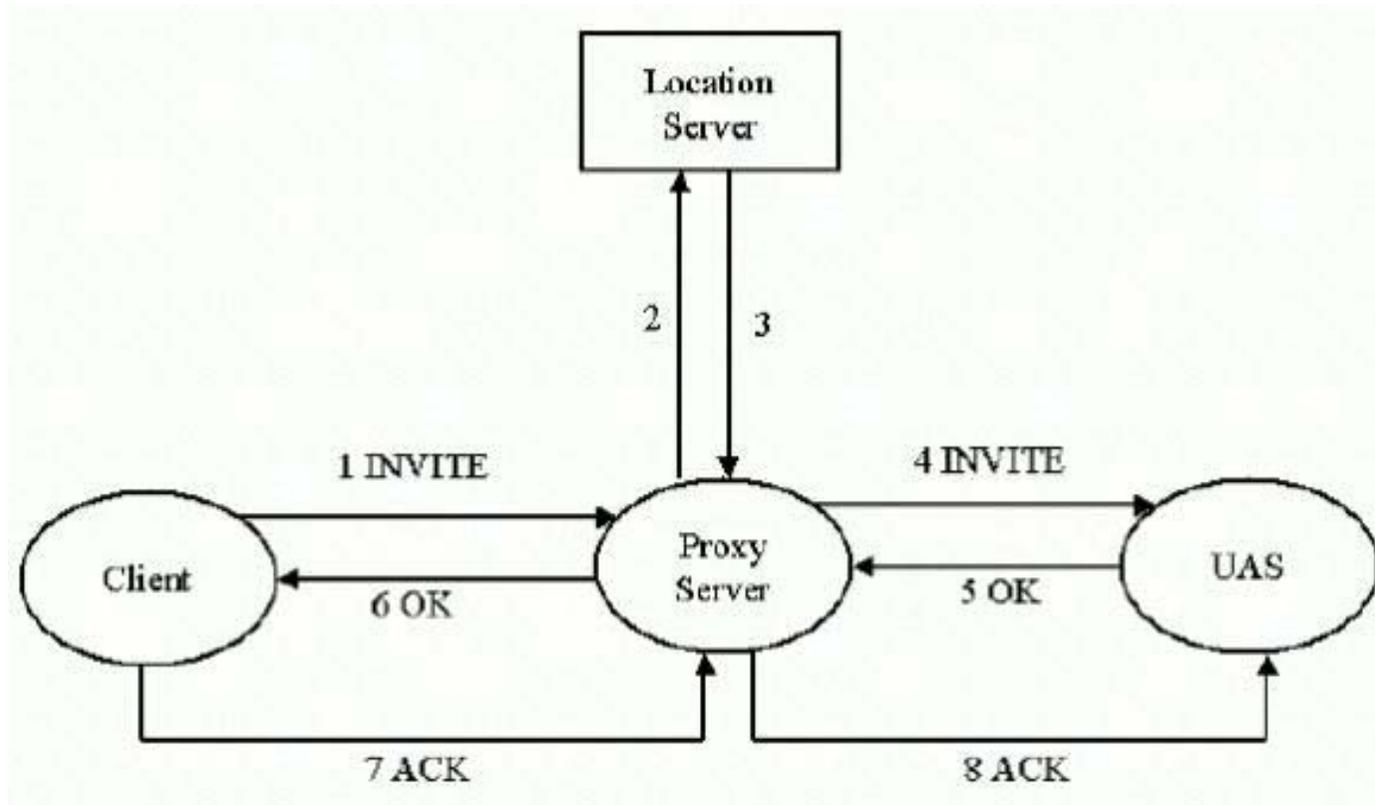
H.323: Ejemplo de sesión



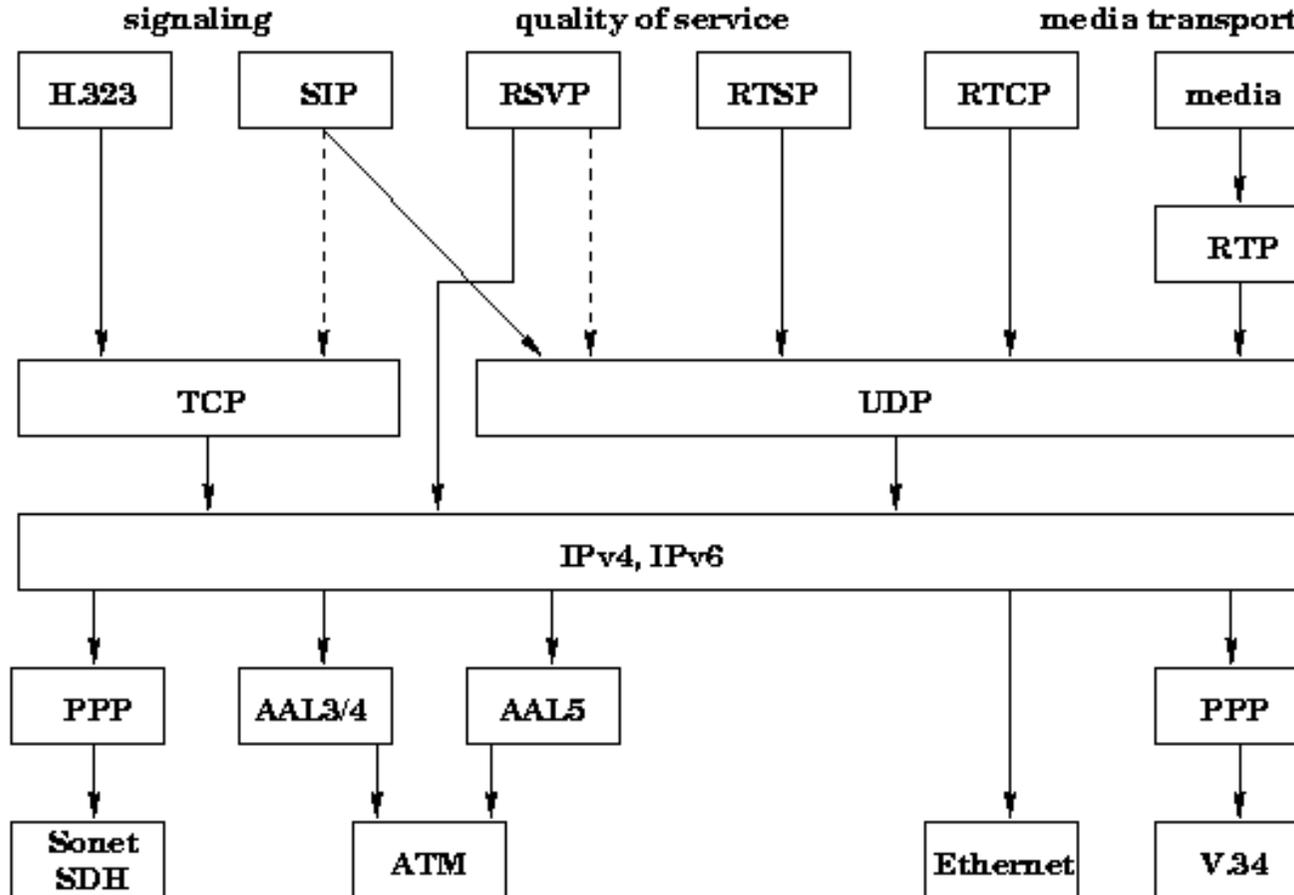
SIP: Arquitectura de red

- **Terminales (Agentes de usuario)**
 - ✓ *Aplicaciones cliente del sistema final*
 - ✓ *Cliente Usuario-Agente (UAC)*
 - ✓ *Servidor Usuario-Agente (UAS)*
- **Servidores Proxi**
 - ✓ *Subconjunto de un GateKeeper H.323*
 - ✓ *No existen zonas de influencia*
 - ✓ *Realiza la traducción de direcciones*
- **Servidores de redirección**
 - ✓ *Acepta peticiones SIP*
 - ✓ *Redirecciona al siguiente servidor SIP*
 - ✓ *No acepta llamadas ni procesan peticiones*

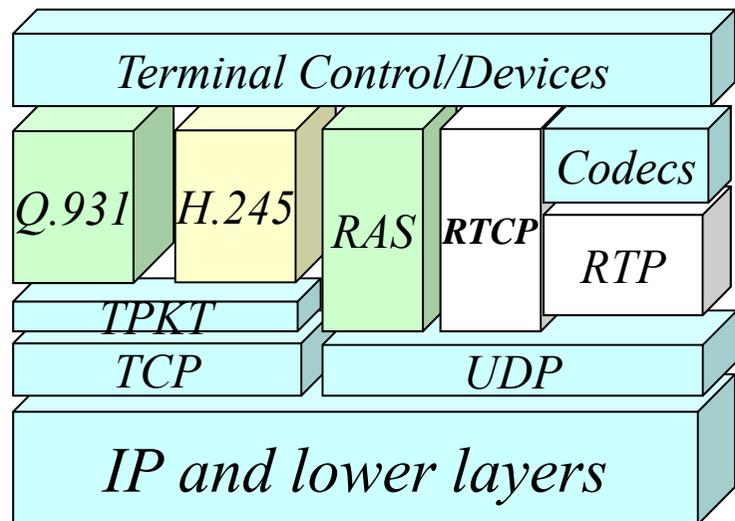
SIP: Interacción de elementos



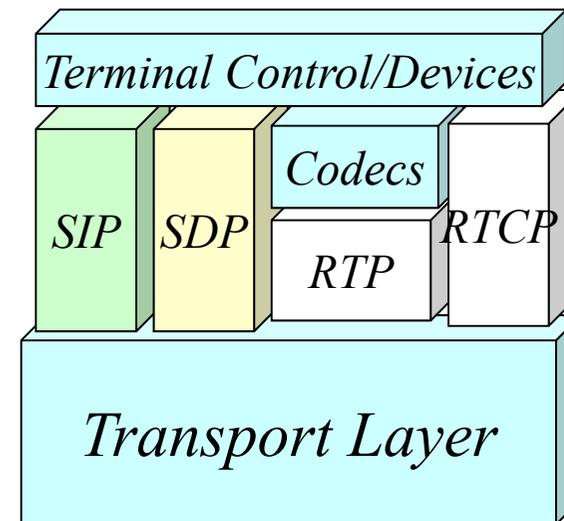
SIP: Arquitectura de protocolos para servicios de transferencia de datos multimedia



H.323 vs. SIP



H.323



SIP

H.323 vs. SIP

	H.323	SIP
Complejidad	Protocolo Complejo	Comparativamente simple
Representación	Binaria (ASN.1)	Textual (ISO 10646)
Compatibilidad	Total a todos los niveles	No es imprescindible
Modularidad	Limitada	Alta
Escalabilidad	Limitada	Muy alta
Señalización	Compleja	Simple
Respaldo	Fabricantes	IETF
Dimensión	Cientos de elementos	Solo 37 cabeceras
Fiabilidad	Difícil detección de bucles	Fácil detección de bucles

