ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN
UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Proyecto Fin de Carrera

Diseño y Configuración de un Sistema de VoIP para la ETSIT de la Universidad Politécnica de Cartagena

AUTOR: Juan Antonio Martínez León
DIRECTOR: Pablo Pavón Mariño

Enero / 2006

Autor	Juan Antonio Martínez León
E-mail del Autor	Juan.Leon@tiscali.es
Director(es)	Pablo Pavón Mariño
E-mail del Director	Pablo.Pavon@upct.es
Codirector(es)	
Título del PFC	Diseño y Configuración de un Sistema de VoIP para la ETSIT de la Universidad Politécnica de Cartagena
Descriptores	Voz sobre *IP* (*VoIP*), telefonía *IP*, *SIP*, plan de marcación

Resumen

El objetivo de este Proyecto Fin de Carrera es el diseño y la configuración de un sistema de telefonía *IP* que permita ofrecer este servicio al conjunto de profesores que forman la Escuela Técnica Superior de Ingeniería de Telecomunicación de la Universidad Politécnica de Cartagena.

El primer paso es estudiar las dificultades y carencias que presenta la transmisión de voz sobre redes de paquetes para determinar un proceso de diseño completo que minimice las mismas y que permita el mantenimiento de conversaciones con una buena calidad de sonido.

Uno de los pasos clave en este proceso es la elección del protocolo de señalización sobre el que se soportará el sistema, pues determina tanto su flexibilidad ante futuras ampliaciones como los componentes a incluir en él.

Entre tales componentes se encuentran el servidor, los terminales de acceso de los usuarios y el *gateway* de interconexión a la RTB; a los que se pueden unir características avanzadas, como las funciones de autenticación, los planes de marcación, etc.

Titulación	Ingeniero de Telecomunicación
Intensificación	Planificación y Gestión de Telecomunicaciones
Departamento	Tecnologías de la Información y las Comunicaciones
Fecha de Presentación	Enero – 2006

Quiero aprovechar estas líneas, ahora que termina un ciclo de mi vida, para dar las gracias a todas las personas que me han hecho crecer moral y profesionalmente, a las que me han dedicado su tiempo y nunca les he dado las gracias, a las que me han apoyado sin pedírselo, a las que me han permitido equivocarme, a las que día a día me han hecho un poco más feliz y a las que me han enseñado que "el éxito no es más que la constancia en el propósito".

Índice de Contenidos

Índice de Tablas

Índice de Figuras

En los Apéndices:

Capítulo 1
Introducción

Este capítulo no es documento dedicado a detallar la historia de los avances acaecidos en la industria de las telecomunicaciones dedicada a las conversaciones de voz, pero se cree conveniente describir a grandes rasgos el proceso de cambio tecnológico que ha llevado al ser humano al punto actual.

Desde que *Alexander Graham Bell* inventase el teléfono, cosa que ahora parece dudosa, los avances tecnológicos asociados a las comunicaciones de voz hasta finales del último siglo se han basado en mejorar principalmente las características asociadas no al medio de transmisión, sino a los equipos y dispositivos que realizan las tareas de conmutación de los circuitos ya que no es posible ni viable construir una topología en malla donde exista una línea dedicada a cada enlace entre dos usuarios del servicio. En este campo, el hito más importante ha sido el paso de analógico a digital (modulación por impulsos codificados - *PMC*), que ha hecho recorrer el camino hasta el desarrollo de conmutadores electrónicos.

Tradicionalmente, el servicio telefónico se ha basado en establecer el circuito virtual entre los extremos de una llamada (conmutación de circuitos - *circuit switching*), fase durante la cual se realiza principalmente el intercambio de señalización de control y encaminamiento; y proceder a realizar el intercambio de información, tiempo durante el cual todos los recursos de la red son reservados entre los extremos de la comunicación (emisor-receptor) hasta que se completa la transferencia.

En cuanto a las redes de datos, cuando *Internet* comenzó con unos militares, universitarios y de investigación; pocos anticiparon la revolución que provocaría en la forma de comunicación; a lo largo de todo el mundo y en un espacio de tiempo relativamente corto, unos 30 años.

Su filosofía, basada en el tratamiento independiente de bloques de información, (paquetes dotados de direcciones de origen y destino que se van retransmitiendo); y el hecho de funcionar sobre un medio en este caso es compartido, de forma que pueden ir por el mismo enlace unidades de información con distinto origen y destino simultáneamente; no adelantaba el éxito de aplicaciones como el correo electrónico o la navegación web, que han llegado a ser tan populares que se usan ya no sólo en todas las organizaciones del mundo, sino en todos los hogares del mundo.

El siguiente paso evolutivo de estas formas de comunicación pasa por la consolidación de las aplicaciones en tiempo real como una nueva herramienta de negocio, principalmente motivado porque casi la totalidad de las expresas y usuarios de ámbito residencial ya tienen acceso a *Internet* a partir de tecnologías de banda ancha o, al menos, una conexión fija. Todo esto provoca una revolución de los servicios a proveer, lo que conlleva el trabajo en aplicaciones novedosas en los campos de:

- Voz (donde la telefonía *IP* es sólo un componente).
- Video.
- Mensajería instantánea.

- Multiconferencias de voz, video y datos.
- ...

En cuanto al modelo telefónico estamos a las puertas de una nueva revolución, puesto que ya existe una cierta cantidad de usuarios que ha experimentado las características de los terminales *software* en sus equipos informáticos; y de un nuevo *handicap*, debido a que se ha de conseguir que los usuarios del servicio tradicional accedan a este modelo de forma no traumática si se desea que éste realmente triunfe.

El término clave en este cambio es lo que denomina voz sobre *IP*, de aquí en adelante, *VoIP* (*Voice over Internet Protocol*), es decir, la posibilidad de realizar llamadas telefónicas y acceder a servicios sobre redes *IP* (de conmutación de paquetes) con una calidad adecuada. Este término cubre desde el uso de Internet para realizar llamadas telefónicas de forma gratuita basándose en una arquitectura igual a igual (*peer to peer*) hasta un cambio de infraestructuras que suponga la sustitución de las redes de tipo conmutado.

Llegados a este punto, la pregunta es por qué realizar este cambio cuando la telefonía analógica tradicional funciona correctamente. La respuesta viene motivada, atendiendo al análisis DAFO (Debilidades, Amenazas, Fortalezas y Oportunidades) por una fortaleza (el coste) y una oportunidad (la posibilidad de abrir nuevos ámbitos de negocio):

- Reducción de costes:
  - Reducción de precios de los equipos de las redes de datos (70% más baratos que los de las redes de voz) y supresión de la necesidad de adquirir equipamiento telefónico.
  - Reducción de precios en los alquileres de líneas. El coste de las líneas de datos es entre un 60% y un 80% que el de las líneas de voz.
  - Reducción de los costes de mantenimiento (50%) y operación (72%). Se requieren menos equipos y, por tanto, menos personal para su mantenimiento.
- Desarrollo y provisión de servicios de valor añadido:
  - Mensajería unificada: e-mail junto con aplicaciones de voz.
  - Soluciones de comercio electrónico y atención al cliente vía web, que conjugan la información de las páginas de las organizaciones con la posibilidad de acceder centros de llamadas, donde algún empleado pueda proporcionar información adicional a los usuarios/clientes.
  - Soporte de aplicaciones multimedia, abriendo la posibilidad de ofrecer video-conferencia, video *streaming*, pizarra compartida, etc.

Una vez expuestas las ventajas, de acaecer el cambio también resulta interesante e incluso conveniente atender a las reacciones que se pueden producir en los distintos agentes que son influenciados por el mismo. De este modo existen básicamente dos puntos de vista:

- Proveedores de servicios:
    - Desde el punto de vista de los proveedores de servicio telefónico tradicional, esta tecnología se ha aplicado principalmente en los enlaces troncales para reducir los costes de transporte.
    - El hecho de que se pueda transmitir tráfico de voz sobre redes de conmutación de paquetes ha convertido a los proveedores de acceso a Internet (*ISPs*) en potenciales proveedores de servicio telefónico, permitiendo y fomentando la competencia.
    - Tanto para proveedores de telefonía clásicos como para los de nueva aparición esta tecnología ofrece la oportunidad de mejorar y ofrecer nuevos servicios, que permitan aumentar sus beneficios.
- Clientes:
    - Grandes empresas: las empresas multinacionales con sucursales en lugares geográficamente dispersos han encontrado en la *VoIP* una forma de ahorrar en su factura telefónica. Este proceso comprende dos pasos:
        - El comienzo del desarrollo de la *VoIP* se da debido al hecho de que estas empresas ya tenían enlaces de datos (*Frame Relay, ATM,* etc) entre sus sucursales para acceder a recursos comunes tales como bases de datos. De este modo, el servicio telefónico sobre redes de paquetes surge como aprovechamiento de la infraestructura de datos, puesto que proporcionaba voz corporativa a coste cero entre distintas sucursales.
        - Un segundo paso se da en el tiempo actual, donde la aparición de diferentes proveedores de telefonía *IP* permite a las compañías realizar llamadas independientemente del extremo final a costes más reducidos.
    - Pequeñas y medianas empresas: la aparición de nuevos proveedores de servicio telefónico con tarifas más económicas han promovido que los responsables de estas organizaciones vean sus accesos a Internet de banda ancha (en España fomentados por planes gubernamentales como la iniciativa PYME.es dentro del plan España.es o más actualmente con la iniciativa Red.es y Todos.es) como una oportunidad para disminuir significativamente su factura telefónica.

Además de entender las consideraciones de los proveedores / clientes resulta conveniente tener una serie de consideraciones en cuenta con respecto al ámbito regulatorio que afecta, sobre todo, a los proveedores.

En nuestro país, la Comisión del Mercado de la Telecomunicaciones (CMT) ha tomado ya iniciativas para facilitar la provisión del servicio de *VoIP* no sólo en comunicaciones internas empresariales o a través de Internet, sino también en el ámbito residencial, pese a que aún no se ha llegado a considerar este servicio como "servicio telefónico disponible al público" (punto que se está debatiendo). Entre estas destacan [1]:

- Mantener la obligación de encaminamiento gratuito de las llamadas al número 112.
- Asignar dos rangos de numeración estos servicios; uno de numeración geográfica compartido con el servicio telefónico fijo (rango 8) y otro de numeración específica (rango 51). Queda de este modo habilitado un bloque de 10 millones de números identificados por el prefijo 51 que, al no tener información geográfica, podrán ser utilizados para prestar servicios nómadas en todo el territorio nacional español.

Por ello, antes de proveer el servicio de telefonía *IP* se debe establecer de forma clara el ámbito de actuación del mismo para conocer la reglamentación aplicable. En próximos capítulos se realiza una exposición teórica de las consideraciones que se han de tener en cuenta de forma previa a la implantación de un servicio de *VoIP* para uso privado con interconexión a la red telefónica básica, se revisan los componentes básicos del mismo y se particularizan a las necesidades de la instalación en un escenario real, la Escuela Técnica Superior de Ingeniería de Telecomunicación (ETSIT) de la Universidad Politécnica de Cartagena (UPCT).

Capítulo 2
Conversaciones de Voz en Redes de Datos

2.1 Introducción

Desde el punto de vista del transporte de información, se distinguen dos grandes categorías en el patrón de tráfico de los servicios que se pueden ofrecer con redes de conmutación de paquetes:

- Tráfico con requisitos de tiempo real (voz sobre IP, video, tráfico multimedia), también denominado tráfico de tipo *stream* (corrientes de tráfico)
- Tráfico tolerante a retardos (generado sobre todo por aplicaciones de datos), también llamado tráfico elástico, debido a la tolerancia que admite en las prestaciones que ofrece la red.

Los criterios de calidad de uno y otro modelo divergen, atendiendo a la finalidad de los usuarios:

- Los usuarios (emisor/receptor) de una corriente de tráfico buscan establecer una comunicación en tiempo real entre ellos; luego la clave de una buena calidad de servicio consiste en mantener en la red los medios para transmitir una cierta información según se va generando.
- Los usuarios de una aplicación que genera tráfico elástico buscan transmitir un lote de información ya cerrado desde el origen, de forma íntegra (es decir, sin errores), y en un tiempo razonable. La integridad de los datos se consigue mediante los mecanismos de protección frente a errores de los protocolos de transporte (típicamente *TCP*). Por tanto la clave de una buena calidad de servicio consiste, en este caso, en lograr que el retardo total en transmitir todo el lote (no cada paquete) no sea excesivo, siendo el parámetro relevante el retardo medio de los paquetes, factor que determina el retardo total de la transacción.

Tomadas en cuenta estas consideraciones se debe denotar que para el tráfico no elástico se presentan una serie de dificultades, que se derivan de la necesidad de ciertos mecanismos de calidad de servicio.

Los usuarios esperan una calidad de voz análoga a la que obtienen en sus llamadas a través de la red analógica, pese a que la información de su voz viaje a través de una red de conmutación de paquetes, luego la calidad de voz percibida será el estándar sobre el que se medirá la calidad del servicio de telefonía *IP*, basándose en la claridad de la voz en la llamada, que es afectada negativamente por:

- Retardo.
- Distorsión.
- Compensación de pérdida de paquetes.
- Compensación del eco.

De los factores anteriormente expuestos ha de tenerse en cuenta que problemas como la distorsión o la compensación del eco ya existían en las redes de transporte analógicas, donde los terminales incluyen filtros y canceladores de eco para mitigar sus efectos.

En cambio, el retardo y la compensación por pérdida de paquetes son naturales de las redes de conmutación de paquetes (derivados de la los problemas de congestión que surgen en éstas), a causa de la naturaleza de medio compartido en las que se basan, y afectan por igual a todas las unidades de transporte que circulan por las mismas.

La solución estos problemas de forma simultánea es complicada, principalmente porque la solución de algunos repercute en un empeoramiento en otros. Sin embargo, existen una serie de márgenes entre los cuales es admisible que se sitúen estos factores y que dan una cierta flexibilidad cara a la implantación de servicios de tráfico en tiempo real.

Por ello se han desarrollado diferentes mecanismos de calidad de servicio (*Quality of Service - QoS*) que ayudan a minimizar el impacto de estos problemas en los datagramas de voz que atraviesan la red, a costa de empeorar las características del transporte de los paquetes de datos convencionales, que no tienen requisitos de tiempo real.

Un estudio más detallado de las dificultades que surgen en la implementación de un servicio de telefonía *IP* utilizando como infraestructura de transporte una red de conmutación de paquetes se realiza a continuación.

2.1.1 Retardo

El retardo es el tiempo que transcurre desde que la voz de un usuario es captada por el micrófono del terminal por el que éste se comunica hasta que llega al altavoz del terminal del extremo remoto.

Los retrasos causan dos problemas: eco y solapamiento.

- El eco es causado por la reflexión de la voz desde el extremo remoto hacia el hablante y es un problema significativo cuando del retraso total de toda la red supera los 50 milisegundos.
- El solapamiento entre la voz de un hablante y la conversación del otro es un problema significativo si el retraso en una dirección llega a ser mayor de 250 milisegundos.

Se distinguen principalmente tres tipos de fuentes de retardo:

- **Retardo algorítmico**: este retardo es causado por la toma y procesamiento de las muestras de sonido que debe hacer el codificador de la voz y se puede definir como el tiempo que se tarda en colocar un *bit* o *byte* en una interfaz. Se relaciona con el tipo de codificación usado y varía desde el tiempo de una muestra simple hasta varios milisegundos. La siguiente tabla muestra los requisitos de ancho de banda para los *códecs* de audio típicos:

Códec	Codificación	Ancho de banda	Tamaño de la Muestra	Ancho de Banda IP típico
G.711	PCM	64 kbps	0.125 ms	80 kbps
G.723.1	ACELP	5,6 kbps	30 ms	16,27 kbps
G.723.1	ACELP	6,4 kbps	30 ms	17,07 kbps
G.726	ADPCM	32 kbps	0,125 ms	48 kbps
G.728	LD-CELP	16 kbps	0,625 ms	32 kbps
G.729	CS – ACELP	8 kbps	10 ms	24 kbps

Tabla 2-1: Tiempos de trama asociados a los distintos *códecs*

- **Retardo de procesado:** es causado por el proceso de toma y codificación las muestras en paquetes para su transmisión sobre la red.
- **Retardo de red:** es producido en función del medio físico (retardo de propagación), los protocolos utilizados para transmitir los datos de voz, la capacidad de los enlaces de la red y el procesamiento que ocurre en la misma (retraso en la gestión de colas). El retardo total acumulado de red puede ser parte significante del total si las variaciones del mismo llegan a ser mayores de 70-100 ms en redes *IP*, si bien la *ITU* (*International Telecommunications Union*) define un retardo admisible entre 0 y 150 ms [2].

Fuente de Retardo	Valores Máximos Admisibles
Retardo Algorítmico	30 ms
Retardo de Procesado	Se considera despreciable
Retardo de Red	70 – 100 ms
Retardo Total Admisible	150 ms

Tabla 2-2: Valores máximos permisibles de los diferentes retardos

2.1.2 Distorsión

En la literatura también se conoce como fluctuación de fase y se da debido a que los paquetes tardan tiempos distintos en atravesar la red. El remitente genera una serie de datagramas de forma regular pero éstos pueden sufrir distintos retardos en la red de paquetes y no llegar con el mismo intervalo de fase a la estación receptora. Eliminar este efecto requiere recoger los datagramas y mantenerlos el tiempo suficiente para permitir a los más lentos llegar y ser colocados en su posición correcta dentro de la secuencia, función que realizan los *búferes* de distorsión y que causa un retardo adicional. Una figura explicativa es la que se muestra a continuación:

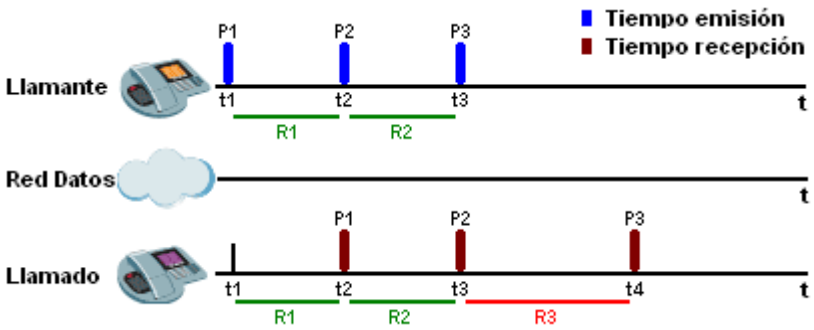

Figura 2-1: Variación del tiempo entre llegadas de un paquete

En la ilustración anterior se observa cómo el llamante genera en t1, t2 y t3 un paquete de voz, siendo el tiempo que hay entre la generación de dos paquetes consecutivos uniforme (R1 = R2).

Una vez generado el datagrama se transmite a la red de datos y circula por ella hasta que es entregado al extremo llamado. Como se puede observar, los paquetes P1 y P2 tardan el mismo tiempo en recorrer la red (R1 = R2), en cambio, el retraso sufrido por P3 (R3) provoca la aparición del fenómeno de la distorsión.

El compromiso entre la eliminación de este factor y el mantenimiento de unas cotas de retardo aceptables ha motivado la aparición de varios esquemas de adaptación del tamaño de estas memorias, cuyo fin es minimizar su tamaño y el retardo asociado. Existen básicamente dos aproximaciones para realizar dicha acción que dependerán del tipo de red que deban atravesar los paquetes.

- La primera aproximación es medir la variación de la distorsión de los paquetes en un periodo de tiempo e incrementar y adaptar el tamaño del *buffer* para igualarlo al calculado. Esta aproximación trabaja mejor con redes que proveen una actuación consistente sobre el tiempo, como redes *ATM*.
- La segunda aproximación es contar el número de paquetes que llegan tarde y crear una relación entre éstos y los procesados con éxito. Esta relación es entonces usada para ajustar los búferes de distorsión con el fin de designar una tasa de retardo permisible. Este acercamiento trabaja mejor en redes con intervalos de llegada entre paquetes muy variables, como ocurre en redes *IP*.

Además de las técnicas descritas en los puntos superiores, la red debe estar configurada y dirigida a proveer el mínimo retardo y distorsión, habilitando una Calidad de Servicio consistente.

2.1.3 Pérdida de Paquetes

En las redes de datos la pérdida de paquetes es un hecho frecuente y esperado, siendo un inconveniente muy grave dependiendo del tipo de red usada (en este caso *IP*). En éstas topologías los datagramas de voz son tratados como datos; de forma que en partes de la red con alta congestión todos serán descartados de igual modo.

Los paquetes de datos carecen de requisitos de tiempo real y pueden ser retransmitidos; en cambio, las pérdidas de datagramas de voz no pueden ser tratadas de esta manera, lo que ha obligado a la definición una serie de algoritmos que minimicen este problema, los principales son:

- Interpolar el paquete perdido por el último recibido durante el intervalo de tiempo en el cual se supone que éste debería ser mostrado. Este esquema únicamente llena el tiempo entre tramas de habla no contiguas; trabaja bien cuando la incidencia de éstas pérdidas no es frecuente pero no es una solución satisfactoria cuando el número de datagramas contiguos perdidos es grande o existen pérdidas a ráfagas.
- Enviar información redundante a expensas de utilizar una mayor ancho de banda. Básicamente replica y envía información del enésimo paquete de voz en el siguiente, luego corrige exactamente la pérdida. Sin embargo, usa mayor ancho de banda y crea un mayor retardo.
- Una solución híbrida. Usar una codificación de voz que use mucho menos ancho de banda para proveer información redundante en el siguiente datagrama, reduciendo el problema del ancho de banda extra requerido a costa de asumir el incremento del retardo asociado.

Se puede pensar que la pérdida de una unidad de transporte de voz no es importante en una conversación, es más, es casi imperceptible puesto que en ninguno de los *códecs* utilizados el tamaño mayor de trama a codificar es suficiente para que el oído humano perciba la pérdida.

El problema en sí no es perder un único paquete, sino que se produzca una pérdida en cadena. Por consiguiente, y atendiendo al refranero popular que enuncia aquello de “más vale prevenir que curar”, será más apropiado establecer mecanismos que conserven en la red este tipo de contenedores a costa de degradar el servicio de los restantes datagramas que circulan por la misma; visión sobre la que se basan, por otra parte, las técnicas de calidad de servicio (*QoS*).

2.1.4 Compensación de Eco

El eco es generado desde la red telefónica hacia la red de paquetes y se convierte en un problema cuando el tiempo de ida y vuelta es mayor a 50 msec, algo que en las comunicaciones de voz en redes de paquetes se da con frecuencia, lo que lo convierte en un efecto molesto para la conversación. Por ello se usan técnicas canceladoras de eco (definidas, por ejemplo, en los estándares G.165 y G.168 de la *ITU*).

La función de un cancelador de es comparar los datos de voz recibidos de la red de paquetes con los datos de voz transmitidos y realizar un filtrado de forma conveniente. Estos elementos están limitados por la cantidad total de tiempo que esperan a que llegue la palabra reflejada además, resulta muy importante configurar la cantidad apropiada de cancelación de eco cuando se instala el equipamiento de *VoIP*.

Si no se configura suficiente cancelación de eco, éste persistirá en la llamada y seguirá manteniéndose el problema que existía a priori, sin embargo, si se configura en demasía, el cancelador tardará más tiempo en converger y eliminarlo.

2.2 Planificación General del Servicio *VoIP*

Es común establecer una serie de parámetros generales cuando se va a implantar un nuevo servicio, de forma que se puedan determinar los criterios finales sobre los que se ha de construir la infraestructura que le debe dar soporte. Esto desemboca en un proceso previo que reduce tanto el tiempo dedicado a la instalación como las inversiones en equipamiento.

En sistemas de *VoIP* los criterios previos generales a tener en cuenta en una instalación vienen determinados por una serie de factores que se pueden resumir en:

- Servicios que serán ofertados y tipos de terminal de usuario soportados, dependiendo del tipo de cliente.
- Elección del protocolo de señalización.
- Calidad de servicio requerida para asegurar una comunicación satisfactoria.
- Riesgos de seguridad, que deben estar claramente identificados, empleándose técnicas apropiadas para asegurar que los terminales de usuario, en particular, estén protegidos frente a ataques.
- Fiabilidad y disponibilidad.
- Cantidad de ancho de banda disponible en la red de acceso del usuario.
- Sistema de tarificación a adoptar.
- Mecanismos de interconexión.

En los siguientes puntos se expone una explicación resumida de estos factores, siendo algunos de ellos desarrollados con más detenimiento en capítulos posteriores de este documento.

2.2.1 Conjunto de Servicios

La primera cuestión a resolver es la determinación del abanico de posibilidades que se ofrecen al usuario. Para ello se hace necesario distinguir entre los distintos tipos de éstos y los servicios que normalmente requieren:

- Usuarios privados que usan *VoIP* para establecer llamadas a través de Internet. Éstos tienen como objetivo establecer llamadas al menor coste posible, anteponiendo éste a otros como la calidad.
- Soluciones corporativas:
    - A nivel interno, los ingenieros de red establecen los servicios a proporcionar y la calidad de los mismos, dependiendo del tipo red interna desplegada.
    - A nivel de interconexión con el exterior, el servicio de telefonía *IP* es suministrado por un proveedor de datos o telecomunicaciones.

Éstos ofrecen una calidad relativamente alta junto con un amplio abanico de características asociadas a las que se establece un precio.

- Soluciones para redes troncales, normalmente en redes privadas de los propios operadores del servicio telefónico.

Otra parte importante del diseño es la elección del terminal de usuario sobre el que se soporta el servicio. Se distinguen:

- Teléfonos analógicos.
- Teléfonos IP.
- Centralitas telefónicas
- Clientes *software* para PCs, incluyendo aplicaciones basadas en web.

2.2.2 Protocolo de Señalización

Existen numerosos protocolos de señalización diferentes que son aplicables a la telefonía *IP*. Incluyen:

- Protocolos para el control de dispositivos como H.248 (*Megaco*), *MGCP*, *NCS*, etc.
- Protocolos de acceso al servicio como *SIP* o H.323.
- Protocolos de señalización del servicio como *SIP*, *SIP-T*, *BICC*, *CMSS*, etc.

La elección de uno u otro dependerá el conjunto de servicios a ofrecer elegido y el equipamiento disponible.

2.2.3 Calidad de Servicio

Uno de los requisitos clave en la provisión del servicio telefónico a través de redes de paquetes es que la calidad de la voz durante la llamada sea equivalente a la que se obtiene en el servicio analógico.

IP, por su naturaleza, provee un servicio *best-effort* que trata a todos los paquetes del mismo modo, siendo necesario implementar una solución de calidad de servicio (*QoS*) apropiada que permita mejorar las características de la entrega de paquetes de voz. Se distinguen principalmente tres tipos de soluciones:

- Servicios Diferenciados (*DiffServ*)
- Servicios Integrados (*IntServ*)
- *MPLS*

También será importante determinar algún mecanismo que mejore el tratamiento de las unidades de transporte de datos a nivel de enlace *OSI*, puesto que parte del equipamiento por el que circularán no será capaz de atender a mecanismos de prioridad de capa de red.

2.2.4 Técnicas de Seguridad

El servicio de *VoIP* y las redes de nueva generación son mucho más susceptibles a ataques que las redes tradicionales de telefonía, de modo que se han de delimitar tres cuestiones de seguridad clave:

- Denegación de servicio: Impide a los usuarios acceder a los servicios ofrecidos por la red.
- Suplantación del servicio: El atacante quiere acceder a un servicio sin tener permiso para ello.
- Invasión de la privacidad: Los usuarios esperan que sus conversaciones sean privadas sin que ningún otro agente pueda escucharlas (con la excepción de la intercepción determinada por ley).

Esto obliga a desarrollar políticas de seguridad que abarcan, desde la autenticación previa al acceso al servicio, hasta técnicas de cifrado, que añaden retardo a los datagramas de voz.

2.2.5 Fiabilidad y Disponibilidad

En las redes de conmutación de circuitos se establece un criterio de fiabilidad que se denomina de los cinco nueves, que consiste en que la red sólo puede fallar durante cinco minutos cada año, es decir, que funcionará correctamente, el 99,999 % del tiempo. En una red *VoIP* se ha de alcanzar cotas parecidas de fiabilidad, para ello se deben tener en cuenta los siguientes aspectos:

- Tolerancia a fallos.
- Creación de una estructura de encaminamiento estable.
- Redundancia.
- Distribución de carga

El primero de los puntos deberá proveer de una rápida convergencia ante cambios de topología. Las tablas de encaminamiento, por consiguiente, deberán ser simples y experimentar el menor número de cambios posible, evitando bucles, descartes de paquetes y una carga de proceso excesiva.

Otro punto clave reside en la adquisición de equipos que ofrezcan redundancia a la red existente, de forma que sean capaces de manejar tráfico en caso de fallo de la infraestructura principal y que mientras ésta esté activa, permitan balancear la carga que han de soportar los equipos que la forman.

En la siguiente figura se observa un ejemplo de equipamientos redundantes:

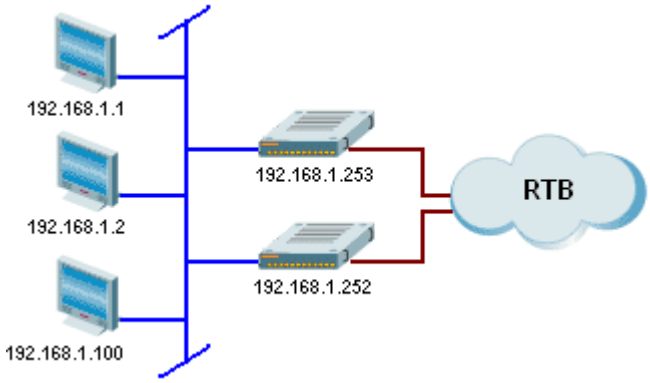

Figura 2-2: Redundancia en el acceso a la red telefónica

En la figura anterior, los *hosts* de la red pueden acceder a la red telefónica convencional a través de dos *gateways* diferentes. Para realizar el oportuno encaminamiento se puede utilizar tanto el protocolo *HSRP* como un elemento de control que aporte la inteligencia necesaria para realizar la tarea, según el protocolo de señalización utilizado.

De este modo, en caso de que el *gateway* 192.168.1.253 esté muy cargado o deje de estar en servicio las llamadas se transferirán a la red analógica a través del 192.168.1.252.

2.2.6 Ancho de Banda

En una red *VoIP*, el tráfico de voz se transporta utilizando el protocolo de transporte en tiempo real (*Real-time Transport Protocol - RTP*). Una muestra de voz típica ocupa menos de 100 *bytes* a los que hay que sumar alrededor de 40 *bytes* más de cabeceras.

Para ahorrar ancho de banda se plantean básicamente tres iniciativas:

- Elegir un *códec* de audio adecuado.
- Aplicar sistemas de detección de voz.
- Aplicar compresión a las cabeceras.

En cuanto a la primera de las premisas, los *códecs* se han desarrollado en dos vías: una enfoca sus esfuerzos en códigos de tipo *PCM* y *ADPCM*, que comprimen la voz explotando las características redundantes de la forma de onda; y la otra, más reciente, emplea procedimientos de procesamiento de señales, que comprimen la señal enviando sólo información paramétrica simplificada sobre la vibración y modulación de la voz original, necesitando menor cantidad de ancho de banda.

El otro gran mecanismo de ahorro en el acho de banda es la detección de actividad de la voz (*Voice Activity Detection - VAD*). En conversaciones normales la comunicación no es realmente bidireccional, sino que primero habla un extremo y después el otro, de forma que se produce un intercambio de información en modo *half-dúplex*.

En redes analógicas convencionales se utiliza un canal en cada sentido perdiendo, por lo menos, la mitad del ancho de banda total. Cuando se utiliza *VoIP* se puede utilizar este ancho de banda para otros propósitos. Para ello se diseñan

mecanismos de *VAD*, que funcionan detectando la magnitud en decibelios y decidiendo cuándo la voz debe dejar de ser entramada. Este sistema padece determinados problemas como la distinción entre el comienzo y el fin de la voz entre la voz del ruido de fondo. Esto significa que si se está en un espacio ruidoso, el sistema puede llegar a ser incapaz de distinguir entre la voz y el ruido de fondo, lo que se conoce como *umbral señal a ruido*.

Comprimir las cabeceras requiere una mayor cantidad de procesado y el mantenimiento del estado de los flujos *RTP*, lo que además incrementa el retardo. Por ello es la técnica menos utilizada de las tres presentadas.