SIP: Direcciones

- SIP Uniform Resource Indicators (URIs)
 - Mismo formato que las direcciones de correo: user@domain
 - Dos esquemas URI:
 - sip:henry@siptest.wcom.com es un “SIP URI”
 - ↗ Formato habitual descrito en RFC 2543
 - sips:henry@siptest.wcom.com es un “Secure SIP URI”
 - ↗ Nuevo esquema descrito en RFC 3261
 - ↗ Usa TLS sobre TCP como protocolo de seguridad en capa de transporte
 - Dos tipos de “SIP URIs”:
 - Address of Record (AOR) (identifica un usuario)
 - ↗ sip:henry@wcom.com (Necesita peticiones DNS SRV para localizar usuarios SIP dentro del dominio wcom.com)
 - Fully Qualified Domain Name (FQDN) (identifica el dispositivo)
 - ↗ sip:henry@127.24.45.4 or sip:henry@cube43.lab.wcom.com
- sip:usuario:contraseña@host:puerto;parámetros-uri?cabeceras

SIP: Ejemplos de direcciones

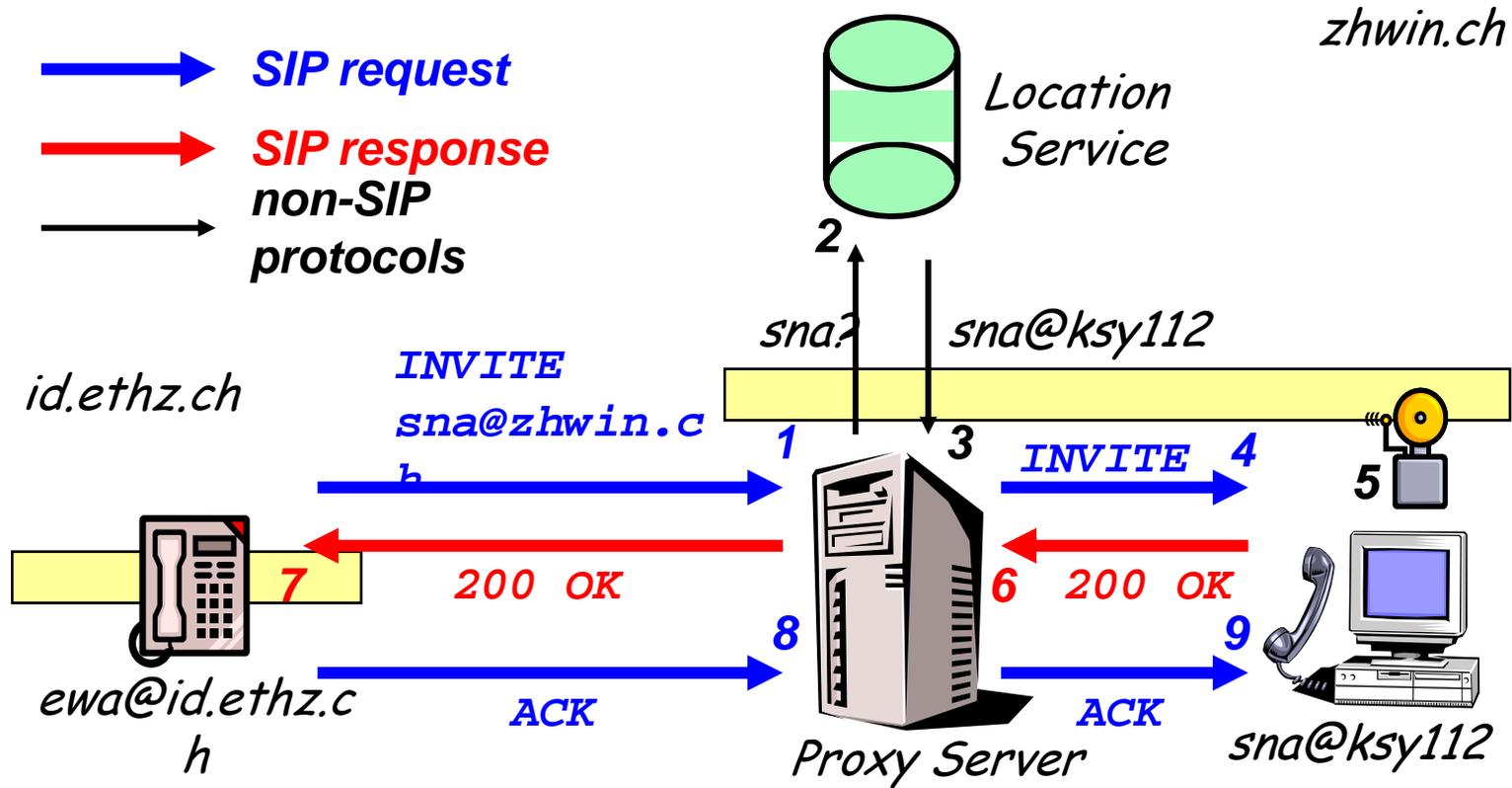
- Ejemplos de “SIP URIs”

- sip:andreas.steffen@zhwin.ch
- sip:andreas.steffen@strongsec.com;transport=tcp
- sip:sna@pcsn.zhwin.ch:3013
- sip:sna@160.85.129.35:3013
- sip:+41-76-340-2556@sipgate.diax.ch;user=phone
- sip:434@gateway.zhwin.ch;user=phone
- sip:zhwin.ch;method=REGISTER

- Por Defecto

- :5060 (destination port)
- transport=udp (transport parameter)
- user=ip (user parameter)
- method=INVITE (SIP method)

SIP: Escenario



SIP: Mensajes

- SIP define la comunicación a través de dos tipos de mensajes.
 - solicitudes (métodos) y
 - respuestas (códigos de estado)
- Formato de mensaje genérico establecido en el RFC 2822 que consiste en:
 - Línea inicial
 - Campos de cabecera (headers),
 - Línea vacía que indica el final de las cabeceras
 - Cuerpo del mensaje que es opcional.

Línea inicial
Cabecera
Línea Vacía
Cuerpo del Mensaje

SIP: Peticiones

- PETICIONES de la versión inicial:
 - INVITE: invitación del llamante
 - ACK: fin de establecimiento (respuesta del llamante a la aceptación del llamado)
 - CANCEL: cancela petición en curso
 - BYE: Abandono de un participante en una sesión múltiple
 - OPTIONS: consulta a un agente de usuario sus capacidades
 - REGISTER: registro de la localización actual de un usuario
- NORMA GENERAL: Excepto ACK todas las peticiones necesitan respuesta

SIP: Respuestas

- RESPUESTAS: códigos numéricos
- PROVISIONALES: (no terminan la transacción)
 - 1XX (Informativas): petición procesada pero no completa (trying, ringing, forwarded, queued...)
- FINALES: (terminan la transacción)
 - 2XX (Éxito): petición completada con éxito en extremo remoto, 200 =“OK”
 - 3XX (Redirección): llamante debe redirigir su petición hacia otra ubicación; “moved temporarily”, “moved permanently”, ...
 - 4XX (Fallo de petición): la petición recibida no puede cursarse. Se sugiere modificación y reintento; ”bad request”, “timeout”, “busy here”, “user not found”...
 - 5XX (Error del servidor): al procesar la información. Se sugiere reintentar; “not implemented”, “version not supported”, ...
 - 6XX (Fallo global): petición no puede ser cursada. Sin reintento; “busy everywhere”, “user does not exist anywhere, session not acceptable, ...)

SIP: Cabeceras

- Formato de encabezamientos SIP (HEADER).
 - To: Identifica al usuario destino de la solicitud.
 - From: Identifica al UAC que inicia la solicitud.
 - Call-ID: Identifica una invitación y todas las transacciones posteriores relacionadas con esa transacción.
 - Call-ID: ges456fcdw211dkfte12ax@workstation1234.university.com
 - Contact: proporciona la URL donde podemos encontrar al destinatario directamente, así descargando a los servidores intermedios que no es necesario estén en el path.
 - Contact: Bob Johnson <sip:Bob@131.160.1.112>
 - Cseq (Command Sequence): Número de secuencia que se antepone al nombre del método (INVITE, ACK..)
 - Cseq: 2 INVITE
 - Record-Route y Route: Estos dos encabezamientos son utilizados por proxies que quieren estar en el path de señalización toda la sesión: por seguridad (firewalls), provisión de servicios relacionados con la sesión...
 - Via: almacenan la identificación de los proxies por los que ha pasado una petición (p.e. para detectar loops)

SIP: Ejemplo de mensaje

(En peticiones) **Solicitud: Método; URI solicitud; Versión SIP**
 (En Respuestas) **Estado: Versión SIP; Código Estado; Frase**

INVITE sip:marconi@radio.org SIP/2.0

Línea Inicial

Campos
Obligatorios

Via: SIP/2.0/UDP lab.high-voltage.org:5060
 To: G. Marconi <sip:Marconi@radio.org>
 From: Nikola Tesla <sip:n.tesla@high-voltage.org>
 Call-ID: 123456789@lab.high-voltage.org
 CSeq: 1 INVITE
 Subject: About That Power Outage...
 Contact: sip:n.tesla@high-voltage.org
 Content-type: application/sdp
 Content-Length: 158

Cabecera

CRLF

Nombre campo:
Valor campo

v=0
 o=Tesla 2890844526 2890844526 IN IP4 lab.high-voltage.org
 s=Phone Call
 c=IN IP4 100.101.102.103
 t=0 0
 m=audio 49170 RTP/AVP 0
 a=rtpmap:0 PCMU/8000

Cuerpo del Mensaje

Campos
definidos en SDP

SIP: Ejemplo de mensajes

Request

INVITE sip:UserB@there.com SIP/2.0

Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:UserA@here.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 12345600@here.com
CSeq: 1 INVITE
Subject: Happy Christmas
Contact: BigGuy <sip:UserA@here.com>
Content-Type: application/sdp
Content-Length: 147

Message Header Fields

v=0
o=UserA 2890844526 2890844526 IN IP4 here.com
s=Session SDP
c=IN IP4 100.101.102.103
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000

Payload

Response

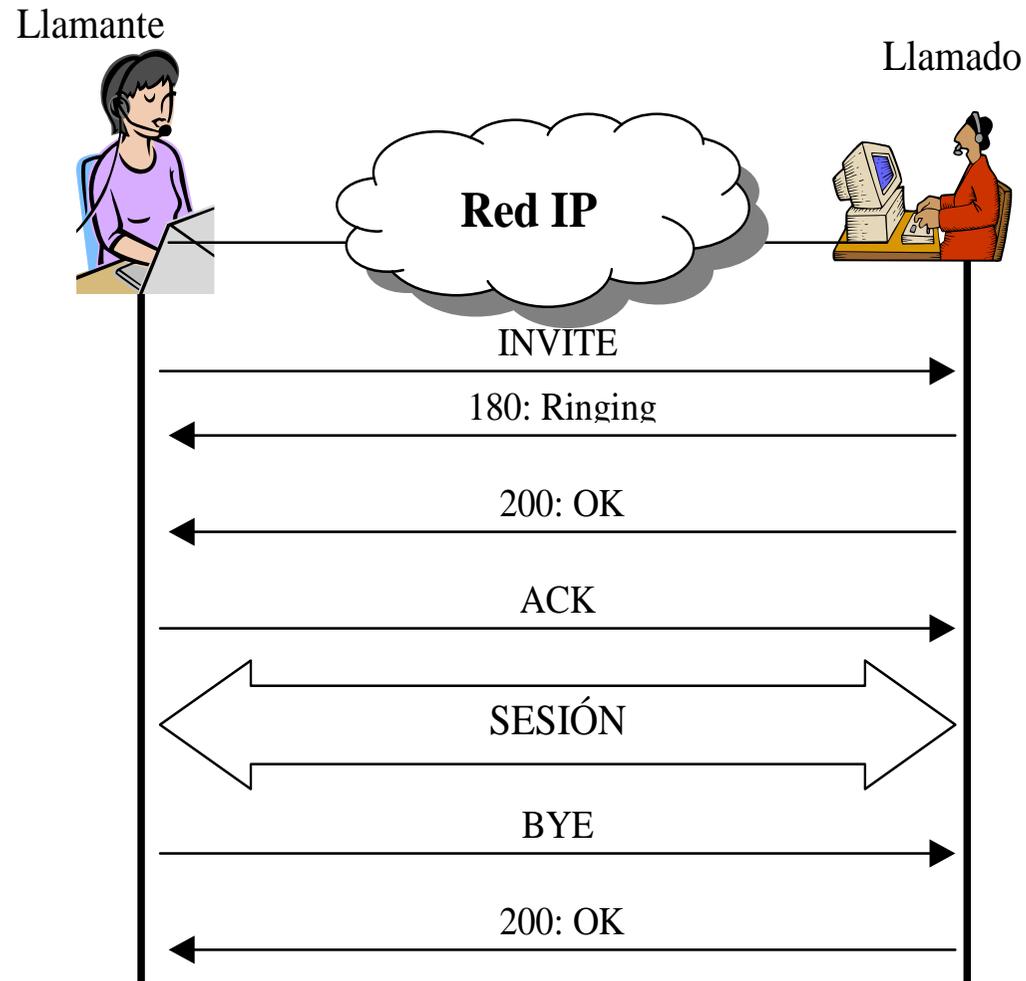
SIP/2.0 200 OK

Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:UserA@here.com>
To: LittleGuy <sip:UserB@there.com>;tag=65a35
Call-ID: 12345600@here.com
CSeq: 1 INVITE
Subject: Happy Christmas
Contact: LittleGuy <sip:UserB@there.com>
Content-Type: application/sdp
Content-Length: 134

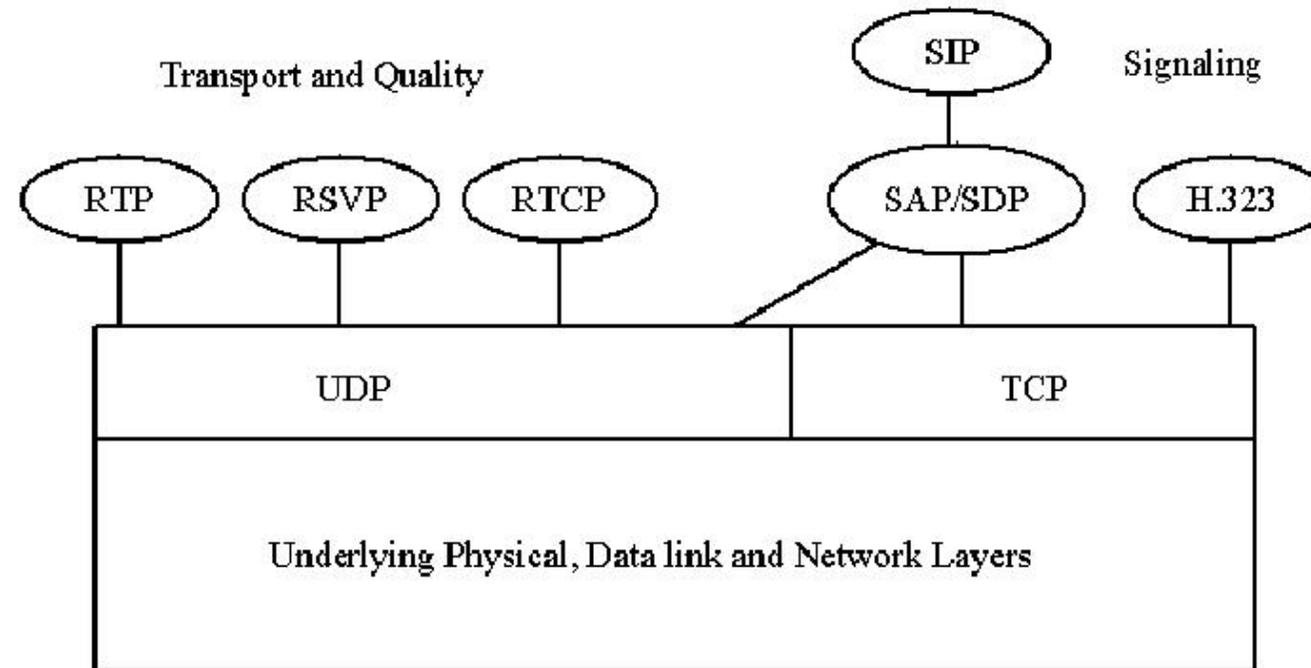
v=0
o=UserB 2890844527 2890844527 IN IP4 there.com
s=Session SDP
c=IN IP4 110.111.112.113
t=0 0
m=audio 3456 RTP/AVP 0
a=rtpmap:0 PCMU/8000

SDP "receive RTP G.711-encoded audio at 100.101.102.103:49172"

SIP: Ejemplo de sesión



VoIP: Protocolos de Soporte



- **RTP: Real Time Protocol**
- **RSVP: Resource Reservation Protocol**
- **RTCP: Real Time Control Protocol**
- **SAP: Session Announcement Protocol**
- **SDP: Session Description Protocol**
- **RTCP: Real Time Control Protocol**

- ... además a nivel de Gateways: **MGCP (Media Gateway Control Protocol)** y **SGCP (Simple Gateway Control Protocol)**

VoIP: RTP / RTCP

- **RTP**: Soporte para el transporte de datos en tiempo real

- **Funciones**

- ✓ Secuenciamiento
- ✓ Identificación del Payload
- ✓ Indicación de trama
- ✓ Identificación de la fuente (SSRC)
- ✓ Sincronización intramedia

- **RTCP**: Soporte para control intercalado en RTP

- **Funciones**

- ✓ Información de QoS
- ✓ Control de sesión
- ✓ Identificación (informativa)
- ✓ Sincronización intermedia

VoIP: RTSP/RSVP

- **RTSP:** Soporte cliente-servidor para el streaming de audio/vídeo.
- Protocolo a nivel de aplicación
- Sintaxis y operaciones similares a HTTP, utilizando URLs
 - Adoptado por múltiples fabricantes
- Funciones
 - ✓ Funcionalidades VCR
 - ✓ Extracción desde el servidor de medios
 - ✓ Invitación al servidor de medios
 - ✓ Insertar medios al servidor

- **RSVP:** Soporte para la reserva de recursos (control de prioridad y latencia)
- No es un protocolo de enrutamiento
- Soporta prot. de enrutamiento unicast y multicast (presentes y futuros)
- Funciones
 - ✓ Unicast y multicast
 - ✓ Modificación de recursos dinámica
 - ✓ Operaciones de routing suplementarias

VoIP: SDP/SAP

- **SDP:** Soporte para caracterización de sesiones multimedia.
- Formato de texto
- Permite incorporar participantes dinámicamente
- Funciones
 - ✓ Identificación de la sesión
 - ✓ Direccionamiento y puertos
 - ✓ Temporizadores
 - ✓ Ancho de Banda utilizado
 - ✓ Información de soporte

- **SAP:** Soporte para caracterización de sesiones multicast
- Suele asociarse a un Proxi
- Los grupos SAP intercambian la información de sesiones
- Funciones
 - ✓ Mantenimiento de integridad
 - ✓ Autenticación
 - ✓ Encriptado

VoIP: SGCP/MGCP

- **SGCP**: soporte para el control de conexiones entre gateways.
- Los MGC convierten TDM a Voz en paquetes
- El control de llamadas reside en los **agentes de llamadas**
- Establece, mantiene y desconecta llamadas, controlando las conexiones entre puntos finales
- Aporta mecanismos de seguridad dados por la IP Security Arch.:
 - ✓ IP Authentication Header (RFC 1826)
 - ✓ IP Encaps. Sec. Payload (RFC 1827)

- **MGCP**: Soporte para VoIP sobre elementos externos de control de llamadas
- SGCP + IPDC (IP Device Control)
- Los GW de telefonía pueden ser controlados por elementos externos
- Los GW incluyen:
 - ✓ Trunk GW (PSTN-VoIP)
 - ✓ VTOA GW (PSTN-ATM)
 - ✓ Residential GW (bucle-VoIP)
 - ✓ Business & Access GW (PBX)
 - ✓ Network Access Servers (ISP)
 - ✓ Circuit&Packet SW

VoIP: Implementaciones

- Aplicaciones en PC
- Coexistencia con RTC
 - De pago
 - De empresa
 - Broadband VoIP
 - Cable VoIP
- Next Generation Network
 - Eliminación de la conmutación de circuitos
 - Creación/Actualización de servicios
 - **Combinando la inteligencia de la red y del and CPE (edge) permite la generación de nuevos servicios y la adaptación de los existentes**
 - **VoIP puede ser desplegado de múltiples maneras**

Smart Edge,
Dumb network
(Inteligencia en el CPE)

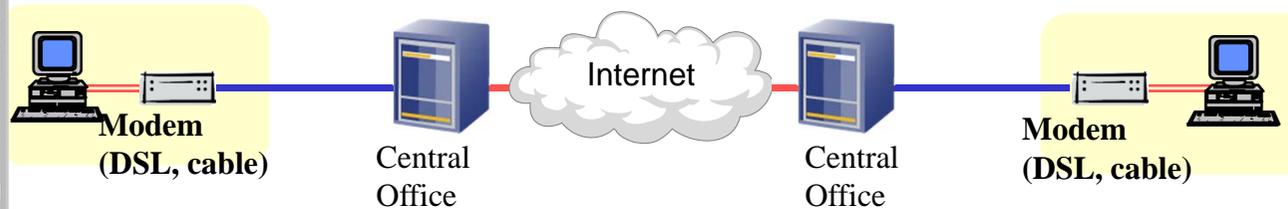


Smart Network,
Dumb Edge
(Inteligencia Centralizada)



Smart Network,
Smart Edge
(Inteligencia Distribuida)