2.2.7 Sistema de Tarificación

La telefonía *IP* debe proveer mecanismos similares a los actuales que permitan a los proveedores generar beneficios. A corto plazo se mantendrá el sistema de tarificación tradicional, el pago por segundos, si bien a largo plazo se prevé que estos métodos se basen en la cantidad de ancho de banda utilizado.

2.2.8 Interconexión

La red de telefonía básica analógica no es una red simple, sino un conjunto de redes operadas por multitud de proveedores. Los acuerdos de interconexión cubren no sólo aspectos relacionados con la unión de redes sino también las divergencias tecnológicas que puedan existir entre éstas (conmutación de paquetes a conmutación de circuitos). Los aspectos más significativos son, entre otros:

- Características de los puntos de interconexión.
- Señalización.
- Tarifas y tarificación.
- Transporte.
- Requisitos regulatorios...

2.3 Consideraciones Específicas de la Red de Datos

La unión de las redes de voz y de datos en una sola red integrada es un reto que requiere un profundo análisis de la infraestructura existente. Por ello, una evaluación objetiva de las capacidades de la red de datos resulta esencial para acometer una implementación de telefonía *IP*, de modo que será interesante:

- Conocer las demandas de tráfico existentes en la red de datos y de voz, hecho que ayudará a determinar los requisitos de calidad de servicio (*QoS*) que se puedan presentar. El estudio de los patrones de tráfico y los picos de utilización en la red de datos determinarán número máximo de llamadas que soporta la misma y el crecimiento potencial de tráfico permisible.
- Detectar y resolver cuestiones de red subyacentes como cuellos de botella en ciertos segmentos de la *LAN* (*Local Area Network*) o

retransmisiones de paquetes de datos frecuentes, clarificará el camino hacia una mejor solución y un encaminamiento más eficiente.

- Conocer la topología de red permitirá el rediseño de ciertas área de la misma para conseguir un mejor rendimiento.

Una vez realizados estos pasos debemos tener en cuenta que el éxito de una implementación del servicio de *VoIP* depende las capacidades de los equipos (*routers, switches,* servidores, *gateways,* etc.) ya existentes en la red de datos, por ello se hace necesario comprobar cada uno de éstos y modificar su configuración (si es posible) para adaptarla a las nuevas necesidades.

En el caso de que exista algún dispositivo cuya configuración no permita adaptarlo para proporcionar el servicio se intentará, en un primer momento, actualizar su *firmware* tomando la decisión de sustituir el aparato como última opción.

2.3.1 El Proceso de Valoración

En ciertos procesos de valoración se inyecta el tráfico de voz en la red de datos y se analizan factores que determinan la calidad con la finalidad de conocer qué mejoras se necesitan en la red para que el servicio de telefonía *IP* funcione adecuadamente. Asimismo también se debe determinar la "salud" de ésta, debido a que será la plataforma de la cual el servicio telefónico dependerá en última instancia.

Por ello habrá que diseñar un completo conjunto de pruebas que ofrezca más información que la mera calidad de la voz, puesto que medidas de este tipo, como el retardo existente, son insuficientes para determinar cuellos de botella en la red, problema que provoca pérdida de paquetes y distorsión. Así, el proceso de implantación de la telefonía *IP* deberá contener los siguientes pasos:

- Reunión de información detallada de la infraestructura de red: tipo de protocolos en uso, políticas y procedimientos, niveles de servicio deseados e inventarios del equipamiento de red afectado. En este punto también se pueden incluir las expectativas de la organización donde se va a instalar el servicio.
- Determinación de la arquitectura física de la red.
- Primeras pruebas de la red.
    - Análisis de varios segmentos de la red con un analizador de protocolos, midiendo variables como picos de utilización, paquetes descartados, retransmisiones, etc.
    - Generación de informes estadísticos y análisis de causas de problemas en la red como, por ejemplo, la congestión o la retransmisión de paquetes.
- Pruebas de calidad de la voz previas.
    - Medición de calidad en la infraestructura existente de voz, por ejemplo centralitas telefónicas y pasarelas de voz, que establecerán las líneas maestras con las que medir las características del tráfico *IP* de voz.
    - Inserción del tráfico telefónico en el flujo de red para medir sus características. Como en el caso de las primeras pruebas de la red, se extraerán informes estadísticos.

- Creación de un documento final que contenga un análisis completo, el veredicto sobre la implantación del servicio, recomendaciones, etc., resaltando las mejoras necesarias en la infraestructura de red para acometer la provisión del servicio telefónico.
- Creación del otro informe final que valore el estado de la red para futuras ampliaciones del servicio y que sirva para determinar si se ha alcanzado el nivel de calidad deseado.

Seguir todo este proceso es importante cara a determinar las características de los productos de telefonía *IP* que se habrán de adquirir, ayudará a tener un mejor conocimiento de los problemas potenciales a los que se enfrenta el ingeniero de red y proporcionará un servicio mejor adaptado a las capacidades disponibles.

Capítulo 3
Arquitecturas y Protocolos de Señalización

3.1 Introducción

En la pasada década, la industria de las telecomunicaciones ha sido testigo de cambios rápidos en la forma en la que las personas y las organizaciones se comunican. Algunos de éstos vienen motivados por el crecimiento explosivo de las aplicaciones basadas en el protocolo *IP* (*Internet Protocol*), superando el tráfico de las redes de paquetes al tradicionalmente asociado a la telefonía analógica convencional.

En el despertar de estos avances tecnológicos quedó claro a los proveedores de telecomunicaciones, compañías y vendedores de equipos que el tráfico de voz era una de las mayores aplicaciones a aplicar sobre *IP*, llegando a la definición del concepto de voz sobre IP (*VoIP*) o telefonía *IP*.

En 1995 aparecieron los primeros productos comerciales de *VoIP* en el mercado. Estos productos fueron el blanco de compañías que deseaban reducir gasto en telecomunicaciones moviendo el tráfico de voz a redes de paquetes. Sin estándares establecidos, la mayoría de las implementaciones fueron basadas en tecnologías propietarias.

Al comenzar a crecer las posibilidades de la tecnología en estas redes surgieron necesidades de interconexión entre soluciones propietarias distintas. Quedó claro entonces que la industria de la voz sobre *IP* necesitaba un conjunto de protocolos estándar que ofreciesen una funcionalidad determinada, resumida en [3]:

- **Traducción de nombre y localización de usuario**. Para establecer la comunicación entre dos puntos es necesario que existan identificadores únicos, como se da con los números de teléfono o las direcciones de correo electrónico, asignados a los usuarios del sistema. Es responsabilidad de la red traducir estos identificadores a una localización geográfica hacia la que dirigir la llamada.
- **Modificación del estado de la llamada**. Se ha de permitir a todas las partes de la comunicación conocer el estado instantáneo de la llamada, incluyendo estados como “llamando”, “activo” y “terminado”. Por ello, el protocolo necesita intercambiar mensajes de forma fiable que puedan informar de los eventos que causan transiciones entre estados, garantizando la convergencia de todas las partes.
- **Intercambio de información multimedia**. Los sistemas finales necesitan ponerse de acuerdo en los tipos de tráfico multimedia a intercambiar (audio, video, texto), qué *códecs* utilizar para cada flujo multimedia y los parámetros para esas codificaciones. Deberá tener en cuenta que los sistemas finales necesitan también intercambiar las direcciones *IP* y los puertos, puesto que la negociación es extremo a extremo.
- **Cambios multimedia**. Debe ser posible ajustar la composición de las sesiones multimedia durante el curso de la llamada, bien porque los

participantes requieren ampliar o reducir la funcionalidad o debido a la aparición o abandono de alguno de ellos.

- **Características pedidas por el usuario**. El protocolo de señalización debe proveer mecanismos a los usuarios para solicitar al sistema servicios avanzados tales como la llamada en espera y el reenvío de la llamada.
- **Características solicitadas por la red**. Algunos servicios de llamada no son explícitamente invocados por los usuarios; sino que son solicitados por el proveedor de servicios durante el progreso de la llamada. Por ejemplo, las características de la monitorización de la llamada puede ser provista por servidores de red.

También pueden ser requeridas otras funciones para obtener un sistema de gestión de comunicaciones completo, si bien éstas quedan fuera del ámbito de la señalización y son específicas de la sesión iniciada.

Quizá el componente más crítico de un protocolo de señalización sea su habilidad para permitir el desarrollo servicios, que pueden ir desde los más simples (retención de llamada, llamada en espera, identificación del llamante, transferencia de llamada) hasta los más complejos (mensajería unificada, aplicaciones de voz interactiva o navegación por voz), donde precisamente se encuentra uno de los beneficios más importante de la migración del servicio de voz a redes de datos.

Varios grupos aceptaron el reto de crear protocolos de señalización que sirviesen tanto para interconectar redes heterogéneas como para dar soporte a las características enunciadas anteriormente. Éstos produjeron resultados independientes, cada uno de ellos con sus características particulares; los protocolos de señalización y control de llamadas para *VoIP* más extendidos son:

- *Media Gateway Control Protocol* (*MGCP*)
- H.248 / *Media Gateway Control* (*MEGACO*)
- H.323
- *Session Initiation Protocol* (*SIP*)

El proceso de desarrollar soluciones de VoIP que tengan un correcto funcionamiento obliga a los ingenieros de red a determinar qué protocolos de los anteriores utilizar dependiendo de cada escenario en particular.

3.2 Arquitecturas Centralizadas y Distribuidas

Uno de los beneficios de la telefonía *IP* es que permite construir redes usando una solución centralizada o distribuida, lo que posibilita a las compañías elegir si la complejidad la sitúan en la red y los terminales son muy simples o si, por el contrario, prefieren desplegar una red simple y proporcionar los servicios a través de terminales más complejos.

En general las arquitecturas centralizadas se asocian con los protocolos *MGCP* y *Megaco* / H.248 (pese a que también se pueden utilizar los protocolos *SIP* y H.323) diseñados para que un sistema maneje de forma centralizada la lógica de conmutación y el control de llamada. A los equipos centralizados se les llama pasarelas de medios

(*media gateways*) y su función es encaminar y transmitir la las cadenas de información de redes *IP* a la RTB, a redes tipo *ATM* y de otros tipos, realizando también la conversión inversa.

En arquitecturas centralizadas la inteligencia de la red está unificada y los extremos son relativamente sencillos, lo que simplifica los flujos de llamada puesto que se replican las características de los equipos de voz tradicionales. Sin embargo también se evita la innovación en los equipos de usuario y se dificulta el desarrollo de nuevos servicios.

Este tipo de soluciones, basadas en una disposición maestro – esclavo son propias de los grandes conmutadores pertenecientes a los operadores del servicio telefónico y su estudio no es objetivo de este proyecto fin de carrera, a pesar de ello se incluye un pequeña reseña informativa en los puntos siguientes.

A las arquitecturas distribuidas se asocian los protocolos *SIP* y H.323, que permiten que la inteligencia de red esté diseminada entre extremos y equipos de control de la llamada. Refiriéndose inteligencia al estado de la llamada, sus características, provisión, tarificación o cualquier otro aspecto del manejo de ésta.

Permitir arquitecturas distribuidas favorece la flexibilidad en la instalación. De este modo las aplicaciones de *VoIP* pueden ser tratadas como otra aplicación *IP* de tipo igual a igual (*peer to peer*), pudiendo añadir inteligencia tanto a los extremos como a los equipos de control de llamadas, todo ello dependiendo del tipo de servicio y los requisitos de la tecnología de la red. Como desventaja se debe apuntar que éstas se apartan del modelo telefónico tradicional, creando soluciones que tienden a ser más complejas.

Si comparamos las características de las soluciones centralizadas y distribuidas obtenemos la siguiente tabla:

	Modelo Centralizado	**Modelo Distribuido**
Operación	• *Gateways* simples • Inteligencia de la aplicación en los servidores de red	• *Gateways* inteligentes • Interacción igual a igual
Desarrollo de Servicios	• Desarrollo basado únicamente en capacidades para los servidores. • Menor tiempo a mercado para nuevos servicios entre diferentes redes	• Desarrollo específico para el equipo • Rápido desarrollo de capacidades para equipos específicos
Despliegue de Servicios	• Sólo actualizaciones en los servidores de control • Los servicios deben estar dirigidos dinámicamente a través de la red	• Se deben actualizar todos los *gateways* para desarrollar nuevas características a través de la red
Coste	• Coste optimizado de las pasarelas • Obsolescencia más lenta del equipamiento	• Mayor coste del conjunto del sistema, sobre todo en sistemas grandes • Los gateways pueden

	del equipamiento	necesitar actualizaciones de hardware en el tiempo
Ejemplos	• *Megaco* / H.248 • *MGCP*	• *SIP* • H.323

Tabla 3-1:Comparación de las soluciones para control del *gateway* centralizadas y distribuidas

Después de la comparativa anterior se aprecia que las arquitecturas de tipo distribuido son las más apropiadas para establecer un sistema de telefonía *IP* partiendo de una red de conmutación de paquetes privada y, por ello, se explican con mayor detenimiento en puntos siguientes de este documento.

3.3 Media Gateway Control Protocol (MGCP)

MGCP es un protocolo que se utiliza para controlar *gateways* telefónicos desde elementos de control de llamada externos denominados agentes de llamada (*call agents*).

Una pasarela (*gateway*) telefónica es un elemento de red que provee una conversión entre señales de audio provenientes de circuitos telefónicos a paquetes de datos sobre Internet o cualquier red de paquetes y viceversa.

En soluciones de voz *MGCP* se separa la inteligencia de control de llamada (fuera del *gateway*) del manejo multimedia, funcionando como un protocolo interno entre componentes separados. *MGCP* entiende que los agentes de control, o agentes de llamada, se sincronizan entre ellos para enviar comandos de forma coherente a los *gateways* bajo su control. De este modo el modelo de comunicación empleado en un intercambio es de tipo maestro / esclavo, de forma que los *gateways* esperan a ejecutar los comandos que les envíen los *call agents*.

Especificando algo más, *MGCP* es un protocolo usado por elementos de control de llamada externos llamados *Media Gateway Controllers* (*MGCs*) para controlar *Media Gateways* (*MGs*). Así, diferentes *gateways* se agrupan lógicamente como si fuesen uno solo. Ejemplos de este tipo de pasarelas incluyen:

- *Gateways* troncales que interconecta la RTB con redes *VoIP*.
- *Gateways* residenciales o de acceso que proveen interfaces analógicos tradiciones (RJ11) o digitales a redes *VoIP*.
- Servidores *IVR* (*Interactive Voice Response*) que proveen de servicios interactivos de voz a redes *VoIP*.

MGCP implementa la interfaz de control con la pasarela como un conjunto de transacciones basadas en un conjunto de comandos con respuesta obligatoria. Se distinguen ocho tipos, clasificados según el sentido de la comunicación:

- De *MGC a MC:*
  - *CreateConnection*: Crea una conexión entre dos extremos; usa *SDP* para definir las características de los puntos participantes.

  - *ModifyConnection*: Modifica las propiedades de la conexión; tiene casi los mismos parámetros que el comando *CreateConnection*.
  - *NotificationRequest*: Solicita al *MC* que envíe notificaciones de las ocurrencias de ciertos eventos en un extremo.
  - *AuditEndpoint*: Determina el estado de un extremo.
  - *AuditConnection:* Devuelve los parámetros relacionados con una conexión.
- Sentido bidireccional:
  - *DeleteConnection*: Finaliza una conexión y recoge estadísticas de la ejecución de la misma.
- De *MC* a *MGC*:
  - *Notify*: Informa de la ocurrencia de ciertos eventos al *MGC*.
  - *RestartInProgress*: Señaliza que un extremo o grupo de ellos entran en servicio o salen de él.

La cabecera de los comandos se compone de:

- Una línea de comando que contiene:
  - Nombre de la petición pedida.
  - El identificador que asocia los comandos y las respuestas.
  - Nombre del extremo que debe ejecutar el comando.
  - Versión de protocolo *MGCP*.
- Un conjunto de líneas de parámetros, compuestas por el nombre del parámetro seguido por su valor.

Todos los campos de la línea de comandos se forman con cadenas de caracteres ASCII imprimibles, separados por caracteres de espacio en blanco (se recomienda usar un único espacio) o tabulaciones.

Para evitar problemas en la utilización de indentificadores, éstos no se pueden reutilizar antes de que hayan transcurrido 3 minutos desde la última transacción completada con dicho identificador.