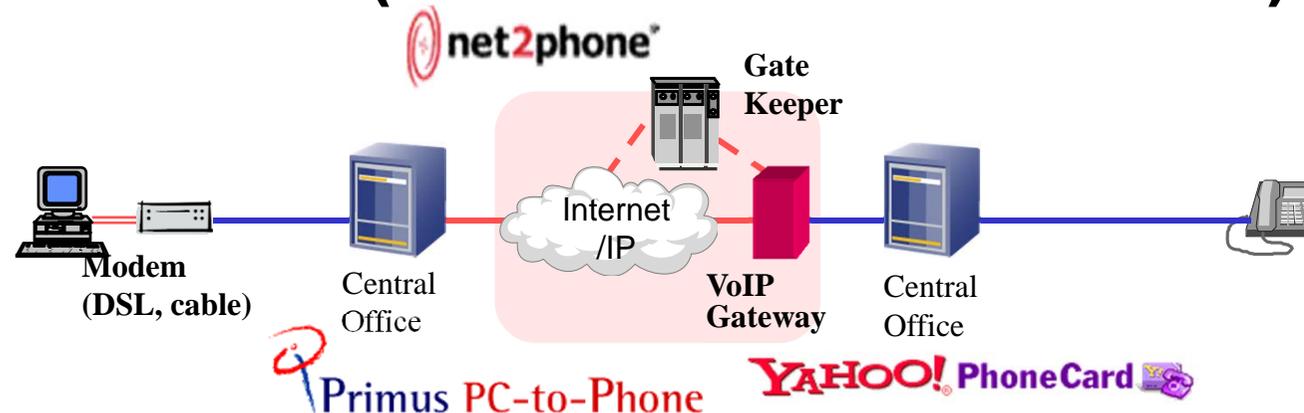
PC-PC (Internet-to-Internet)



- Usuario
 - Llamadas gratuitas entre PCs
 - Baja calidad y disponibilidad... (es lo que dicen los operadores!)
- Tecnología
 - Software de cliente (Microsoft NetMeeting) en ambos extremos
 - Dispositivos especializados de provisión de VoIP (Videoconferencia)
- Red
 - El tráfico atraviesa la red pública (Internet)
 - No compatible con RTC

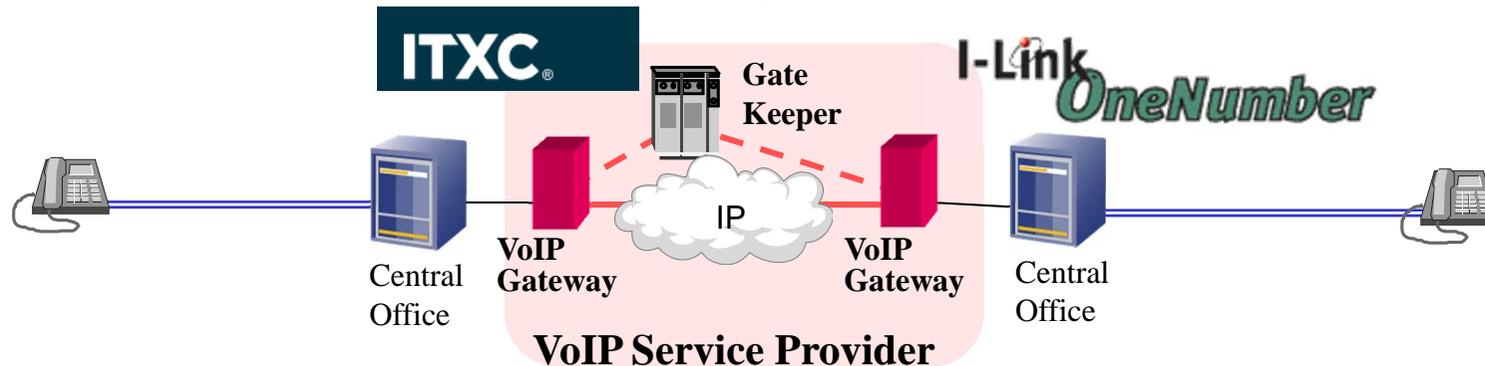


PC-Teléfono (Interconexión con RTC)



- Usuario
 - Llamadas entre un PC y un teléfono tradicional, con mediación
 - Coste asociado a la interconexión
 - Calidad variable, no solo dependiente del ISP
- Tecnología
 - Software de cliente
 - Dispositivos especializados de provisión de VoIP (Videoconferencia)
- Red
 - Tráfico transportado por la red pública (Internet) o por el proveedor del servicio de interconexión
 - Requiere de un VoIP Gateway para realizar la interconexión

Teléfono-Teléfono (Túneles de VoIP)



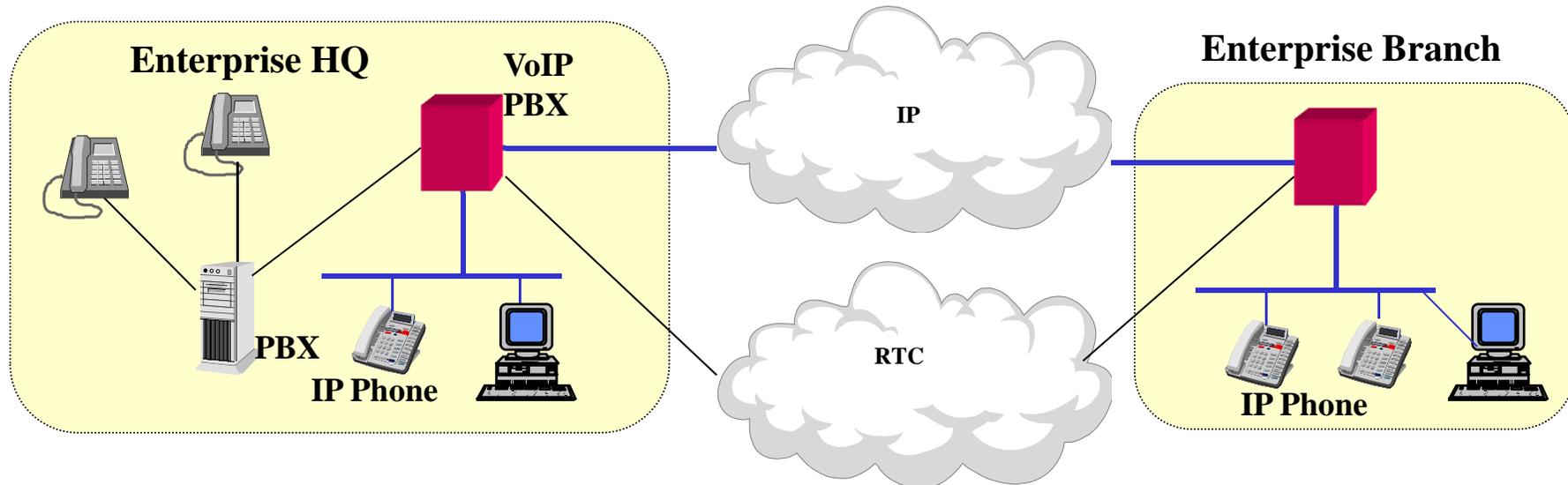
▪ Usuario

- Llamadas de larga distancia baratas mediante tarjetas prepago
- Servicios ofrecidos desde 1995 debido al sobrecoste de tarifas internacionales
- Puede requerir marcaciones de hasta 24 dígitos hasta establecer la llamada
- Calidad variable, dependiente del proveedor del servicio de túnel VoIP
- VoIP es invisible

▪ Red

- Cobertura e interconexión global entre ISPs y proveedores de VoIP

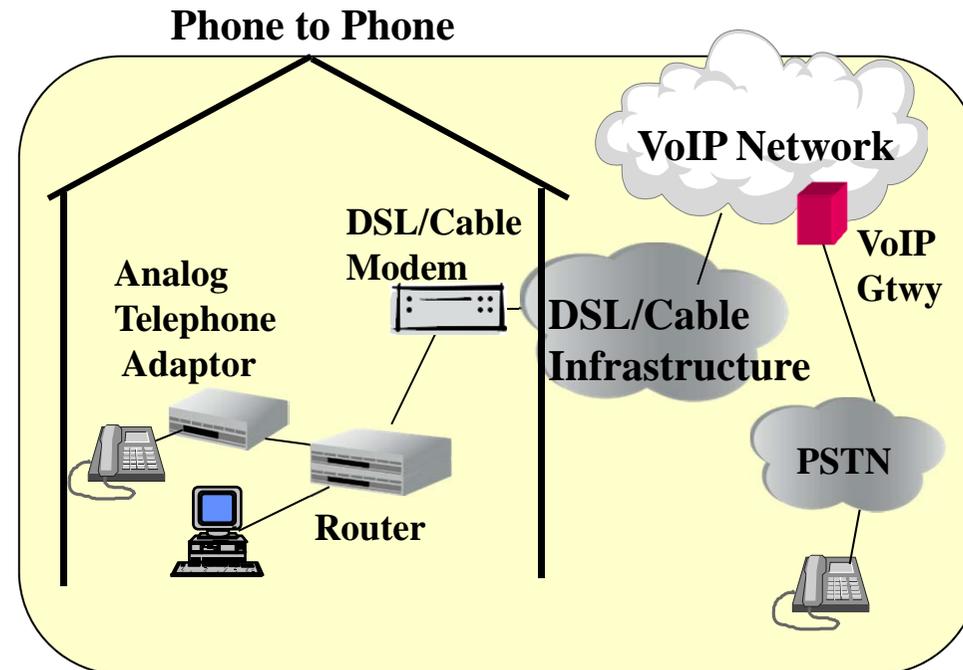
Teléfono-Teléfono (Centralitas PBX)