Este protocolo fue originalmente una propuesta para *Megaco/H.248* y se encuentra descrito en un *RFC* debido a que algunos fabricantes ya habían comenzado a aplicarlo a sus productos y el *IETF* decidió completarlo. Cabe destacar que no es representativo de la dirección hacia la que se mueve la industria y no es, de hecho, un verdadero estándar abierto, ofreciendo una reducida interoperabilidad entre vendedores.

3.4 MEGACO / H.248

Este protocolo es el resultado del trabajo conjunto del *IETF* y del grupo de estudio 16 de la *ITU-T*, denominado *Megado* por el *IETF* y Recomendación H.248 por la *ITU-T*. *Megaco / H.248* es muy parecido a *MGCP* desde el punto de vista de la arquitectura y de la relación entre el elemento de control (de hecho se basa en él y en *MDCP – Media Device Control Protocol*) y el *gateway*, si bien soporta un rango mayor de redes, tales como *ATM*. Así el *MG* (*Media Gateway*) transforma cadenas provenientes de una red analógica a paquetes o celdas dentro de un protocolo como *RTP* (*Real-Time Transport Protocol*).

Este modelo se basa en dos conceptos fundamentales: las terminaciones y los contextos:

- Las terminaciones identifican flujos de medios o recursos, implementan señales y generan eventos, tienen propiedades y mantienen estadísticas. Pueden ser permanentes o transitorios (efímeros) y se definen en paquetes que se asocian a terminaciones individuales.

  Los paquetes son los mecanismos de extensión primarios en *Megaco* / H.248. Definen nuevos comportamientos de los terminales a través de propiedades adicionales, eventos, señales y estadísticas que pueden definirse basándose en paquetes existentes, lo que permite un desarrollo más rápido y un incremento de la innovación potencial.
- Los contextos se refieren a asociaciones entre colecciones de terminales, definen la comunicación entre terminaciones y actúan como un punto de mezcla.

La estructura de comandos de este protocolo es muy simple. Todos los mensajes están en formato ASN.1 y se usan para manipular terminales, contextos, eventos y señales. La siguiente es la lista de comandos:

- *Add*: Este comando añade una terminación a un contexto. La primera terminación de un contexto se usa para crear dicho contexto.
- *Modify*: Modifica las propiedades, eventos y señales de una terminación.
- *Subtract*: Desconecta una terminación de su contexto y devuelve las estadísticas de la participación de la misma en él. Sustraer la última terminación de un contexto elimina tal contexto.
- *Move*: Cambia una terminación a otro contexto.
- *AuditValue*: devuelve el estado actual de propiedades, eventos, señales y características de las terminaciones.
- *AuditCapabilities*: Devuelve todos los posibles valores de las propiedades de una terminación, eventos y señales disponibles en el *gateway* de medios.
- *Notify*: Se usa para que el *MG* informe al *MGC* de la ocurrencia de eventos.
- *ServiceChange*: Permite al *MG* notificar al *MGC* que una terminación o grupo de ellas está a punto de salir de un servicio o acaba de salir de él,

anunciar su disponibilidad a un *MGC* (registro) ó llevar a una terminación o grupo de ellas dentro de un servicio.

Todos los comandos anteriores son enviados desde el *MGC* al *MG* excepto *Notify* (de *MG* a *MGC*) y *ServiceChange* que puede ser también, como se ha detallado, enviado por el *MG*.

Los comandos entre el *MGC* y el *MG* son fácilmente aglutinados en transacciones, usando reglas de construcción simples y flexibles, de forma que el tráfico asociado a las cabeceras puede ser reducido significativamente. Éstas agrupaciones usan descriptores para asociar elementos relacionados, incrementando la flexibilidad y la habilidad para clarificar desde dónde vienen los datos.

3.5 Recomendación H.323

H.323 es un estándar que especifica los componentes, protocolos y procedimientos que proveen servicios de comunicación multimedia (audio en tiempo real, video y datos) sobre redes de paquetes, incluidas las basadas en *IP*.

H.323 fue especificado por el Grupo de Estudio 16 de la *ITU-T*. La versión 1 de esta recomendación fue aprobada por la *ITU* (*International Telecommunications Union*) en Octubre de 1996, formando parte de la familia de recomendaciones *ITU-T* H.32x, que proveen servicios de comunicación multimedia sobre varios tipos de redes. Su objetivo fue definir un estándar para las comunicaciones multimedia sobre redes que no aseguran calidad de servicio.

El crecimiento de aplicaciones de *VoIP* y telefonía *IP*, donde surgieron nuevos requisitos derivados de la interconexión de redes heterogéneas, obligó a desarrollar la segunda versión de este estándar que fue aceptada en Enero de 1998.

Esta nueva versión permitió interoperar con usuarios de otras tecnologías como RSDI (H.320), *ATM* (H.321) o incluso de la red telefónica básica (RTB que queda definida con H.324). De este modo, un cliente H.323 puede establecer comunicaciones con un teléfono normal de la RTB, lo que permite a las empresas a migrar de tecnologías basadas en redes de conmutación de circuitos a redes de paquetes.

No se debe obviar que H.323 no es un protocolo por sí mismo sino que define cómo usar otros protocolos de una manera específica para crear un servicio, especificando además los componentes y procedimientos.

Como logros principales de esta recomendación podemos señalar:

- La estandarización de los protocolos permite a los diversos fabricantes evolucionar en conjunto.
- Los usuarios no deben preocuparse sobre las posibilidades de su interlocutor, existiendo una negociación de las capacidades de cada punto de la línea.
- Debido a su apoyo sobre *IP* es independiente del tipo de red física que lo soporta, permitiendo la integración con las grandes redes actuales.
- Por su propia estructura, es independiente del *hardware*, pudiendo ser implementado tanto en los ordenadores actuales o en equipos específicos, como teléfonos *IP* y consolas de videoconferencia.

- Permite realizar control de acceso dentro de la red. Este sistema se diversifica en dos vías, la primera de ellas impide establecer una conexión entre dos usuarios si los recursos de la red no pueden asegurar que la misma se vaya a realizar de una forma óptima y sin perjudicar al resto de conversaciones que ya se estén produciendo. La segunda restringe el acceso a un usuario si éste no tiene los permisos necesarios para establecer la conexión, independientemente del estado en el que se encuentre la red en dicho momento.

La pila de protocolos H.323 está diseñada para operar sobre la capa de transporte de una red subyacente, por tanto, puede ser usada sobre cualquier tecnología de conmutación de paquetes.

También abarca otros protocolos que permiten conferencias multipunto y comunicaciones no sólo audio sino también de video y datos, con lo cual se produce una integración real de los servicios, permitiendo diferentes modelos tales como correo electrónico, correo de voz, fax o videoconferencia desde un mismo entorno.

La norma H.323 se basa principalmente de siete componentes: codificadores de audio, codificadores de video, registro H.225.0, admisión y estado (*RAS*); señalización de llamada H.255.0; control de señalización H.245; *Real Time Protocol* (*RTP*); y *Real Time Control Protocol* (*RTCP*) que se verán con más detalle en siguientes puntos de este capítulo.

3.5.1 Arquitectura H.323

En una implementación general de H.323 se requieren elementos diferentes: los terminales, los *gateways* o pasarelas, los *gatekeepers* y las unidades de control multipunto (*MCU, Multipoint Control Unit*). Su distribución se representa en la siguiente imagen:

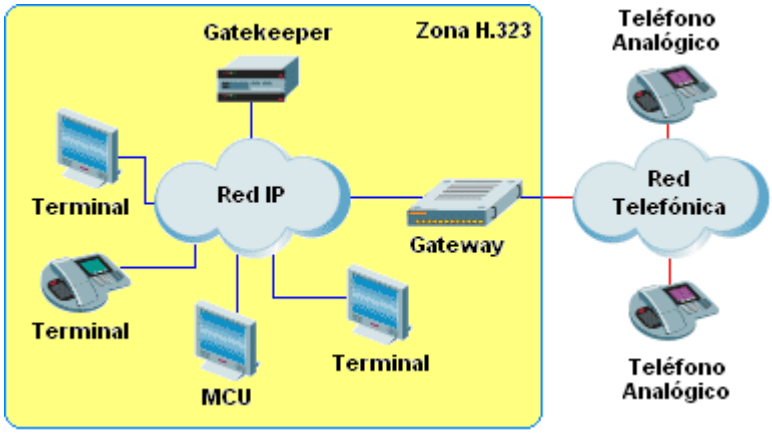

Figura 3-1: Imagen de la Arquitectura H.323

También existe un quinto elemento, el servidor *proxy* H.323. El funcionamiento general de cada uno de ellos se describe en los siguientes apartados.

3.5.1.1 Terminales

Un terminal, cliente o punto final es un extremo donde se originan y finalizan tanto las secuencias de datos como la señalización y su única limitación es que sea conforme con la recomendación. Este elemento debe incluir el tratamiento de la señal para su envío por la red de datos; realizando la captación, digitalización y compresión de la voz.

Para realizar sus funciones debe tener una serie de componentes obligatorios, entre los que se encuentra una unidad de control que implemente el intercambio de capacidades con otros extremos y se encargue de la señalización, *códecs* de audio para empaquetar la voz y una interfaz de la red de paquetes a la que se desea conectar.

Los terminales pueden ser de tipo *software* o *hardware*. Tanto la apariencia como la funcionalidad de cara al usuario de los clientes *hardware* es igual a los teléfonos actuales, con las limitaciones que esto supone en cuanto al desarrollo de nuevos servicios.

Por otro lado las aplicaciones *software* suponen un cambio en la percepción telefónica del usuario, puesto que se ejecutan sobre un ordenador personal; pero ofrecen unas capacidades potenciales muy superiores, permitiendo la aparición de soluciones de muy diverso tipo. Entre éstas se pueden destacan:

- Agenda compartida y personal enlazada a sistemas estándar como *LDAP*.
- Buzón de voz de características avanzadas.
- Manejo remoto del propio equipo con realización de tareas automáticas.
- Organización de llamadas.
- Rellamada automática.
- Funciones de reconocimiento de voz.

En la mayoría de los casos estos terminales soportan, como mínimo, comunicaciones de audio, si bien también pueden dar soporte a comunicaciones de video, siempre y cuando contengan las funcionalidades asociadas pertinentes (como *códecs* de video); a transacciones de datos y a aplicaciones de pizarra compartida.

3.5.1.2 Gateway

La recomendación H.323 lo define como un componente opcional, sin embargo es necesario cuando se quiere establecer una comunicación entre redes heterogéneas, ya que actúa como interfaz entre ellas. A través de estos componentes es posible la operación entre terminales H.323 y otros de tipo RSDI (H.320), *ATM* (H.321) o RTB (H.324) como se ha expuesto en líneas anteriores.

Este dispositivo es el responsable de aceptar peticiones de llamada entre la red H.323 y la red no-H.323 y realizar las conversiones necesarias para que la conversación se pueda llevar a cabo. Las tareas que realiza comprenden la conversión del formato de datos, traducción de la señalización de control, de los códigos de audio y video y funcionalidades asociadas al establecimiento y liberación de la comunicación en ambos extremos de la red. Además, estas funciones las realiza de forma transparente para el usuario en ambos sentidos, pudiendo tanto recibir como emitir llamadas sin ningún problema.

Dependiendo del tipo de red a la cual debe realizar la conversión se requiere un soporte distinto (RDSI, *ATM* ó RTB), por lo que sus características serán las propias de éstas redes. Por ello se considera un terminal dentro de las redes que interconecta, debido a que la señalización que maneja cara a cada tipo de tecnología es la misma que intercambiaría un terminal común.

Para comprender mejor el funcionamiento de este tipo de equipos se procede a establecer un ejemplo. En la imagen inferior, el *terminal 1* establece una comunicación con un teléfono analógico convencional de la red telefónica básica. En este ejemplo el *gateway* deberá realizar las funciones básicas relacionadas con las tres etapas que se producen en una conexión:

- Establecimiento de la conexión: se ha de encargar de generar o traducir la señalización necesaria que haya sido generada por el *Terminal 1* para establecer la comunicación con el teléfono analógico.
- Mantenimiento de la conexión: gestionará la conversión de los distintos paquetes a voz analógica y viceversa, teniendo en cuenta factores como la codificación del audio.

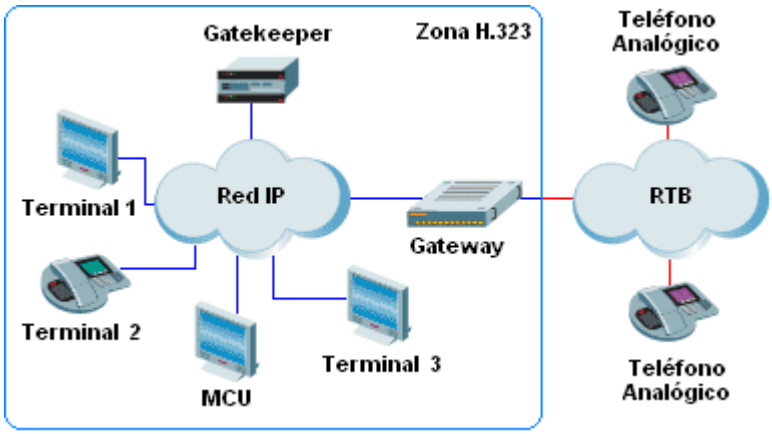

Figura 3-2: Llamada H.323 entre una red IP y la RTB

- Liberación de la conexión: se vuelve tanto a traducir la señalización generada dentro de la red de paquetes como a generar la señalización no proporcionada para la correcta liberación de la comunicación de ambos extremos.

3.5.1.3 Gatekeeper

Este elemento también es opcional en la recomendación pero se puede considerar el cerebro de una red H.323. Realiza una serie de tareas básicas sobre todos los extremos que están registrados en él, que incluyen:

- Traducción de direcciones: Un *gatekeeper* mantiene una base de datos que permite efectuar traducciones entre aliases y sus correspondientes números telefónicos (E.164) o direcciones de red para poder alcanzar el terminal de destino.

- Control de admisión y acceso: Se puede basar en la disponibilidad de ancho de banda, limitaciones en el número simultáneo de llamadas o el registro de privilegios de los extremos. Se hace utilizando mensajes de señalización *RAS*.
- Administración del ancho de banda: Los administradores de red pueden gestionar el ancho de banda especificando limitaciones en el número de conversaciones simultáneas y limitando la realización de llamadas a terminales específicos en determinadas franjas horarias.
- Administración de zona. Se define como zona H.323 al conjunto de terminales, *gateways* y *MCU*s registrados y dirigidos por un mismo *gatekeeper* de forma que el *gatekeeper* provee las funciones anteriores para todos los elementos de la arquitectura que se encuentran en su zona de control.

Como capacidades opcionales se definen:

- Señalización de control de llamada. En una conferencia punto a punto, el *gatekeeper* puede procesar mensajes señalización H.225 además de permitir que los extremos intercambien este tipo de mensajes entre ellos mismos.
- Autorización de la llamada, donde este componente decide si aceptar o rechazar la llamada en función de parámetros como la hora del día o la situación del destinatario de la misma.
- Dirección de llamada. El *gatekeeper* podría mantener información acerca de todas las llamadas activas que él puede controlar un su zona, controlando la disponibilidad de ancho de banda y encaminando llamadas al *gateway* apropiado en función de la capacidad disponible. También se podrían desarrollar servicios añadidos tales como los asociados a seguridad, tarificación, correo de voz o desbordamiento de llamadas.

Las capacidades de un *gatekeeper* vendrán dadas, en última instancia, por la implementación *software* que se disponga del mismo.

3.5.1.4 Unidades de control multipunto (*MCU*)

Este componente también es opcional y permite habilitar conferencias entre tres o más extremos, incluso extendiendo una conferencia punto a punto. Para implementar su funcionalidad es necesario un controlador multipunto (*MC*) de forma obligatoria, acompañado de cero o más procesadores multipunto (*MP*).

Llamada y señalización de control son encaminadas a través del *MC* de forma que las capacidades de los extremos son determinadas y los parámetros de la comunicación negociados, centralizando la localización del establecimiento de una llamada multipunto. El *MP* maneja la mezcla, transacción y procesamiento de las cadenas de audio, video y datos distribuyéndolas a los extremos de la conferencia.

Se pueden establecer dos tipos de conexiones en este tipo de conferencias. El primero de ellos representa un modelo centralizado en el cual cada terminal establece una conexión punto a punto con el *MCU*, que es quien determina las capacidades de éstos y envía a cada uno la cadena de mezcla. El segundo representa un modo no

centralizado donde el *MCU* asegura compatibilidad en la comunicación pero las cadenas son transmitidas en modo *multicast* y combinadas en cada terminal.