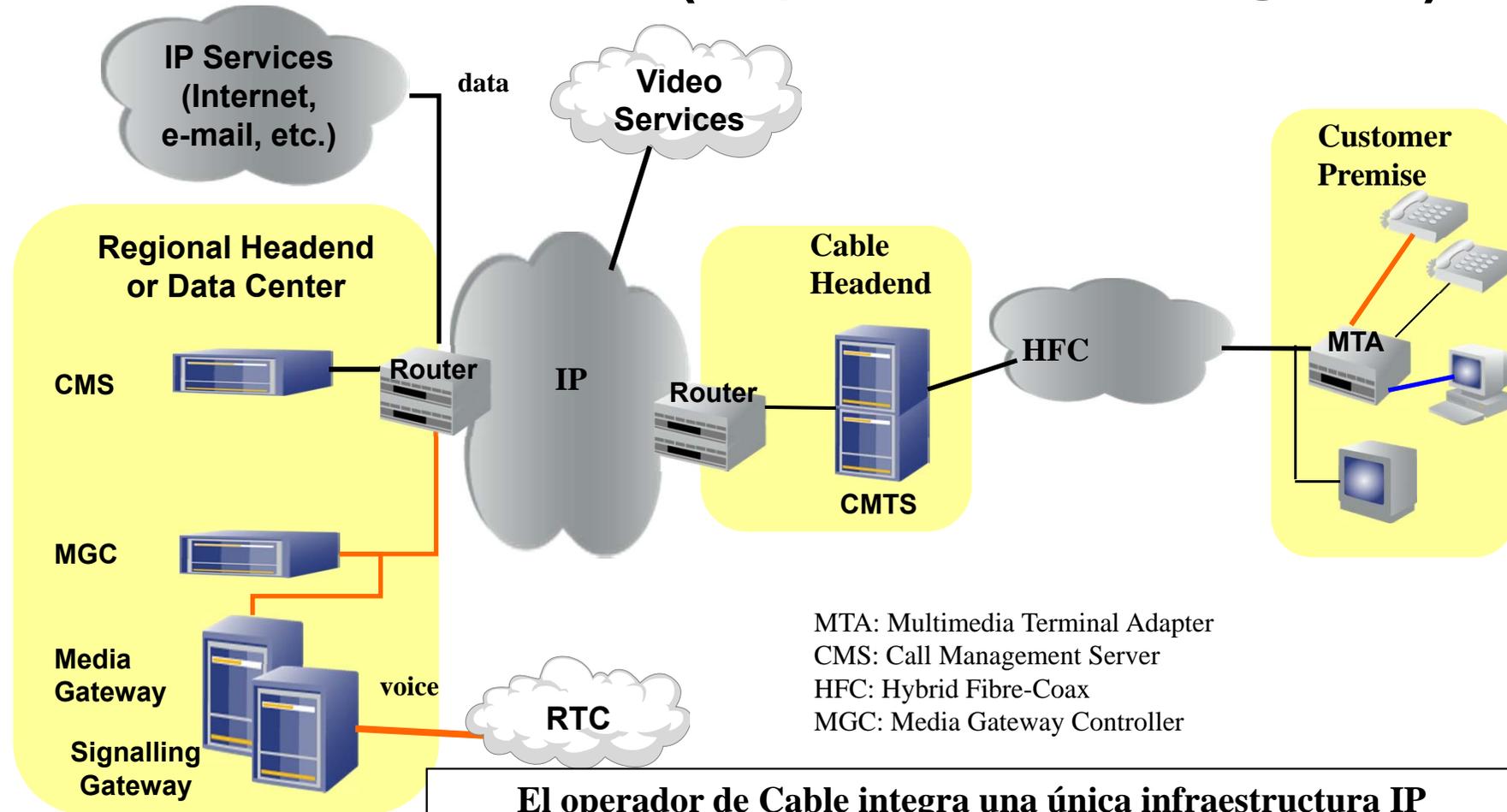
- Centralitas PBX con capacidad VoIP proveen las funciones de Media Gateway y router
- Soporte para voz y datos
- La voz/datos interempresas es transportada por la red IP.
- La RTC solamente cursa el excedente de tráfico y la voz hacia destinos RTC
- Acceso a los recursos corporativos desde accesos remotos
- Selección de red en función del destino

Teléfono-Teléfono (ISP de Banda Ancha)

- El propio ISP utiliza y provee el servicio de VoIP
- Usuario:
 - Llamadas locales y entre operadores “amigos” gratuitas
 - Llamadas a RTC de bajo coste (o gratuitas)
 - Servicios similares a la línea RTC
 - Interconexión totalmente transparente
- Proveedor
 - Fiabilidad – Servicio best effort
 - Dependiente de la disponibilidad de proveedor de acceso
 - Portabilidad de números
 - Llamadas de emergencia



Telefono-Teléfono (Arquitecturas convergentes)



El operador de Cable integra una única infraestructura IP ofreciendo TODOS los servicios (voz, datos, TV and Móvil) compitiendo con los Operadores de Red

Source: Lemur Networks

VoIP: Problemas

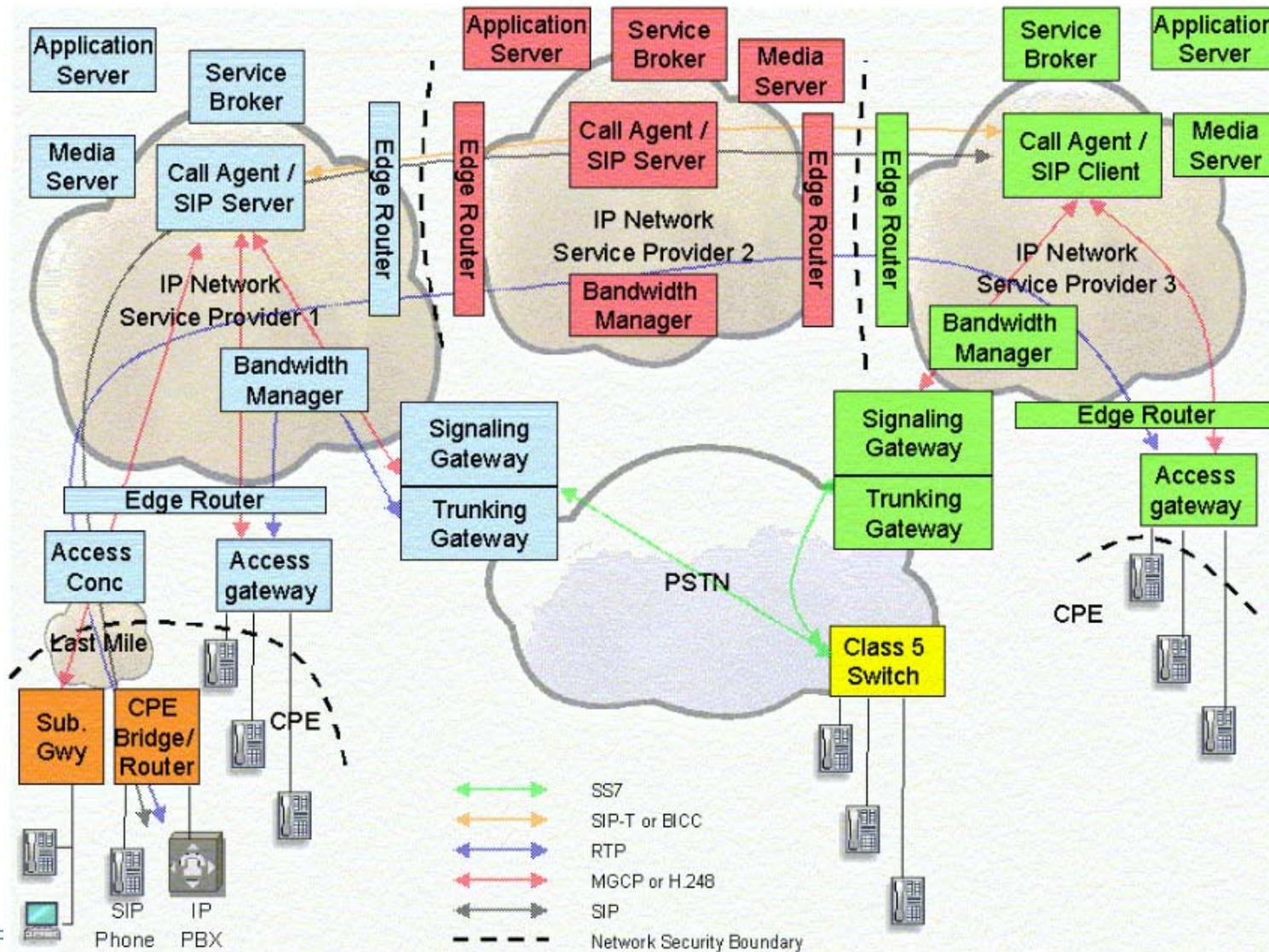
Técnicos

- La interoperabilidad e interfuncionamiento de islas VoIP
- Interconexión por defecto (y menos eficiente) PSTN / TDM
- La calidad de voz depende de la ingeniería de la red IP
- Mecanismo de seguridad
- Interceptación legal
- Alimentación local o alimentación de red en caso de cortes de energía
- Llamadas de emergencia
- Agotamiento de los números de teléfono

Regulatorios

- Según la CMT:
 - Numeración independiente
 - No se permite la portabilidad
 - El abonado no tiene derecho a aparecer en directorios telefónicos.
- Mientras la telefonía tradicional, fija y móvil, está regulada como Servicio Telefónico Disponible al Público (STDP), la telefonía IP está regulada como "servicios vocales nómadas con capacidad multimedia" (SVN)
- Llamadas de emergencia

La "Selva" de la VoIP



P2P: Definición

- *"Peer-to-peer es una forma de estructuración de las aplicaciones distribuidas tales que los nodos individuales tienen roles simétricos. En lugar de estar dividido en clientes y servidores, cada uno con funciones bien distintas, en las aplicaciones P2P un nodo puede actuar como un cliente y un servidor ."*
 - Peer-to-peer Research Group, IETF/IRTF, June 24, 2004
(<http://www.irtf.org/charters/p2prg.html>)
- El servicio precursor: "Computación P2P es la distribución de los recursos informáticos y de servicios de intercambio directo entre los sistemas."
 - Estos recursos y servicios incluyen el intercambio de información, ciclos de procesamiento, almacenamiento caché y disco.
 - Se aprovecha de las capacidades de almacenamiento y conectividad de la red, lo que permite a los usuarios aprovechar su poder colectivo para el "beneficio" de todos.
 - http://www-sop.inria.fr/mistral/personnel/Robin.Groenevelt/Publications/Peer-to-Peer_Introduction_Feb.ppt

P2P = Compartir

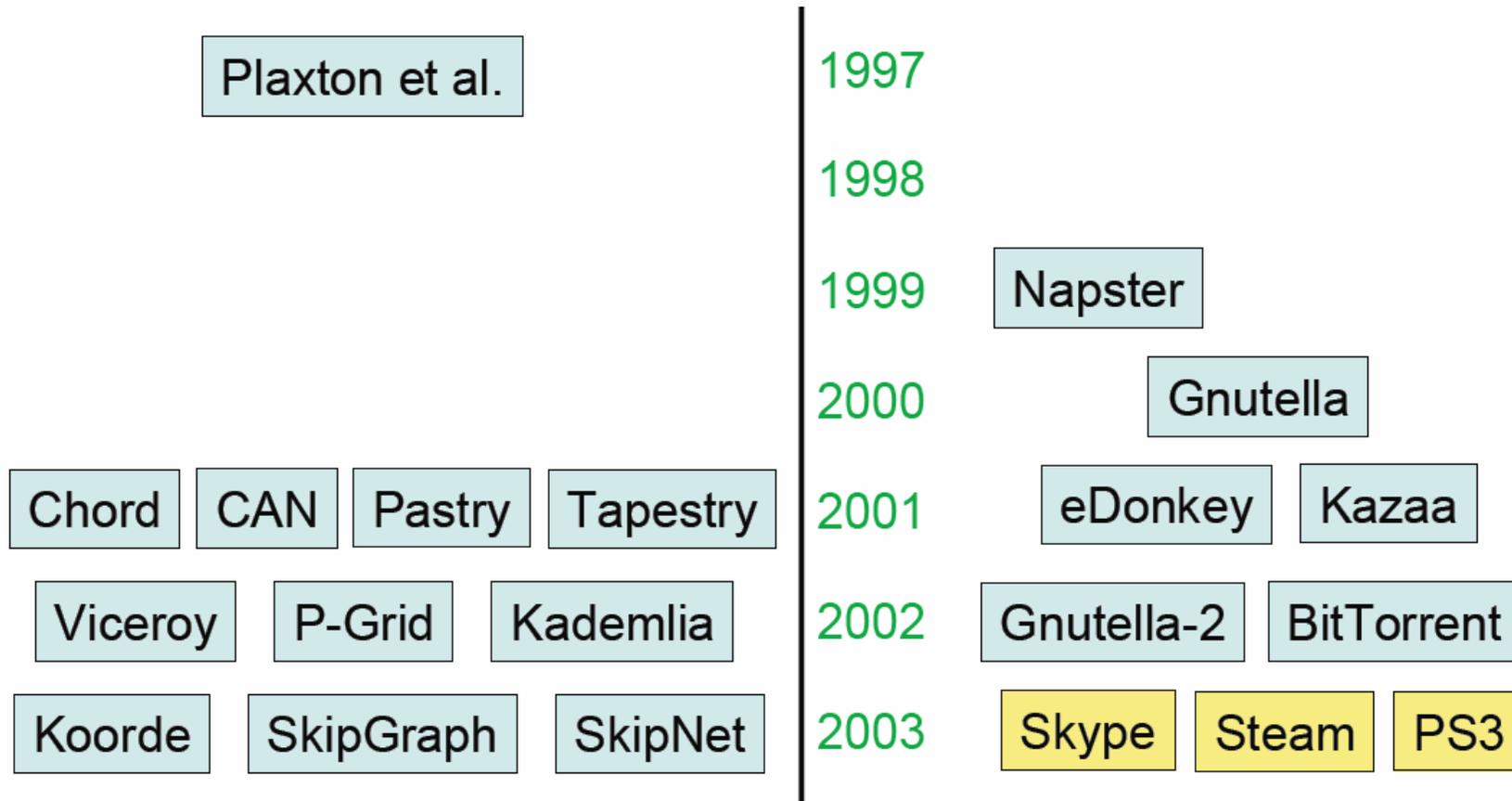
¿Qué podemos compartir?

- Recursos relacionados con la informática
- Recursos relacionados con la computación:
 - Ciclos de CPU - @ seti casa, GIMPS
 - Ancho de Banda - PPLive, PPStream
 - Espacio de almacenamiento - Oceanstore, Murex
 - Datos - Napster, Gnutella
 - Gente - Buscador de Amigos
 - Cámara, micrófono, Sensor, Servicio???.....

¿Cómo podemos compartir?

- Todos los nodos son tanto clientes como servidores ("servent")
 - Proporcionan y consumen datos
 - Cualquier nodo puede iniciar una conexión
- No hay fuentes centralizadas de datos
- *"La última forma de la democracia en Internet"*
- *"La última amenaza contra el derecho y la protección de contenidos en Internet"*

El BigBang del P2P



P2P: Características

- **Uso eficiente de los recursos**
 - ancho de banda no utilizado, almacenamiento, capacidad de procesamiento en el extremo de la red...
- **Escalabilidad**
 - No hay necesidad de implementar servidores para satisfacer la demanda
- **Los consumidores de recursos también donan recursos**
 - Los recursos agregados crecen de forma natural con la utilización
- **Confiabilidad**
 - Replicas
- **Distribución geográfica**
 - No existe ningún punto único de fallo
- **Facilidad de administración**
 - Auto-organización de los nodos
 - Incluye la tolerancia a fallos, replicación y balanceo de carga

Redes P2P vs Redes Overlay

Red P2P

Se refiere a las aplicaciones que se aprovechan de los recursos (almacenamiento, ciclos, los contenidos, la presencia humana) disponibles en los sistemas finales de la internet.

Red Overlay

Se refiere a las redes que se construyen en la parte superior de otra red (por ejemplo, IP).

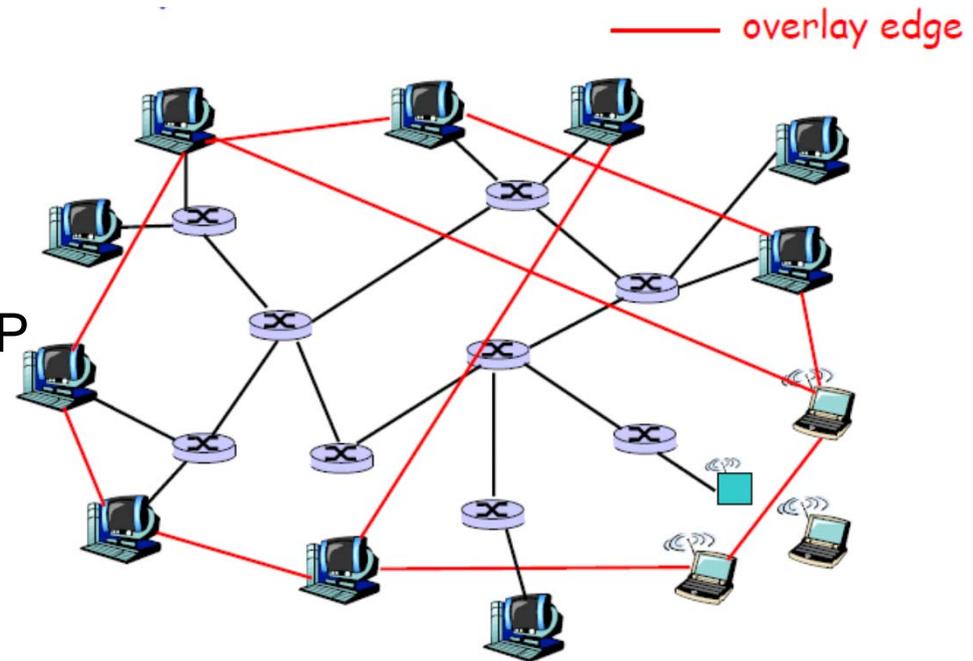
Red overlay P2P

Cualquier red overlay que se construye por pares de Internet en la capa de aplicación, en la parte superior de la red IP

Red Overlay \approx Red de capa de aplicación

- Flexibilidad de diseño
 - Topología, mantenimiento
 - Tipos de mensajes
 - Protocolo
 - Mensajería a través de TCP o UDP

- Red física subyacente transparente al desarrollador
 - Algunas explotan proximidad



Redes overlay P2P: Funcionamiento

- Nodos virtuales (Virtual Edge)
 - Los enlaces se establecen por conexión TCP
 - o simplemente un puntero a una dirección IP
- Mantenimiento del overlay
 - Periódicamente ping para asegurarse que el vecino sigue vivo
 - O verificar su vitalidad analizando su mensajería
 - Si el vecino deja de funcionar, puede que desee establecer nuevo edge
 - Los nuevos nodos necesitan “arrancar” en el overlay
- El principal problema son las altas tasas de deserción (churn)
 - Definición de **Churn** en P2P:

Topología dinámica y acceso intermitente
debido a la llegada/desaparición de nodos

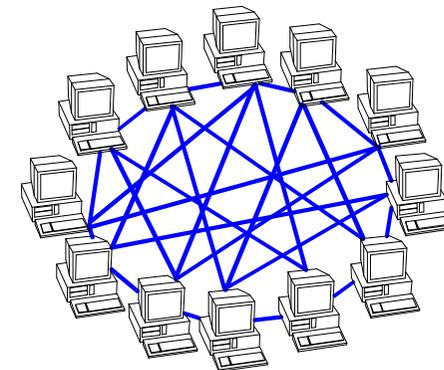
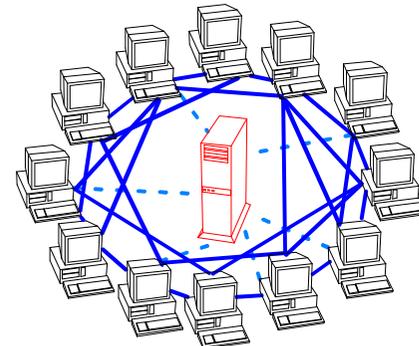
Clasificación de redes overlay

- Redes Overlay Estructuradas
 - Se basan en Tablas Hash Distribuidas (DHT)
 - la red asigna claves a cada elemento de datos, mapea cada “peer” en un grafo y relaciona cada clave con cada “peer”.
 - Ejemplos: Content Addressable Network (CAN), Chord, Pastry
- Redes Overlay No estructuradas
 - Las red organiza a los peer en un grafo aleatorio, plano o jerárquico.
 - Para enviar consultas utiliza inundaciones o paseos (Walkthrough) Cuando un peer recibe la consulta, envía una lista de todo el contenido coincidente a los peer de origen.
 - A su vez se dividen en:
 - Directorios centralizados
 - Sistemas P2P puros
 - Sistemas P2P Híbridos
 - Ejemplos: FreeNet, Gnutella, KaZaA, BitTorrent
- Overlay de redes multicast
 - Los peer se organizan en un árbol para el multicast.

Redes no estructuradas (I)

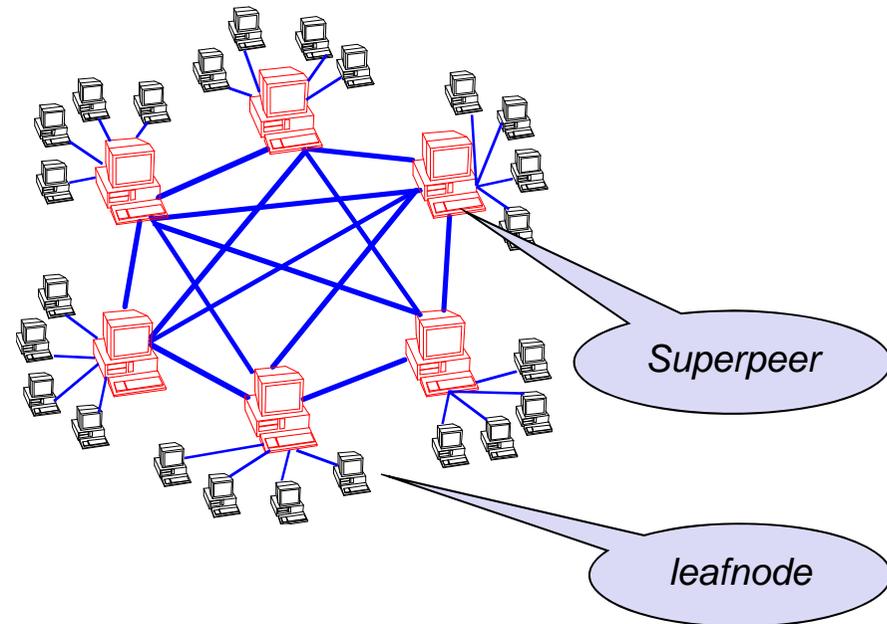
- Sistemas P2P basados en Directorio centralizado
 - Todos los peers están conectados a una entidad central
 - Los Peers establecen conexiones entre sí para el intercambio de datos de usuario (por ejemplo un mp3)
 - La Entidad central es necesaria para prestar el servicio
 - Es una especie de base de datos de índice / grupo
 - Mantiene tablas de búsqueda / de enrutamiento
 - Ejemplos: Napster, Bittorrent