Atendiendo a su arquitectura se diferencian, a su vez, dos tipos: los basados en un servidor en el cual se instala el *software* adecuado y los que comprenden un *hardware* y un *software* específico para proveer las funcionalidades necesarias. De cualquier modo, la configuración mínima de la que deberán disponer consta de un *MC* y uno o más *MP*.

A pesar de que los *gatekeepers, gateways* y *MCU* son elementos separados lógicamente en el estándar H.323 pueden ser implementados en un mismo dispositivo físico.

3.5.1.5 Servidor *Proxy* H.323

Este dispositivo actúa en la capa de aplicación y puede examinar los paquetes entre dos aplicaciones que se comunican. Sus funciones básicas se destinan a la mejora de la seguridad y la calidad de servicio de las conversaciones.

En términos de calidad de servicio pueden garantizar a terminales que no soporten *RSVP* (*Resource Reservation Protocol*) el establecimiento de conversaciones satisfactorias, debido a que los *proxys* sí que pueden usar este protocolo entre ellos independientemente de las capacidades del terminal, consiguiendo un encaminamiento específico de aplicación.

En términos de seguridad se destaca que sólo el tráfico H.323 pasa a través de este equipamiento, por lo que las políticas establecidas pueden ser definidas en este punto y evitar al *firewall* la sobrecarga que supone el procesamiento de todos los datagramas asociados a comunicaciones de tiempo real.

3.5.2 Pila de Protocolos H.323

H.323 es independiente de la red de paquetes y de los protocolos de transporte sobre los que circula y, de hecho no los especifica. Sin embargo, son necesarios una serie de componentes para completar la funcionalidad de la recomendación H.323 tal y como se enumeran a continuación:

- Códecs de audio.
- Códecs de video.
- H.225 para registro, admisión y estato (*RAS, Registration, Admission and Status*).
- H.225 para señalización de llamada.
- H.245 para señalización de control.
- Protocolo de transferencia en tiempo real (*RTP, Real-time Transfer Protocol*)
- Protocolo de control en tiempo real (*RTCP, Real-time Control Protocol*)

En la siguiente figura se ilustra la pila de protocolos que utiliza H.323:

<table>
<tr><td rowspan="2">H.245</td><td colspan="2">H.225</td><td colspan="2">Flujos de audio / video</td></tr>
<tr><td>Control de llamadas</td><td>RAS</td><td>RTCP</td><td>RTP</td></tr>
<tr><td colspan="2">TCP</td><td colspan="3">UDP</td></tr>
<tr><td colspan="5">IP</td></tr>
<tr><td colspan="5">Capas físicas / datos</td></tr>
</table>

Figura 3-3: Pila del conjunto de protocolos H.323

3.5.2.1 Códecs de Audio

Los *códecs* de audio transforman las señales de voz captadas por los micrófonos de los terminales H.323 en paquetes, utilizando algoritmos de codificación; y decodifican la información de voz binaria recibida en datagramas desde la red datos, para que sea transformada en una señal analógica de voz perceptible por el odio humano. Los terminales H.323 soportan varios tipos de algoritmos de codificación y decodificación, la *ITU* especifica los siguientes:

Códigos de Audio	Título
G.711	Modulación PCM de la voz 64 kbps
G.722	Codificación de audio de 7kHz en 64 kbps
G.723.1	Codificaciones para conversaciones multimedia de tasa dual transmitiendo a 5.3 y 6.3 kbps
G.728	Codificación del habla al 16 kbps usando código de bajo retraso con predicción lineal.
G.729	Codificación del habla a 8 kbps usando estructuras conjugadas de códigos algebraicos de predicción lineal.

Tabla 3-2: *Códecs* de audio recogidos bajo H.323

No es necesario que estén implementados todos los algoritmos en los terminales H.323 pero para establecer comunicaciones entre ellos es básico que tengan definido, al menos, un código en común, que la norma especifica que sea el G.711.

3.5.2.2 Códecs de Video

De manera similar, los *códecs* de video transforman las señales captadas por una cámara en información binaria y decodifican los paquetes recibidos para poder mostrar las imágenes que transportan. La implementación de *códecs* de video es opcional dentro de H.323, si bien los algoritmos se encuentran definidos en las recomendaciones H.261 y H.263. Al igual que en el caso anterior para poder intercambiar video entre dos terminales deben soportar, al menos, una codificación de forma común.

3.5.2.3 H.225 para Registro, Admisión y Estado

H.225.0 *RAS* es un protocolo modelo cliente-servidor que se usa entre los puntos finales y el *gatekeeper* y que define cómo éstos localizan y se registran en el mismo. También define la forma en que el *gatekeeper* localiza a los terminales, admite otros nuevos en la zona y especifica sus permisos de acceso, por lo que resulta necesario la apertura de un canal *RAS* entre el extremo y el *gatekeeper* de forma previa al establecimiento de cualquier otra conexión.

También se deberá conocer la localización del *gatekeeper* para poder contactar con él. El modo de contacto puede ser estático, es decir se configura a los terminales para que siempre conecten con la misma dirección, o se puede obtener a través de procesos de autodescubrimiento como el que se muestra a continuación:

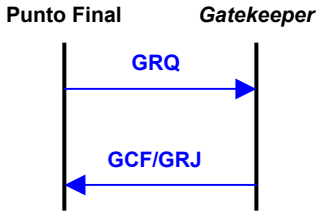

Figura 3-4: Autodescubrimiento del *gatekeeper* H.323

El él, cada punto final envía un mensaje *GRQ* (*Gatekeeper Request*) en modo *multicast* para buscar el *gatekeeper*. Como respuesta a este mensaje el elemento buscado responde con un *GCF* (*Gatekeeper Confirm*), con la dirección *IP* buscada o un *GRJ* (*Gatekeeper Reject*), avisando que no quiere aceptar el registro.

El registro en el *gatekeeper* viene dado por el siguiente intercambio de mensajes:

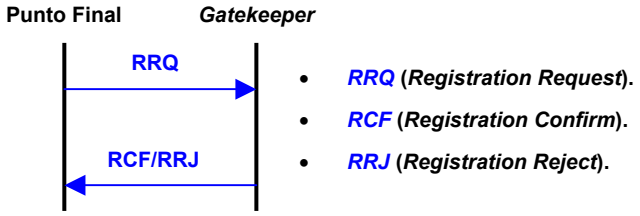

Figura 3-5: Registro en un *gatekeeper* H.323

Mientras que la operación contraria viene dada por:

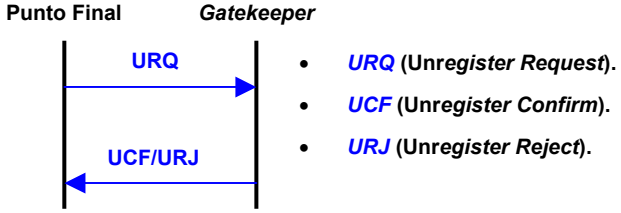

Figura 3-6: Finalización del registro en un *gatekeeper* H.323

Los mensajes de admisión permiten un control férreo del ancho de banda, autorizando o no a los extremos a realizar una llamada. Los mensajes intercambiados se muestran a continuación:

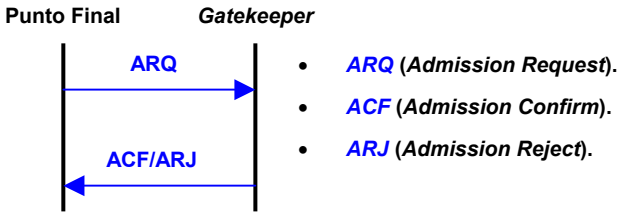

Figura 3-7: Admisión de una llamada H.323

Para el control de estado se utilizan otro conjunto de mensajes, que permiten monitorizar si el punto final está en línea o no debido a una condición de fallo. Este conjunto de mensajes son:

- *Information Request* (*IRQ*). Lo envía el *gatekeeper* para solicitar el estado.
- *Information Request Response* (*IRR*). Se envía desde los puntos finales como respuesta a un *IRQ* o en caso de solicitud de actualizaciones periódicas.
- *Status Enquiry*. Se envía fuera del canal *RAS* y su utilidad es comprobar el estado de la llamada, verificando si ésta sigue activa.

3.5.2.4 H.225 para Señalización de Llamada

H.255.0 es un protocolo usado para el establecimiento de la conexión del control de señalización H.245. Se utiliza para establecer una conexión entre dos extremos H.323 o entre un extremo y el *gatekeeper*. Se consigue intercambiando mensajes Q.931 y Q.932 a través del canal H.225. Los más comunes son:

- *Setup*. Es enviado por el llamante para establecer la conexión con el llamado.
- *Call Proceeding.* Lo envía el llamado hacia el llamante para avisar de que el establecimiento de llamada ha comenzado.
- *Alerting*. Lo envía el llamado para avisar de que el sonido de llamada se ha iniciado.
- *Connect*. Lo envía el llamado indicando que se responde a la llamada.
- *Reselease Complete*. Lo envía el extremo que inicia la desconexión, indicando que la llamada ha sido liberada.
- *Facility*. Se usa para solicitar o confirmar servicios suplementarios.

El intercambio principal se observa en la siguiente figura:

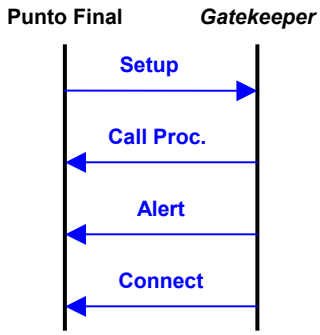

Figura 3-8: Mensajes de señalización en el establecimiento de la llamada

3.5.2.5 H.245 para Señalización de Control

H.245 es un protocolo de intercambio de mensajes de control, negocia tanto las capacidades de audio y video de los terminales como las características de la llamada entre ellos. Transporta información del tipo:

- Intercambio de capacidades
- Apertura y cierre de canales lógicos usados para transportar flujos multimedia.
- Mensajes de control de flujo
- Comandos generales e indicaciones.

Define los siguientes procedimientos y mensajes para realizar su propósito:

- *Capability Exchange*. Para intercambiar las capacidades entre dos extremos, que incluyen los tipos de tráfico soportados, los *códecs* y las velocidades de muestreo.
- *Master-Slave Termination*. Para determinar qué punto final es el principal y cuál es el secundario para una llamada, cuando ambos extremos solicitan acciones similares a la vez.
- *Round-Trip Delay*. Para medir el retardo entre los puntos finales llamante y llamado.
- *Logical Channel Signaling*. Abre y cierra el canal lógico que transporta la información de audio, vídeo y datos.

Mensajes, que pueden ser encapsulados dentro del canal de señalización H.225, evitando la creación de un canal de control H.245 separado. De este modo se mejora el tiempo de conexión de la llamada y la asignación de recursos, teniendo además la posibilidad de conmutar de nuevo a una conexión H.245 separada.

3.5.2.6 Protocolo de Transferencia en Tiempo Real *RTP*

H.323 utiliza también los protocolos *RTP* y *RTCP* del *IETF* (*Internet Engineering Task Force*). Las funciones del *Real-time Transport Protocol* (*RTP*), o protocolo de transporte de tiempo real, son facilitar la transmisión y tránsito de tráfico de tiempo real entre extremos sobre redes de servicios *multicast* o *unicast*. Este tráfico puede ser de audio, video o datos que, por la aplicación a que dan cobertura, sean críticos.

RTP no realiza ninguna reserva y tampoco garantiza calidad de servicio para los datos que transporta. Además aporta identificación del tipo de carga, número de secuencia, tiempo de envío y monitorización de la entrega y se basa en los servicios del protocolo de trasporte *UDP* (*User Datagram Protocol*) para proporcionar multiplexación y *checksum*.

Como última nota, destacar que *RTP* también puede ser utilizado con otros protocolos de transporte distintos a *UDP*.

3.5.2.7 Protocolo de Control en Tiempo Real *RTCP*

RTCP (*Real-time Transport Control Protocol*) es utilizado para proveer un mínimo control sobre la entrega de datos, monitorizando la calidad de la misma a través de información de retroalimentación tanto en el emisor como en el receptor. También es capaz de cubrir funcionalidades de identificación de usuarios de una forma escalable y es independiente de las capas de transporte y red que le son subyacentes.

3.5.2.8 Implementación Mínima

En este apartado se pretende detallar cuáles son los protocolos mínimos que deben soportar los elementos principales de la arquitectura H.323.

Terminales

Los terminales H.323 deben soportar los siguientes protocolos:

- H.245 para intercambiar las capacidades del terminal y crear los canales multimedia.
- H.225 para la señalización y establecimiento de la llamada.
- *RAS* para registro y otros controles de admisión en un *gatekeeper*.
- *RTP*/*RTCP* para transportar la información de audio y video.

Gateways

Un *gateway* conforme a la norma H.323 debe soportar:

- H.245 para el intercambio de capacidades.
- H.225 para señalización, establecimiento y liberación de la llamada.
- H.225 *RAS* para registro con el *gatekeeper*.
- Del lado de la red que no es de paquetes el *gateway* soporta protocolos específicos de ese tipo de red (por ejemplo RDSI ó SS7).

Gatekeepers

Los protocolos que utiliza un *gatekeeper* son:

- H.245 para el intercambio de capacidades.
- H.225 para señalización, establecimiento y liberación de la llamada.
- H.225 *RAS* para registro con el *gatekeeper*.

Para entender mejor la utilidad de los protocolos de señalización anteriores se plantea un ejemplo en el siguiente apartado.

3.5.3 Ejemplo de Funcionamiento

Dos terminales se pueden comunicar directamente sin la utilización de un *gatekeeper* pero las funciones que éste añade, como el control de admisión y la gestión del ancho de banda, son muy interesantes cara al establecimiento de las llamadas y hacen que sea incluido en la gran mayoría de las implementaciones.

El proceso de iniciación de una comunicación en una red H.323 comienza con el registro de los diversos terminales durante el arranque de estos. De este modo, no tenemos ningún problema asociado a la movilidad de los diversos usuarios entre puestos y se puede realizar una traducción adecuada de los identificadores de usuario para obtener su localización física.

El registro en el *gatekeeper* se puede realizar de forma estática o dinámica. En el método estático, se configura el terminal para almacenar estáticamente la dirección *IP* del mismo. A través del método dinámico, el terminal envía al *gatekeeper* un mensaje de petición a una dirección multicast, y el *gatekeeper* que escucha dicha llamada responde la petición con un mensaje de confirmación que contiene su dirección *IP*.

Es responsabilidad del *gatekeeper* monitorizar todo el tráfico generado por las diversas comunicaciones, a efectos de mantener un nivel aceptable de saturación de la red. El control de ancho de banda permite al administrador fijar un límite de utilización, por encima del cual se rechazan las llamadas, bien sean internas o externas.

Otro aspecto importante que debe manejar el *gatekeeper* es el enrutamiento de las llamadas; este dispositivo puede redireccionar éstas al *gateway* más indicado o elegir un nuevo destino si el original no está disponible. Este punto es donde una solución *software* puede dotar al administrador del sistema de herramientas potentes de control y extensión de los mecanismos de funcionamiento.

En cuanto a otras capacidades añadidas, podemos pensar en el control de costes de llamadas, control de centros de atención al cliente, etc. Se pueden presentar dos escenarios diferenciados dependiendo del tipo de llamada [4].

- Llamada Interna: Llamada hacia un terminal situado en la propia red, asumiendo que la señalización de la llamada se intercambia de forma directa entre los terminales:
    - Búsqueda del *gatekeeper* (si se ha configurado por el método dinámico).

- Envío desde el terminal llamante de una petición de admisión para conectar con el equipo que se desee.
- Respuesta del *gatekeeper* confirmando que contiene la dirección *IP* de la máquina receptora de la llamada, o denegando la conexión en el caso de no tener permiso para establecer la llamada o no haber ancho de banda suficiente.
- Negociación del establecimiento de la conexión y los códigos de audio a utilizar durante la misma.
- Conversación de voz.

Figura 3-9: Inicio de una conexión entre dos terminales H.323 dentro de la misma red

El intercambio de mensajes completo que ocurre durante esta transacción se ilustra en la siguiente figura:

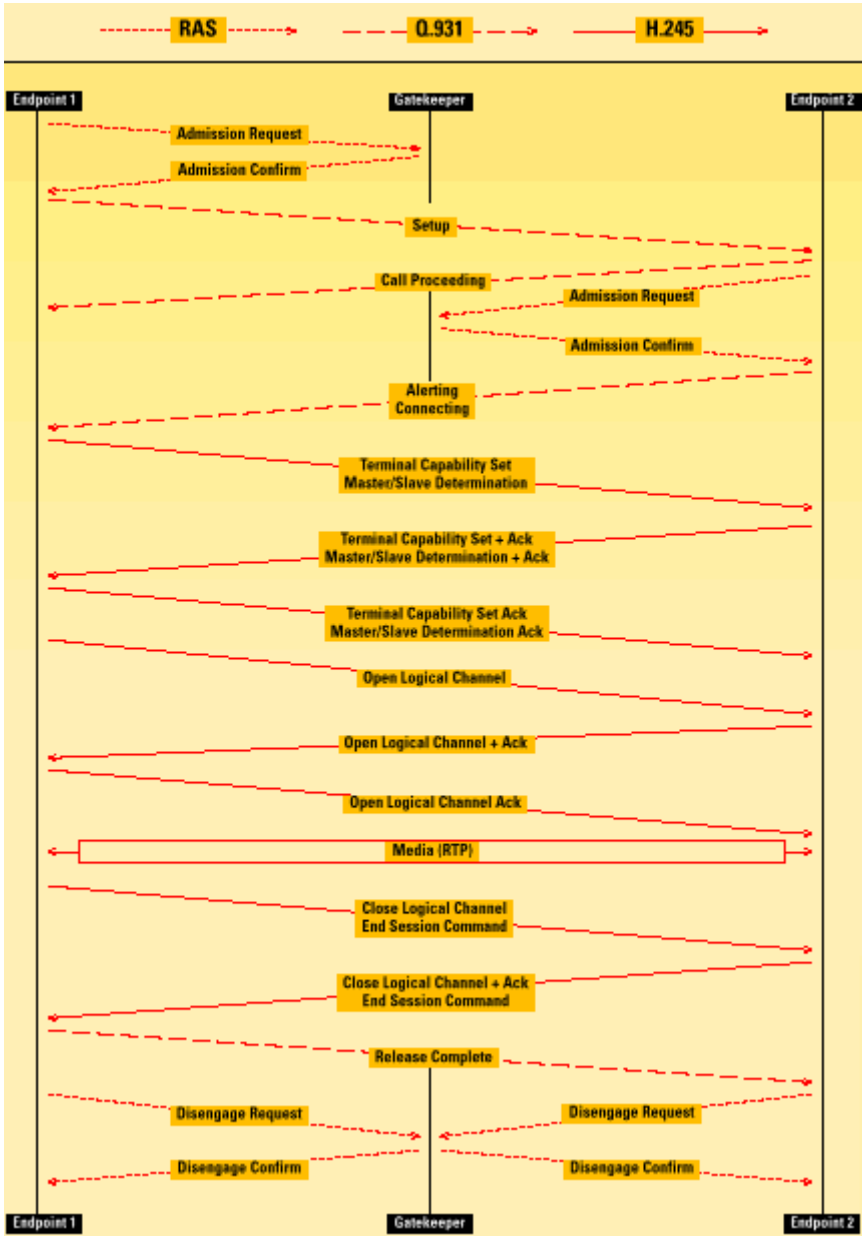

Figura 3-10: Intercambio de mensajes básico para una llamada entre terminales H.323 de la misma zona [5]

- Llamada Externa: Llamada hacia un terminal que se encuentra situado en una red de paquetes distinta a la que se encuentra el llamante.

    - Búsqueda del *gatekeeper* local (si se ha configurado por el método dinámico).
    - El iniciador se pone en contacto con el *gatekeeper*, quien gestiona el control de acceso del terminal llamante y genera la señalización necesaria hacia el terminal remoto hasta que encuentra el segundo *gatekeeper*.
    - Búsqueda del *gatekeeper* remoto por parte del terminal remoto y nuevo intercambio de señalización que sirve al *gatekeeper* local para ponerse en contacto con el *gatekeeper* de la red de destino.

- Intercambio de señalización entre ambos *gatekeepers* con el fin de gestionar el acceso hacia el terminal llamado.
- Intercambio de la señalización necesaria entre ambos terminales y se negocian los parámetros de la comunicación.
- Conversación de voz.

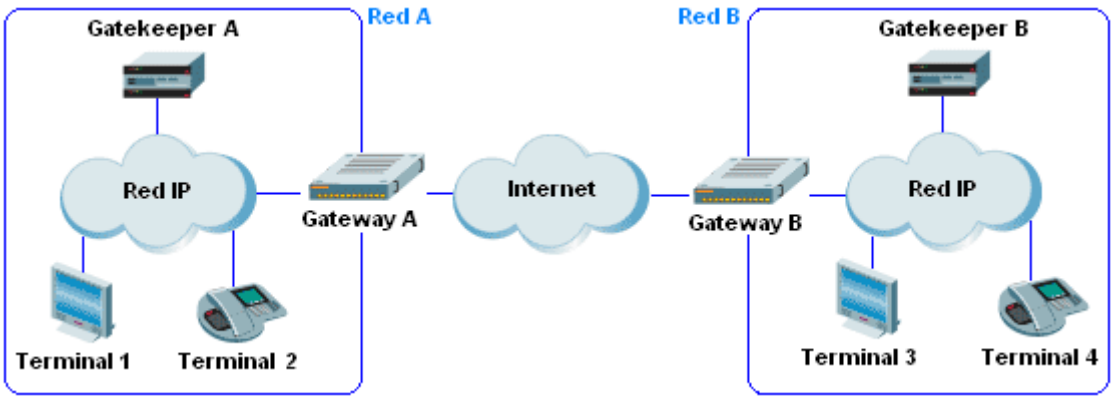

Figura 3-11: Ejemplo de llamada entre terminales H.323 de redes distintas

En el ejemplo anterior se seguirían los pasos descritos a continuación para una conversación entre el *terminal 1* de la red A y el *terminal 3* situación en la red B:

- El *terminal 1* busca su *gatekeeper* local (si se ha configurado por el método dinámico), si el método utilizado es el estático no es necesario este paso.
- De nuevo, el *terminal 1* se pone en contacto con su *gatekeeper* (el *gatekeeper* A) para que éste gestione su control de acceso y le permita comunicarse con el *terminal 3*.
- El *gatekeeper* A envía la señalización adecuada hacia el llamado.
- El equipo llamado busca el *gatekeeper* B e intercambia información con el *gatekeeper* A. Esta información contiene la localización del *gatekeeper* B.
- Se ponen en contacto ambos *gatekeepers* y el de la red del llamada (*gatekeeper B*) realiza el control de acceso hacia el terminal remoto.
- Se intercambia la señalización necesaria entre ambos terminales y se negocian los parámetros de la comunicación.
- Se procede a la conversación de voz.

Para encontrar un mayor grado de detalle en el intercambio de mensajes que se da durante una conversación H.323 se recomienda consultar [4].

3.6 Protocolo de Inicio de Sesión (SIP)

El protocolo de inicio de sesión (*Session Initiation Protocol - SIP*) es un estándar del IETF (*Internet Engineering Task Force*) que fue diseñado, como su nombre indica, como un método para establecer, mantener y terminar sesiones entre usuarios de *Internet* a nivel de aplicación. Ha sido extendido y actualizado a través de nuevos *RFCs* y, hoy día, se utiliza para ofrecer servicios de *VoIP*, video y mensajería instantánea.

Se pueden utilizar invitaciones *SIP* para establecer sesiones y transportar descripciones de la sesión, además soporta comunicaciones *multicast* así como llamadas punto a punto y a multipunto. Las comunicaciones se pueden establecer y terminar utilizando estas cinco facetas *SIP*: localización de usuario, capacidad de usuario, disponibilidad de usuario, configuración de llamada y manejo de llamada.

Además, se fundamenta en un comportamiento petición-respuesta basado en texto plano que trabaja en la capa de aplicación del modelo *OSI* (*Open System Interconnection*) y provee capacidad de:

- Determinar la localización del extremo objetivo de la llamada. *SIP* soporta resolución de direcciones, redirección de llamadas y traducción de nombres.
- Determinar las capacidades multimedia del terminal receptor. A través de *SDP* (*Session Description Protocol*) se busca el nivel de servicios comunes a todos los extremos que permita establecer la comunicación.
- Determinar la disponibilidad de un extremo. Se comprueba que el destino de la llamada está disponible y se devuelve si no lo está, un mensaje indicando cuál es el motivo que impide la llamada.
- Realizar cambios durante la sesión de la llamada, tales como la agregación de más extremos o la modificación de las características de los medios.
- Manejar la transferencia y terminación de las llamadas.

Estas capacidades proporcionan a *SIP* de una serie de atributos que lo convierten en un potente marco de trabajo, y que han sido claves para su desarrollo actual. Éstas se pueden agrupar en:

- Eficiencia: Las funciones de señalización consumen poco ancho de banda en relación con el flujo multimedia debido a que toda la información que se pide para el establecimiento de la llamada se incluye en el mensaje inicial.
- Movilidad: La utilización de *SIP* permite a los usuarios registrarse desde diferentes ubicaciones con una misma dirección (similar a las de correo electrónico); por consiguiente, las llamadas pueden ser dirigidas correctamente a su destinatario independientemente del lugar donde éste se encuentra situado.
- Comunicación entre terminales heterogéneos: El hecho de que *SIP* provea mecanismos de negociación de las características soportadas por los participantes de una llamada (a través del protocolo *SDP*) permite que

distintos tipos de éstos se puedan conectar siempre que tengan, al menos, unas mínimas capacidades multimedia comunes, independientemente del tipo de *códec* que éstos utilicen.

- Simplicidad en el desarrollo de nuevas aplicaciones: La estructura de este protocolo es mucho más fácil de extender para programar nuevas aplicaciones que la de otros existentes equivalentes, principalmente debido a que la codificación de sus mensajes se realiza en modo texto.
- Flexibilidad en el intercambio de paquetes: En *SIP* las rutas de señalización y de tráfico multimedia son totalmente independientes, pudiendo atravesar diferentes equipos de diferentes redes físicas.
- Búsqueda eficiente del llamado: Se basa en criterios de bifurcación donde se asocian múltiples equipos a una sola dirección, de tal modo que éstos pueden ser contactados de forma simultánea o secuencialmente (desbordamiento en programación de centralitas telefónicas), de acuerdo con la política establecida.
- Extensión: *SIP* ha sido concebido como una arquitectura modular y flexible, que permite mejorar las características funcionales y facilita su crecimiento. También provee la posibilidad de retirar las opciones no usadas, lo que evitará que el protocolo se vuelva poco manejable.

Las características, capacidades y extensiones que se han ido desarrollando a lo largo del tiempo han hecho que *SIP* haya crecido hasta el punto de convertirse en el protocolo estándar de intercambio de tráfico multimedia en la tercera generación de telefonía móvil (3G, *UMTS*). Otro hito no menos importante es la adopción por parte de *Microsoft* de este protocolo para el desarrollo de *Microsot Office Live Communication Server 2003* (incluye un servidor *SIP* y una *API* que de la que se espera que aparezcan nuevas aplicaciones) ó *Windows Messenger*.

Con el impacto que tienen estos dos servicios se espera que el número de usuarios *SIP* llegue a ser de decenas de millones de personas y se comience a usar como base en aplicaciones de juegos en tiempo real, llamadas con video, aplicaciones compartidas y monitorización y control domóticos.

La difusión aplicaciones basadas en *SIP* que añaden nuevas capacidades suplementarias a las descritas en los *RFCs* pueden llevar a errores o dudas. Por ello se enuncian a continuación una serie de conceptos, que se pretende queden claros de forma previa a la exposición teórica de las características del protocolo. Éstos son:

- *SIP* es un protocolo de señalización independiente de la capa de transporte; puede funcionar sobre varios tipos protocolos de este nivel como *UDP*, *TCP* y *STCP* (*Stream Transmission Control Protocol*) en tecnologías como *IP*, *ATM* o *X*.25. Únicamente necesita un servicio de entrega datagramas y es independiente de la capa de paquetes.
- *SIP* no especifica ni incluye ningún tipo de capacidad de calidad de servicio, trabaja con otros protocolos para conseguir esta funcionalidad.
- *SIP* es independiente de cualquier protocolo de seguridad y podría ser usado con varios de éstos, como *TLS* (*Transport Layer Security*) e *IPSec*¸ si bien especifica un método de autenticación.
- *SIP* es un protocolo igual a igual, no es un protocolo de control de *gateway* como HGCP o H.248.

- *SIP* provee métodos para controlar las sesiones, pero no especifica las aplicaciones y servicios que usarán estas sesiones; como resultado *SIP* no garantiza un comportamiento específico de una aplicación.
- *SIP* es independiente del conjunto de *códecs* elegidos, ofreciendo la flexibilidad de iniciar sesiones de diferentes tipos.

Los campos de la cabecera de este protocolo se pueden visualizar en [6] y su estructura de mensajes y respuestas será detallada en epígrafes posteriores de este documento.

3.6.1 Arquitectura *SIP*

Existen principalmente dos elementos básicos en la arquitectura del protocolo *SIP*: los agentes de usuario (*SIP User Agent*) y los servidores de red (*SIP Network Server*). Un resumen de su funcionalidad se expresa a continuación:

- El agente de usuario (*UA*). Es un dispositivo *hardware* o *software* que implementa el protocolo *SIP* y es usado por una persona o proceso que actúa en lugar de un usuario. Consiste en dos componentes principales:
  - Agente de usuario cliente (*UAC, User Agent Client*), que inicia las llamadas.
  - Agente de usuario servidor (*UAS, User Agent Server*), que responde las llamadas.
- El servidor de red *SIP* (*SIP Network Server*). Maneja la señalización asociada a múltiples llamadas proveyendo resolución de nombres y localización de usuarios; consiste en tres grandes grupos:
  - Servidor de registro *SIP* (*SIP Register Server*). Básicamente es una base de datos que contiene la traducción de todos los agentes de usuario que hay en un dominio y sus respectivas identidades, nombre y número de traslaciones teniendo acceso además a información de su presencia. Su funcionalidad se basa en la recepción de mensajes de registro de los extremos, que permiten traducir la dirección *SIP* a una ubicación física.
  - Servidor proxy *SIP* (*SIP Proxy Server*). Reenvía los mensajes *SIP* a múltiples servidores proxy, creando un árbol de búsqueda, con el fin de que éstos alcancen su destino. Hay dos modos de funcionamiento de los servidores de este tipo: sin estado (el servidor *olvida* la información una vez que la respuesta es enviada) y con estado (el servidor registra información previa de forma que es capaz de usarla para mejorar la transferencia de mensajes)
  - Servidor de redirección *SIP* (*SIP Redirect Server*). Ayuda a los extremos a encontrar la dirección deseada redirigiendo sus peticiones hacia otros servidores.

Una imagen ilustrativa de la disposición de los equipos de la arquitectura SIP puede verse en la siguiente figura:

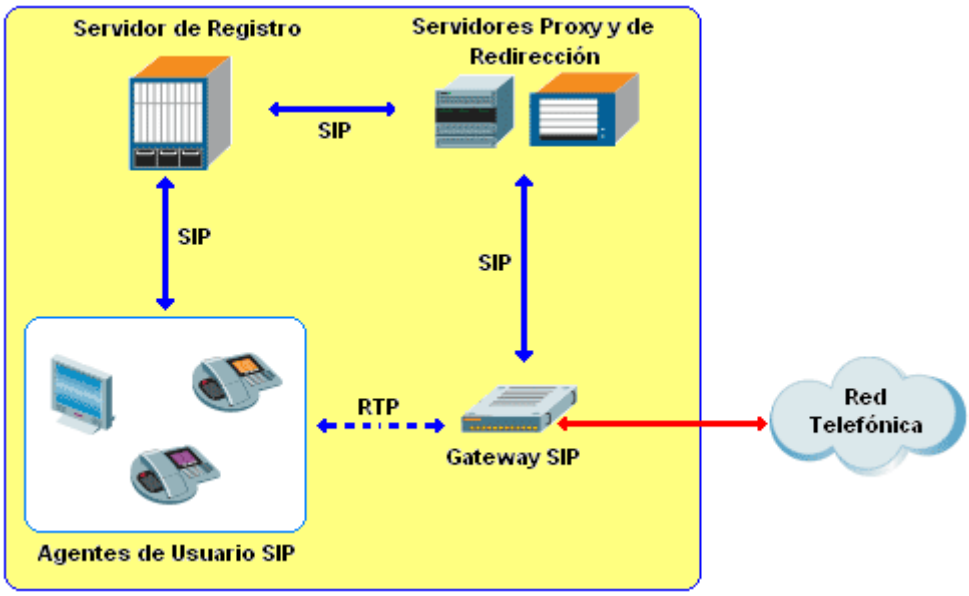

Figura 3-12: Arquitectura *SIP* [7]

La funcionalidad de cada uno de estos elementos se detalla en mayor grado en los siguientes epígrafes.