- Sistemas P2P puros
 - Cualquier entidad terminal puede ser eliminada sin pérdida de funcionalidad
 - No hay entidades centrales
 - Los Peers establecen conexiones entre sí al azar
 - Para los mensajes de petición de ruta y respuesta
 - Para insertar mensajes de solicitud
 - Ejemplos: Gnutella, Freenet



Redes no estructuradas (y II)

- Sistemas P2P híbridas
 - Característica principal :
Introducción de otra capa
jerárquica dinámica
 - Proceso de elección para
seleccionar una asignación
de “Superpeers”
 - Superpeers: de alto grado
(grado $\gg 20$, dependiendo
del tamaño de la red)
 - Leafnodes: conectado a uno
o más Superpeers (grado
 < 7)
 - Ejemplos: eMesh, KaZaA



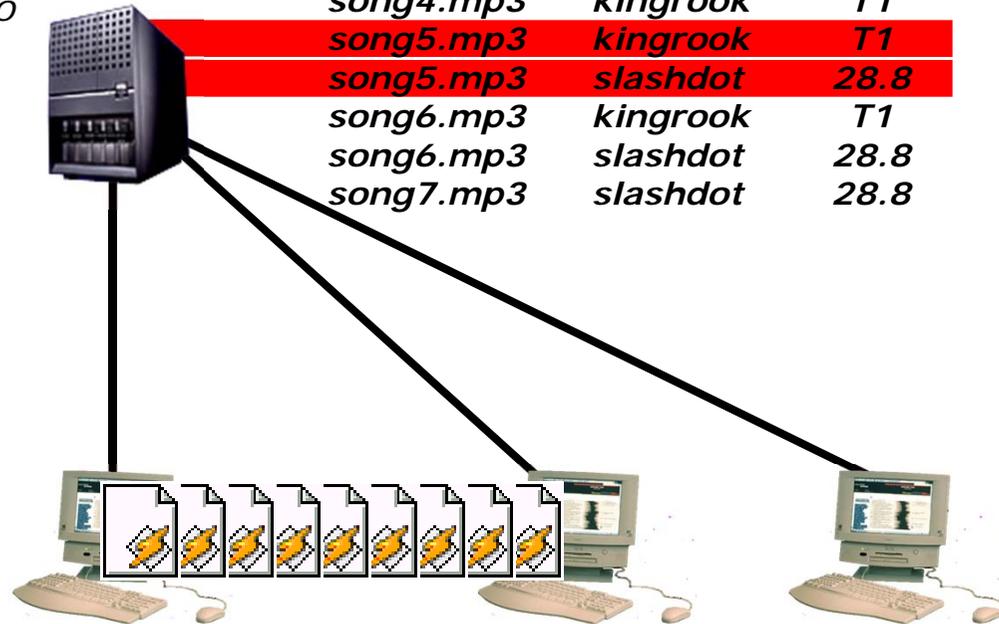
Napster (P2P de Directorio Centralizado)

1. El usuario lanza la aplicación cliente y conecta con el servidor de Napster, indicándole la dir. IP y los contenidos propios
2. Napster crea un directorio dinámico a partir de las librerías de mp3 de los usuarios conectados
3. *beastieboy* introduce una búsqueda



4. Napster muestra los resultados a *beastieboy*
5. *beastieboy* se conecta directamente con *kingrook* para transferir el archivo

Title	User	Speed
song1.mp3	beastieboy	DSL
song2.mp3	beastieboy	DSL
song3.mp3	beastieboy	DSL
song4.mp3	kingrook	T1
song5.mp3	kingrook	T1
song5.mp3	slashdot	28.8
song6.mp3	kingrook	T1
song6.mp3	slashdot	28.8
song7.mp3	slashdot	28.8



"beastieboy"

- song1.mp3
- song2.mp3
- song3.mp3
- **song5.mp3**

"kingrook"

- song4.mp3
- song5.mp3
- song6.mp3

"slashdot"

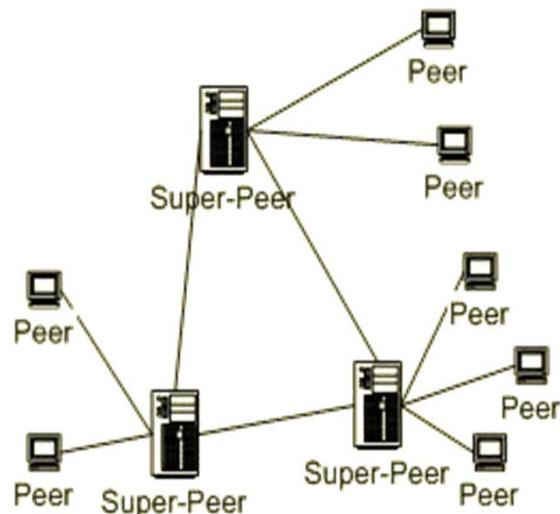
- song5.mp3
- song6.mp3
- song7.mp3

Búsqueda centralizada / Copia P2P

iMesh / Kazaa (Híbrido de Napster y Gnutella)

- Los Superpeers actúan como centros de búsqueda local
 - Cada Superpeer es similar a un servidor de Napster para una pequeña porción de la red
 - Se eligen automáticamente en función de sus capacidades (almacenamiento, ancho de banda, etc) y la disponibilidad (tiempo de conexión)
 - Cada uno gestiona 100-150 peers y se conecta a 30-50 de su nivel
- Los usuarios cargan su lista de archivos a un Superpeer
 - Los Superpeers intercambian periódicamente listas de archivos

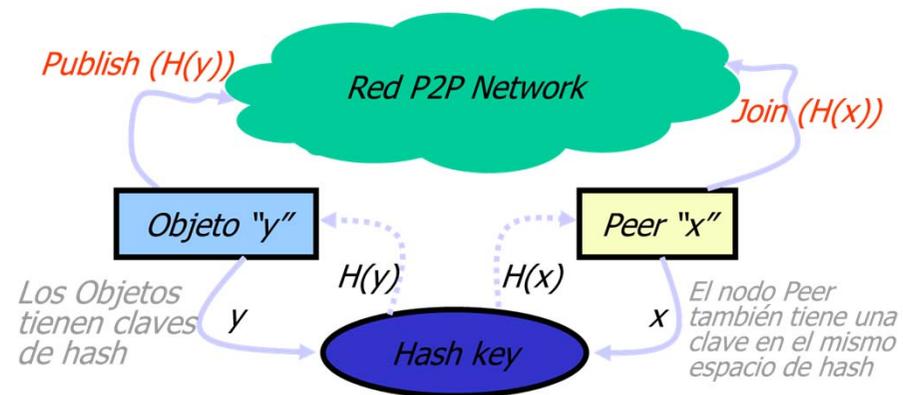
- Peticiones:



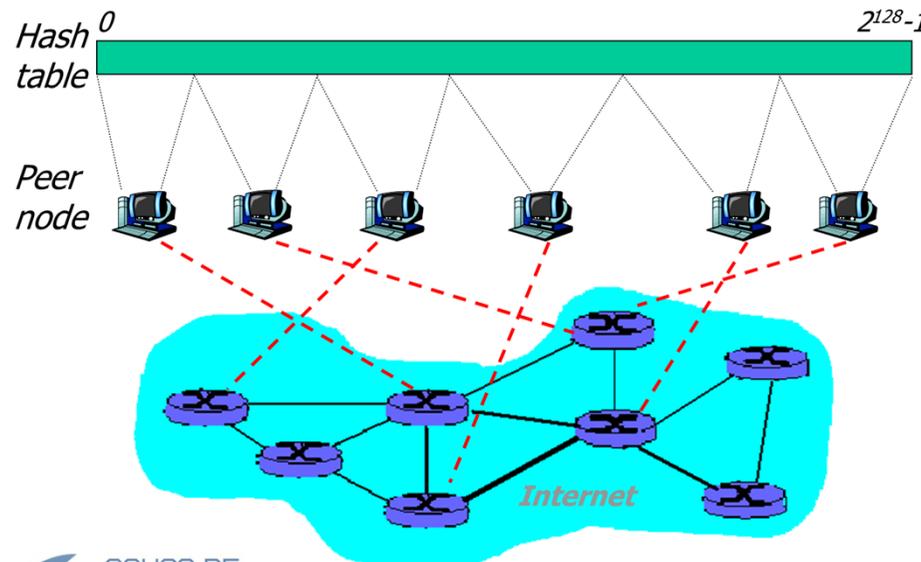
1. Cada archivo tiene un hash y un descriptor
2. El cliente envía una consulta de palabras clave a su líder de grupo
3. El líder del Grupo responde con los resultados:
 - Para cada resultado: metadatos, hash, la dirección IP
 - Si reenvía la consulta para otros líderes de grupo, responde con los resultados de éstos
4. El Cliente selecciona entonces los archivos para descargar y envía peticiones HTTP a los Peer que lo comparten utilizando como identificador el hash

Redes Overlay Estructuradas: DHT

- DHT=Distributed Hash Table
 - Entrada: clave (nombre de archivo)
 - Salida: Valor (localización del archivo)
- Cada nodo es responsable de un área de la tabla hash, de acuerdo con la clave hash del nodo.
- Los Objetos se sitúan en (gestionado por) el nodo con la clave hash más cercana



El objeto se localiza en el peer con clave hash más cercana



- Cada clave apunta de manera uniforme a todos los nodos en la red
- Cada nodo mantiene información acerca de sólo unos pocos otros nodos
- Los mensajes pueden ser enviados a un nodo de manera eficiente
- Cada petición/consulta sólo afectará a unos pocos nodos

Redes Overlay Estructuradas: Enrutamiento

- DHT es el interfaz genérico
- Las diferencias entre soluciones radican en la implementación de este interfaz
 - Chord [MIT]
 - Pastry [Microsoft Research UK, Rice University]
 - Tapestry [UC Berkeley]
 - Content Addressable Network (CAN) [UC Berkeley]
 - SkipNet [Microsoft Research US, Univ. of Washington]
 - Kademlia [New York University]
 - Viceroy [Israel, UC Berkeley]
 - P-Grid [EPFL Switzerland]
 - Freenet [Ian Clarke]
- Todos estos sistemas son a menudo los conocidos como enrutamiento P2P o simplemente redes overlay P2P

BitTorrent... Overlay P2P Desestructurada... pero no tanto

- Actualmente el 20-50% del tráfico de Internet es BitTorrent
- Se necesita un software especial
 - BitTorrent, BitTyrant, uTorrent, LimeWire ...