3.6.1.1 Agentes de usuario

Un agente de usuario representa un sistema final y se puede implementar tanto en un teléfono *IP* como en un *software* de llamadas. Las funciones principales de sus componentes son:

- Un *UAC* (*User Agent Client*) se usa para generar una petición *SIP*, basándose en algún estímulo externo como el clic de botón de un ratón o la señal telefónica analógica, y procesar una respuesta.
- Un *UAS* (*User Agent Server*) recibe las solicitudes enviadas y de genera una respuesta en lugar del usuario. Esta respuesta acepta, rechaza o redirige peticiones en función de alguna entrada del usuario, estímulo externo, resultado de la ejecución de algún programa o algún otro mecanismo.

Cabe destacar que los agentes de usuario contienen la máquina de estados completa *SIP* y que pueden ser usados sin servidores intermedios, además son los únicos elementos de la arquitectura que tienen una dirección *SIP*, que es de la forma *usuario@host*.

Los servidores no suelen tener dirección *SIP* propia y se identifican a partir de sus direcciones *IP* o nombres de dominio *SIP* y los puertos *TCP*/*UDP*. Por defecto, los servidores *SIP* escuchan en los puertos *TCP* y *UDP* 5060, pero pueden utilizar cualquier número de puerto.

Atendiendo de nuevo a la dirección *SIP,* la parte de usuario puede ser un nombre o un número de teléfono mientras que la de *host* vendrá dada por un nombre de dominio o una dirección de red. Si tomamos como ejemplo un usuario llamado Andrés Cano que trabaja en la Universidad Politécnica de Cartagena, cuya extensión

telefónica es 968325952 y tiene como dirección *IP* la 212.128.20.252, su dirección SIP podrá ser:

- sip:andres.cano@upct.es
- sip:968325952@212.128.20.252

Para conocer con mayor detalle el direccionamiento posible aplicado al protocolo *SIP* se recomienda consultar [8].

3.6.1.2 Servidores de red

La función principal de estos elementos es permitir a los extremos *SIP* intercambiar mensajes, registrar la localización del usuario y permitir la movilidad entre redes. Estos servidores permiten definir y aplicar políticas de seguridad y encaminamiento así como autentificar y dirigir las localizaciones de los usuarios.

Con los avances de *SIP*, la lógica de los servidores va complicándose de forma incremental, atendiendo necesidades como tratar con varias topologías de red (como *Internet*, redes celulares o redes de ancho de banda residenciales), aplicar políticas de encaminamiento complejas, implementar mecanismos de seguridad y realizar extensiones del protocolo para ir adaptándolo a las funcionalidades que se requieren.

Con frecuencia, este tipo de servidores han de manejar altas tasas de mensajes, siendo imperativo producir una ejecución en tiempo real escalable, un alto rendimiento y un bajo retardo. Como se ha expresado con anterioridad, la funcionalidad de un servidor *SIP* se comenzó dividiendo en tres unidades lógicas:

- Servidor de registro *SIP* (*SIP Registrar Server*).
- Servidor de redirección *SIP* (*SIP Redirect Server*).
- Servidor Proxy *SIP* (*SIP Proxy Server*).

A éstas se les han unido nuevos modelos a medida que el protocolo ha ido creciendo. Estos componentes son:

- Agente de usuario *back-2-back* (*Back-2-Back User Agent - B2BUA*).
- Servidor de presencia (*Presence Server*).
- Servidor de eventos (*Events Server*).

La separación que se ha hecho anteriormente es lógica, es decir, es una separación para facilitar el entendimiento pero es aceptable tener en un mismo dispositivo características funcionales asociadas a más de uno de los servidores anteriormente definidos. Por ejemplo, un servidor *SIP* podría beneficiarse de tener la capacidad de encaminar algunos mensajes como un servidor *proxy* y redirigir otros como un servidor de redirección.

Servidor de Registro

Un servidor acepta peticiones de registro y sitúa la información que recibe en una base de datos, que sirve para traducir la dirección *SIP* de los agentes de usuario a

una localización dentro del dominio. Además, almacena otros parámetros como identidad, nombre o número de transacciones.

Estas peticiones son generadas por los clientes para establecer o borrar asociaciones entre su dirección externa *SIP* y la dirección o direcciones con la que quieren contactar; pero también pueden ser usadas con el fin de recuperar todas las asociaciones almacenadas para una dirección *SIP* específica.

El servidor de registro procesa peticiones *REGISTER* para un específico conjunto de dominios y usa un servicio de localización para almacenar y recuperar información sobre la localización. Este servicio podría funcionar en una máquina remota y ser contactado usando un protocolo apropiado (como *LDAP*).

Para entender mejor la utilidad del proceso de registro se va a proceder a poner un ejemplo que se asocia a la ilustración de la siguiente página. De este modo, la forma de proceder es la siguiente:

1. **Álvaro** descuelga su teléfono *IP SIP* o abre el *software SIP* de telefonía de su *PC*. Al comenzar el funcionamiento de su agente de usuario, éste emitirá una petición de registro que será dirigida a un servidor que proporcione este servicio.
2. El servidor de registro almacena los datos del cliente que ha iniciado el proceso en un servicio de localización, como un servidor *LDAP* o *Radius*.

Por su parte **Esther** también ha realizado el mismo proceso y decide realizar una llamada a **Álvaro** utilizando su dirección *SIP*: alvaro@upct.es.

3. Al iniciar la llamada se envía una invitación que llega hasta el servidor *proxy*.

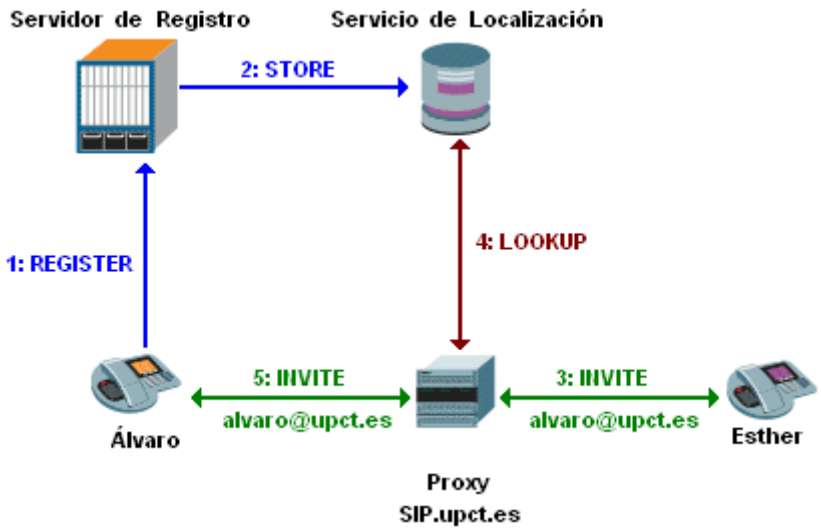

Figura 3-13: Ejemplo de establecimiento de una llamada entre terminales *SIP*

4. Éste busca en el servicio de localización todos los datos disponibles sobre el estado de **Álvaro** en la red, incluida la localización del equipo donde se encuentra.

5. De este modo, terminada la consulta reenvía la invitación de **Esther** a **Álvaro** y ya será éste quien decidirá si atender o no la llamada.

El ejemplo anterior muestra los pasos que se dan durante el inicio de una aplicación de telefonía IP *SIP* y la importancia del registro para el establecimiento de una conexión pero se ha supuesto que el cliente ya tiene configurado su agente de usuario con la dirección y puerto del servidor de registro.

El siguiente punto a tratar es el procedimiento a usar en caso de que el punto final desconozca cuál es la dirección de dicho servidor. En este caso se aprovecha el nuevo tipo de registro de recursos *SRV* en *DNS* que funciona de la siguiente forma:

- El cliente consulta al *DNS* por el registro *SRV* con el tipo de servicio y el nombre de dominio *DNS* perteneciente a la dirección *SIP* propia.
- El servidor *DNS* responde con el nombre del *host* y el puerto *TCP*/*UDP* del servidor *SIP*.
- El agente de usuario pregunta al servidor *DNS* por el registro *A* del nombre del *host* del servidor.
- El servidor *DNS* responde con la dirección *IP* del servidor *SIP*.

El diagrama de intercambio de mensajes puede verse en la siguiente figura:

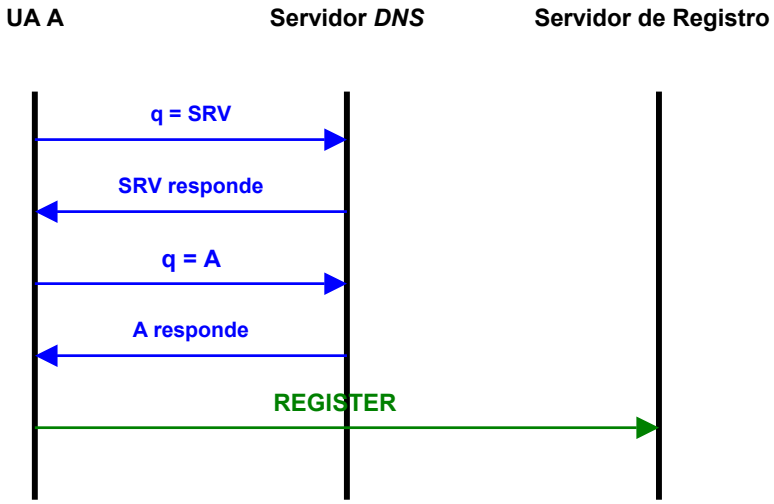

Figura 3-14: Localización del servidor *SIP* a través de *DNS*

La gran ventaja que aporta este sistema es la posibilidad de añadir o eliminar servidores *SIP* sin la necesidad de actualizar cada cliente.

También existen otras soluciones como la utilización de *DHCP* directamente o a través de la opción del servidor *TFTP* de *DHCP*, que hace que los clientes se descarguen un archivo completo de configuración con el nombre de dominio *DNS* (o dirección *IP*), y el puerto adecuado *TCP*/*UDP* del servidor; o realizar el descubrimiento del servidor a través de mensajes *multicast*.

Servidor de redirección

La redirección se define como un procedimiento simple y rápido que permite a los servidores extraer información de encaminamiento para peticiones en respuesta a los clientes, de forma que éstos dispositivos se auto excluyen de la ruta de mensajes de esa misma transacción.

Esta funcionalidad es la más simple de todas. Estos servidores reciben peticiones *SIP* y responden invitando al cliente a contactar con un conjunto de direcciones alternativas, que son devueltas como cabeceras de contacto en el mensaje de respuesta.

Sin embargo, no suelen tener disponible el estado general las conversaciones (llamadas, suscripciones), sólo el de las transacciones individuales que ellos manejan, lo que hace que sean altamente escalables y proporcionen actuaciones de alto rendimiento.

El siguiente ejemplo muestra un escenario de redirección donde se ha obviado el paso de los mensajes a través de los servidores *proxy* con el objetivo de no entorpecer el entendimiento de la operatividad de los servidores de redirección.

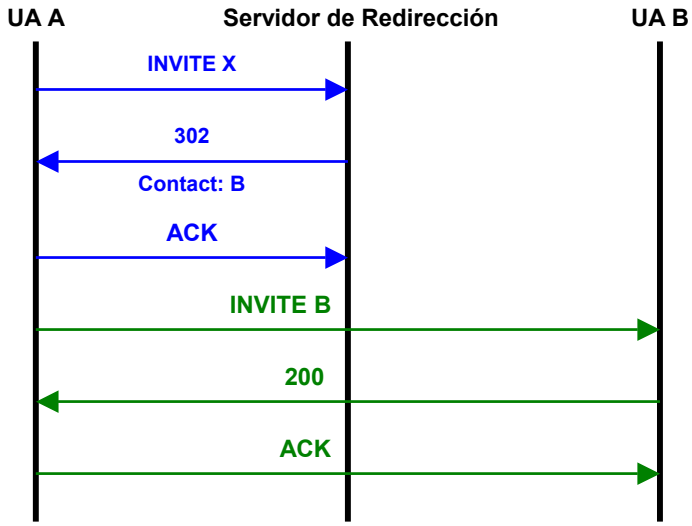

Figura 3-15: Petición de redirección *SIP*

Únicamente destacar que la segunda petición de INVITE se genera con los mismos identificadores de diálogo, conversación, origen y destino que la primera, sólo se diferencia de ésta en el número de secuencia (*Cseq* [6]), que va en formato decimal, y el método pedido.

Por último destacar que el campo *Cseq* ordena transacciones dentro de un diálogo, provee los medios para identificarlas de forma única y permite diferenciar entre nuevas peticiones y peticiones retransmitidas.

Servidor *Proxy*

Estos elementos encaminan peticiones *SIP* a los agentes de usuario servidores (*User Agent Servers - UAS*), que podrían atravesar varios *proxys* en su camino hasta un *UAS*; y respuestas *SIP* a los agentes de usuario clientes (*User Agent Clients - UAC*), que serán encaminadas a través del mismo conjunto de *proxys* atravesados por la petición en sentido inverso.

Cada uno tomará decisiones de encaminamiento, modificando la petición antes de enviarla al siguiente elemento. Podrían verse entonces estos dispositivos como *routers* a nivel *SIP*, sin embargo, su encaminamiento es más complicado que solamente reenviar mensajes basándose en una tabla de encaminamiento. El estándar permite a los *proxys* realizar acciones como validar peticiones, autenticar usuarios, bifurcar respuestas, resolver direcciones, cancelar llamadas pendientes así como detectar y manejar bucles. Esta versatilidad permite a los administradores usar este tipo de equipos para diferentes propósitos y distintas localizaciones en la red creando de políticas de *proxy*.

Los servidores de este tipo se diseñan para ser lo más transparente posible a los agentes de usuario, por ello tienen una limitada capacidad para variar los mensajes. Por ejemplo:

- No pueden modificar el cuerpo *SDP* de un mensaje *INVITE*.
- Exceptuando unas pocas excepciones, no pueden generar peticiones por iniciativa propia.
- No pueden terminar una llamada existente generando una petición de tipo *BYE*.

Además, la especificación *SIP* define dos tipos de *proxys*, que son:

- *Proxys* sin estado. Cuando reciben una petición la procesan y reenvían el mensaje sin guardar ninguna información sobre la transacción, es decir, una vez que el mensaje es reenviado, estos dispositivos *olvidan* haberlo manejado. Se usan en redes troncales, debido a que permiten el manejo de altas tasas de caudal y pueden realizar funciones de balanceo de carga. Por contrapartida este tipo de servidores también tiene desventajas, que se resumen en:
  - No pueden asociar respuestas con peticiones porque no retienen ningún conocimiento de las peticiones reenviadas y tampoco pueden saber si una transacción se realizó correctamente o no.
  - No pueden asociar retransmisiones de peticiones y respuestas con instancias previas de dichos mensajes, por ello procesa las retransmisiones exactamente como si fuera la primera copia del mensaje.
  - Si se pierde el mensaje, el *proxy* no lo retransmite.
- *Proxys* con estado. Estos equipos procesan transacciones en lugar de mensajes individuales, tanto transacciones del servidor (recibir peticiones y devolver respuestas) como del cliente (para enviar peticiones y recibir respuestas); quedando el estado de la transacción y el conjunto de mensajes enviados almacenado. Esto permite un mejor procesado de los mensajes entrantes, por ejemplo, un servidor *proxy* con estado puede identificar retransmisiones de un mensaje y sólo enviarlas en situaciones en las que se requieran y también puede generar él mismo este tipo de mensajes en caso de pérdidas. Sin embargo, también tiene una serie de desventajas asociadas que son:
  - Consumo de memoria. Este tipo de *proxys* necesitan un mayor espacio de memoria por mensaje procesado y por un tiempo

mayor que sus homónimos sin estado. Esto tiene un impacto negativo, tanto en la máxima capacidad del *proxy* como en el número de llamadas o transacciones que puede manejar de forma concurrente.