▪ Idea básica

- Los clientes que descargan un archivo al mismo tiempo, se ayudan entre sí (es decir, también suben trozos entre sí)
- Los clientes de BitTorrent forman un enjambre (swarm): una red overlay aleatoria

Publicar un archivo

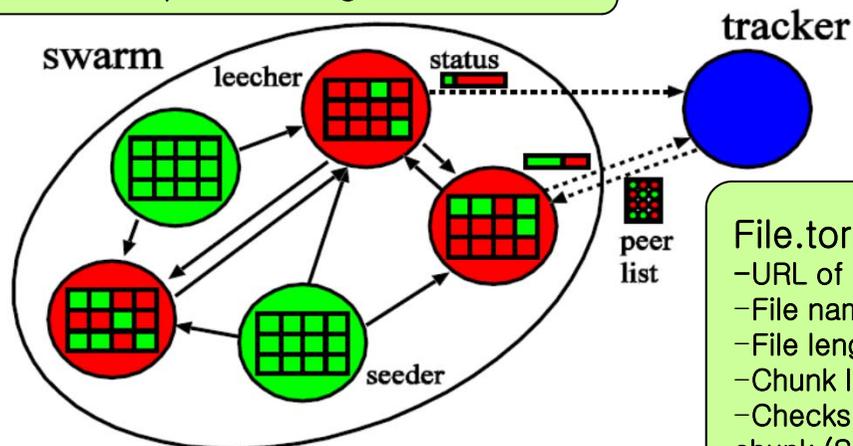
- Se sube un archivo ".torrent" a una web
- Arrancar un tracker, servidor que:
 - Da acceso a peers aleatoriamente, desde/hacia donde descargar
 - Recoge estadísticas del swarm

- Existen implementaciones "trackerless" que hacen uso de Kademia DHT (ejm. Azureus)

Descargar un archivo

- Con un cliente bittorrent se selecciona el archivo ".torrent" del recurso a descargar

Seeder – peer con el archivo completo
Leecher – peer descargando el archivo



File.torrent :
 -URL of tracker
 -File name
 -File length
 -Chunk length
 -Checksum for each chunk (SHA1 hash)

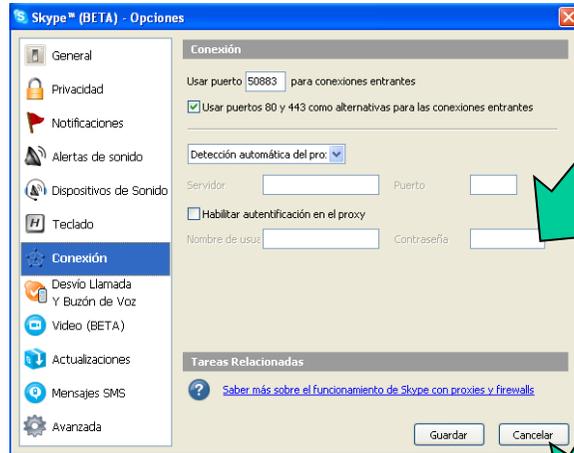
... todo está pensado

- El cliente primero pregunta a 50 peers seleccionados aleatoriamente del listado del tracker
 - Aprende todos los detalles acerca de los “chunks” (256K) que contienen
- Selecciona un chunk e intenta descargarlo por partes (16K) desde los nodos vecinos
 - La descarga no se lleva a cabo si el vecino está desconectado o deniega la descarga (choking)
 - Solamente los chunk completos pueden ser compartidos con otros
- Solamente se permite la descarga a 4 vecinos (unchoking)
 - Periódicamente (30s) se permite descargar a un peer aleatorio (optimistic unchoking)
 - El resto del tiempo se permite (cada 10s) la descarga a peers que permitan a su vez descargar.
- BitTorrent sigue la filosofía del “tit-for-tat” (cooperar primero y entonces observar lo que hacen en el otro lado)
 - Un cliente permite la descarga a otros peers que permiten descargar desde ellos
 - Optimistic unchoking solo durante el arranque
- Qué chunk se selecciona para descargar?
 - El chunk que aparece con menos frecuencia (el más raro) entre los vecinos (decisión Local)
 - Incrementa la diversidad de los trozos descargados
 - Incrementa la probabilidad de que todas las partes estén disponibles incluso cuando la fuente original (seed) desaparece antes de que otro nodo haya descargado el archivo completo
 - Excepto el primer chunk
 - Elegido aleatoriamente (preferiblemente que aparezca en muchos vecinos)

Skype... VoIP ó P2P

- Skype es un cliente gratuito P2P desarrollado por Kazaa.
- Permite a sus usuarios realizar:
 - Llamadas de voz.
 - Videoconferencias.
 - Mensajería instantánea
- La red Skype es una red Overlay P2P que conecta iguales entre sí.
 - **Nodo Normal:** nodo donde se ejecuta el cliente Skype
 - **Supernodo:** nodo con una dirección IP pública que tiene suficiente CPU, memoria, y ancho de banda de red
 - **Servidor de conexión:** Permite la autenticación del usuario y almacena los nombres de forma única y las contraseñas del usuario.
- Para la codificación de audio Skype se utilizan los códecs iLBC, iSAC y iPCM, todos ellos desarrollados por GlobalIPSound que permiten utilizar frecuencias comprendidas entre los 50 y los 8000 Hz.
- Skype utiliza AES (*Advanced Encryption Standard*).

Skype: modo de funcionamiento (I)



Un cliente Skype abre puertos TCP y UDP que son generados aleatoriamente durante la instalación.

Skype mantiene en la caché del cliente una tabla con pares de direcciones IP de supernodos.

Existe una lista de contactos que se almacena encriptada en un fichero XML llamado 'config.xml'.

```
<?xml version="1.0" ??
- <config version="1.0" serial="6625" timestamp="1135714201.11">
- <Lib>
+ <Account>
+ <BCM>
- <Connection>
- <Bandwidth>
  <CurSlotLength>6008</CurSlotLength>
  <LastRtTestTime>1135714068</LastRtTestTime>
  <OutHistory>7974</OutHistory>
</Bandwidth>
<DisablePort80>0</DisablePort80>
+ <EventServers>
- <Firewall>
  <TcpInHistory>-1431655768</TcpInHistory>
  <UdpInHistory>-1431655768</UdpInHistory>
  <UdpOutHistory>1431655807</UdpOutHistory>
</Firewall>
- <HostCache>
  <_1>140.115.23.23:62601</_1>
  <_10>87.69.48.254:1586</_10>
  <_100>140.121.135.224:3256</_100>
  <_101>217.199.108.68:35749</_101>
  <_102>217.199.108.67:59107</_102>
```

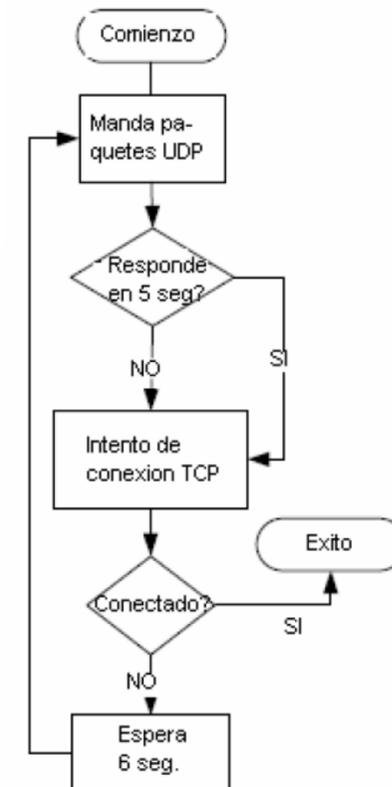
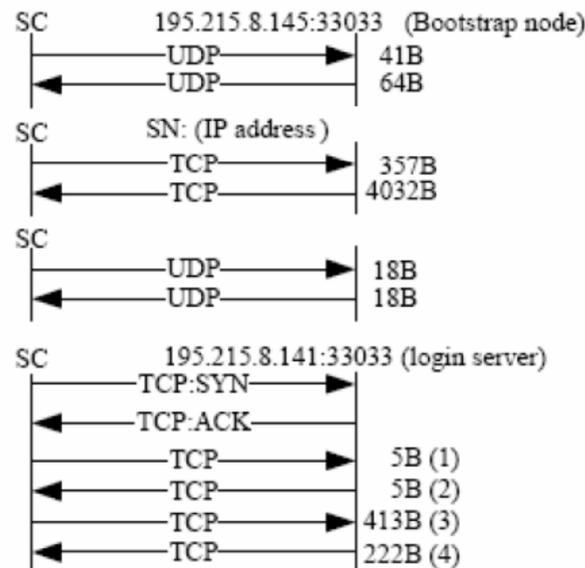
```
<CentralStorage>
  <LastBackoff>0</LastBackoff>
  <LastFailure>0</LastFailure>
  <LastSync>1135714076</LastSync>
  <NeedSync>0</NeedSync>
  <SyncSet>
    <u>
      <skypebuddy1>2f1b8360:2</skypebuddy1>
      <skypebuddy2>d0450f12:2</skypebuddy2>
```

Skype: modo de funcionamiento (II)

- Arranque
 - Tras la instalación envía un HTTP 1.1 GET Request al servidor de Skype
- 1ª Conexión
 - Operación mas crítica
 - Se autentica nombre y contraseña de usuario
 - Descubre nodos Skype con direcciones IP públicas
 - Determina si está detrás de un cortafuegos o de un NAT

- Siguietes conexiones

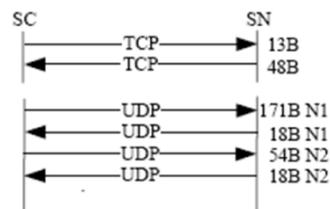
- Conexión P2P con el servidor
 - ↗ 212.72.49.141
 - ↗ 195.245.8.141
- Único componente centralizado
- Autentica cliente mediante usuario y contraseña



Skype: modo de funcionamiento (y III)

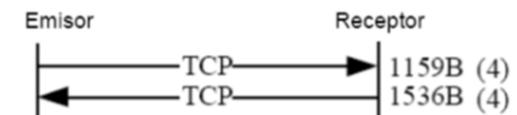
▪ Búsquedas

- El cliente envía la petición al supernodo, éste le responde enviándole 8 nodos.
- El cliente manda paquetes UDP a los 8 nodos.
 - Si el usuario a buscar no se encuentra, el cliente vuelve a enviar la petición al servidor y este le responde enviándole 16 nodos.
 - El proceso se repite hasta que se encuentra el usuario
- Como media el cliente encuentra al usuario tras entrar en contacto con más de 24 nodos



▪ Establecimiento de llamada

- La información se intercambia sobre TCP
- Intercambio inicial mediante challenge-response



▪ Codificación y transferencia

- Transferencia de voz se realiza sobre UDP
- La codificación utilizada es iSAC

▪ Mensajes de tiempo de vida

- El cliente Skype refresca su conexión con el supernodo mediante mensajes TCP cada 2 minutos