- Eficiencia. Este tipo de servidores necesitan más ciclos de reloj de la *CPU* para procesar un mensaje por lo que se reduce la velocidad del dispositivo.
- Complejidad de implementación. Estos equipos necesitan una mayor cantidad de elementos lógicos para realizar el procesado de los mensajes. La evolución de *SIP* ha provocado que en ciertos casos se requiera el uso de técnicas especiales de almacenamiento y búsqueda en memoria, que hacen la implementación de estos dispositivos nada trivial y añaden casos especiales que deben ser comprobados.
- Complejidad subyacente de la pila *SIP*. Un *proxy* requiere cierta flexibilidad de la capa *SIP*, especialmente en las capas de transporte y transacción.

Agente de usuario *back-2-back*

Un *B2BUA* es una entidad lógica que recibe una solicitud, la procesa como un agente de usuario servidor (*UAS*) y para determinar cómo esta petición debe ser respondida, actúa como un *UAC* y genera la respuesta.

Esta entidad es parecida en muchos puntos a un servidor *proxy*, pero posee un control más ajustado sobre la conversación y no tiene las limitaciones que define el estándar *SIP* para el *proxy* (un *proxy* no puede desconectar una llamada o alterar mensajes). Mientras maneja mensajes de petición y respuesta, el *B2BUA* reenvía peticiones y respuestas a nivel de conversación de un extremo a otro, como hace en algunos casos un *gatekeeper* H.323. La creación de una conversación a través de un *B2BUA* se muestra en la siguiente figura:

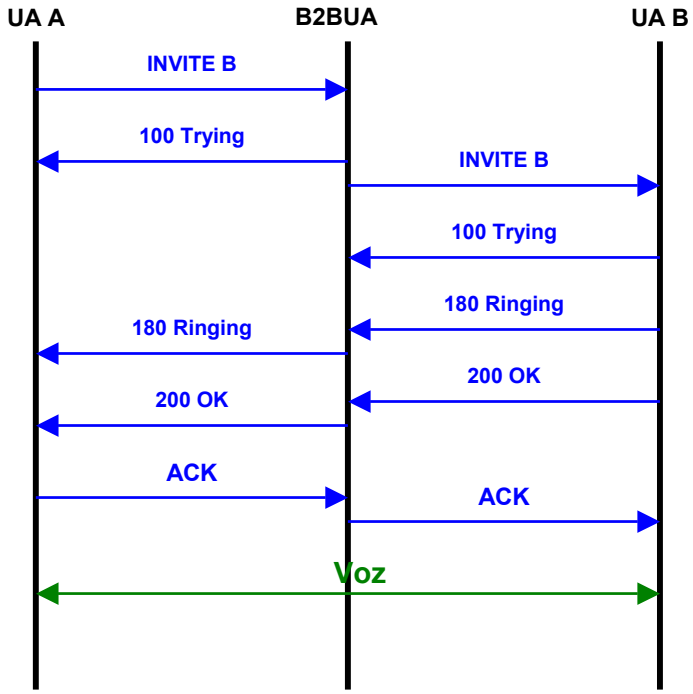

Figura 3-16: Creación de diálogo con un *B2BUA*

El elemento servidor oculta la identidad del iniciador de la llamada y permite la modificación de la cabecera y paquete *SDP* (códecs, IP y puerto, etc), que va dentro del mensaje *SIP INVITE*. Además es preferible que el *B2BUA* permita a la aplicación iniciar acciones, tales como desconectar un lado de la conversación o incluso iniciar una conversación con un extremo.

Este dispositivo es beneficioso en entornos con los siguientes requisitos:

- La red tiene que iniciar o modificar el estado de las transacciones.
- Tener que desconectar llamadas en la red (por ejemplo si el llamante se conecta sobre un sistema prepago) o modificarlas (por ejemplo cambiando la elección de *códecs* en el proceso de establecimiento de una llamada o durante la sesión).
- Conectar una llamada entre dos *UA*s como un elemento de control (*Third Party Call Control – 3PCC*).
- Necesidad de realizar cambios en mensajes que están prohibidos en la implementación de un *proxy SIP* (modificar, añadir, borrar cabecera o el cuerpo, cifrar / descifrar el mensajes o partes de él, comprimirlo, etc.).
- Ocultar los usuarios de una red que está detrás de un servidor *SIP*.
- Control preciso de la llamada, por ejemplo para propósitos de tarificación.

Proveer este control estricto requiere una implementación que mantenga el estado, con las siguientes consecuencias:

- Retiene más información, de modo que tiene una menor capacidad y escalabilidad comparado con un servidor *Proxy*.
- El establecimiento / liberación de la conexión es más complejo.
- Rompe la arquitectura *SIP* extremo a extremo y, por consiguiente *S/MIME* no trabajará correctamente.
- A *B2BUA* es un punto simple de fallo, luego es altamente recomendado añadirlo a un dispositivo de alta disponibilidad, como un servidor *SIP*, porque así completa el conjunto de funcionalidades estándar del servidor.

Servidor de Eventos

Este elemento permite a una entidad suscribirse a notificaciones en el cambio de estado de otras entidades. En este caso el *notificador* es el responsable de recibir la petición de suscripción (*SIP SUBSCRIBE*), validarla y crear un objeto que incluya el nombre del paquete de eventos y el parámetro identificador de la cabecera de eventos. Adicionalmente, también es el responsable de recoger el estado de los recursos y enviar mensajes de notificación (*NOTIFY*) a los suscritos.

Este elemento podría, a su vez, determinar la realización de autenticación y autorización, acción normalmente se recomienda dada la información que se provee. De este modo, después de que la suscripción sea validada se crea un objeto de suscripción y se envía un mensaje *NOTIFY*.

El mensaje podría contener un cuerpo expresando el estado actual de los recursos, de acuerdo al paquete de eventos que sea procesado y a la política de

aplicación. *SIP* usa este mecanismo para la implementación de varias funcionalidades y permite varios paquetes de eventos en sus *RFCs* y *drafts* relacionados. Ejemplos de estos paquetes de eventos son:

- Presencia. Provee información acerca de la voluntad y disponibilidad de un usuario para comunicarse con otros de la red.
- *Winfo*. El conjunto de usuarios suscritos a un recurso particular por un paquete de eventos específico y el estado de sus suscripciones se denomina *Watcher information* (*Winfo*). Esta suscripción también puede relacionarse con el estado de la suscripción de otros usuarios y de su presencia.

Usando este marco de trabajo general en un servidor de eventos, éstos pueden ser construidos para manejar paquetes específicos de eventos, tales como un servidor de presencia.

Servidor de Presencia

La presencia es un servicio que permite a una parte conocer la habilidad y la voluntad de la otra parte para participar en una llamada incluso antes de que se intente realizar la misma. Un usuario interesado en recibir información sobre la presencia de otro puede suscribirse a su estado y recibir notificaciones del sistema de presencia.

El sistema de presencia se compone de los siguientes elementos:

- Uno más agentes de usuario de presencia (*Presence User Agents - PUA*), que recogen la información sobre la presencia en el lado del cliente.
- Uno o más agentes de presencia (*Presence Agents - PA*), que son responsables de:
  - Recibir y manejar suscripciones de presencia.
  - Recibir y agregar datos de presencia de los *PUAs*.
  - Componer los estados de presencia de los fragmentos de datos de presencia.
  - Notificar a todos los usuarios suscritos los cambios de estado.
- *Proxys SIP*. Reenvían peticiones de información de suscripción y notificaciones.
- Servidor de presencia. Es una entidad física que procesa peticiones de suscripción (*SUBSCRIBE*) tanto desde un agente de presencia como de un servidor *proxy*.

La siguiente figura ilustra el intercambio de mensajes en un escenario común del servicio de presencia. En ella se aprecia como múltiples suscriptores reciben actualizaciones del estado de presencia; un cambio de estado o de la presencia obligará a generar una secuencia de mensajes *NOTIFY* al agente de presencia cuyo destino será todos los suscriptores, de acuerdo con la política de la aplicación.

Tanto el suscriptor como el agente de presencia pueden finalizar una suscripción. Un suscriptor puede enviar en mensaje *SUBSCRIBE* con valor cero en el

campo Expires de la cabecera (indica la duración de la suscripción), lo que causa la terminación inmediata de la suscripción. Hecho ésto el agente de presencia notifica el estado más reciente, lo que permite recuperar el estado de presencia sin subscribirse al agente de presencia, en el estándar esta acción se denomina *Fletcher* o *Poller.*

El agente de presencia puede terminar la suscripción enviando una petición con el campo *Subscription-State* indicando ésta ha finalizado.

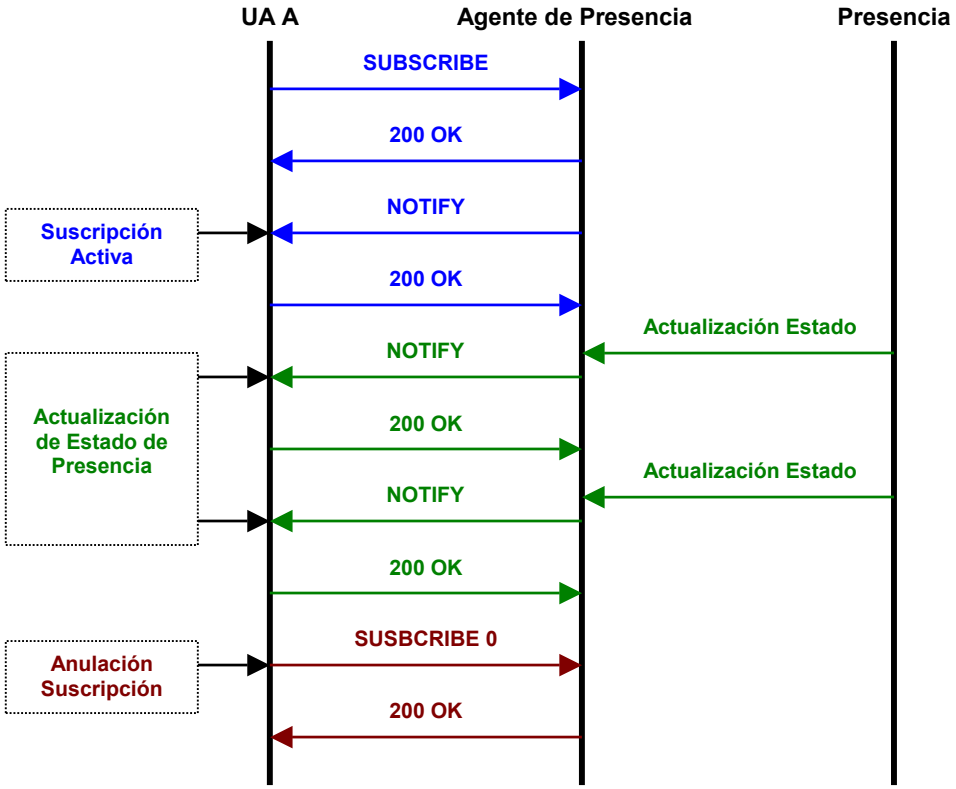

Figura 3-17: Fases de la sesión de suscripción al servicio de presencia *SIP*

3.6.2 Pila de Protocolos SIP

Mientras que en H.323 se utilizaba una serie de protocolos para completar la funcionalidad de estándar, en *SIP* se definen una serie de mensajes a intercambiar en la propia especificación del protocolo, que pueden transportar información de otros protocolos como *SDP* para intercambiar capacidades en los inicios de sesión. La pila de protocolos *SIP* se muestra en la siguiente figura:

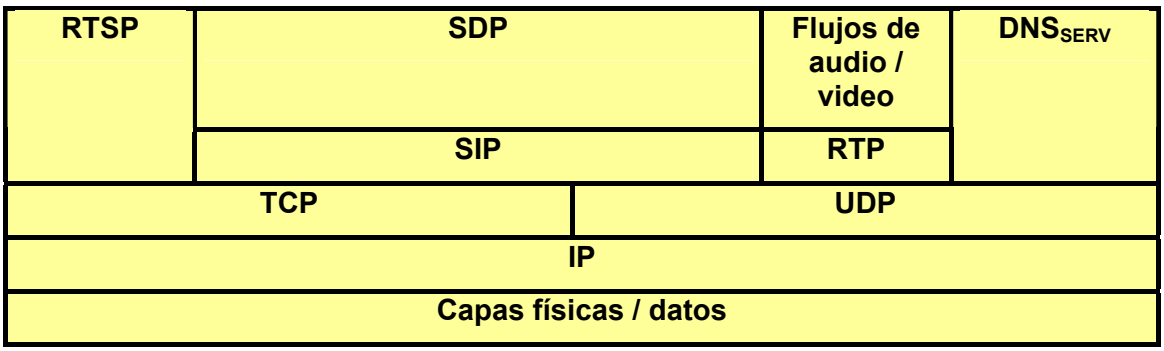

Figura 3-18: Pila del conjunto de protocolos relacionados con *SIP*

La unidad básica de intercambio de este protocolo son los mensajes, cada uno contiene una cabecera que describe los detalles de la comunicación, transportados sobre *TCP* o *UDP*. La estructura básica de un mensaje *SIP* comprende tres partes:

- Línea de inicio
- Cabeceras
- Cuerpo

La línea de inicio indica el propósito del mensaje, las cabeceras proporcionan los detalles del mismo y el cuerpo opcional suministra detalles añadidos que no encajan en las cabeceras. El formato de la línea de inicio depende de si el mensaje es una solicitud o una respuesta:

- *Solicitud*: <Método><Solicitud *URI*><Versión *SIP*>
  - <Método>: Se detallan en el siguiente punto
  - <Solicitud *URI*>: Es el *URL SIP* de la entidad que debe recibir el mensaje.
  - <Versión *SIP*>: Actualmente la *SIP/*2.0
- *Respuesta*: <Versión *SIP*><Código de estado><Frase razón>
  - <Versión *SIP*>: Actualmente la *SIP/*2.0
  - <Código de estado>: Las respuestas a las peticiones vienen codificadas, tal y como se muestra en el siguiente punto.
  - <Frase razón>: Abreviatura del código de estado.

3.6.2.1 Mensajes *SIP*

Las diferentes transacciones definidas en puntos anteriores requieren la presencia de una serie de comandos que permitan establecer la comunicación de forma adecuada entre las distintas entidades *SIP*. Las peticiones especificadas en los primeros *RFCs* fueron seis [9]:

- *INVITE*. Indica al receptor que está siendo invitado a participar en dicha sesión, incluyendo una descripción de la sesión en el cuerpo del mensaje. Si se envía de nuevo durante el transcurso de la llamada (*re-INVITE*) permite variar las características de la sesión.
- *ACK*. Confirma un establecimiento de sesión, sólo puede ser usado en respuesta a un *INVITE*.
- *BYE*. Antes de liberar realmente la llamada, el agente de usuario envía esta petición al servidor indicando su deseo de liberar la conexión.
- *CANCEL*. Se utiliza para abortar cualquier petición que esté en progreso, no afecta a las peticiones terminadas en las que las respuestas finales ya fueron recibidas.

- *OPTIONS*. Se utiliza para consultar y reunir las capacidades de medios mínimas entre los extremos de la comunicación en una fase diferente a la del establecimiento de la sesión. La respuesta debe incluir los métodos *SIP* que soporta
- *REGISTER*. Se utiliza en los clientes para enviar información de localización a los servidores, que realizarán una asociación entre la dirección *SIP* del usuario y su situación física actual.

El hecho de que este protocolo haya sido extendido con el paso del tiempo, de forma que le ha sido añadida funcionalidad, ha causado que el conjunto de los anteriores seis mensajes fuese insuficiente, por lo que se han definido algunos más que se enuncian a continuación:

- *INFO* (*RFC* 2976). Se utiliza para transportar información del nivel de aplicación de llamadas a lo largo de la ruta de señalización *SIP*.
- *MESSAGE* (*RFC* 3428). Es una extensión que permite la transferencia de mensajes instantáneos una vez iniciada la sesión, transportando su contenido en formato *MIME*.
- *SUBSCRIBE* (*RFC* 3265). Se utiliza para indicar el deseo de recibir información de presencia / disponibilidad acerca de algún usuario o evento, como respuesta se recibirán mensajes de tipo *NOTIFY*.
- *NOTIFY* (*RFC* 3265). Se utiliza para proveer información sobre cambios de estado que no están relacionados con una sesión específica. En puntos anteriores se ha visto su aplicación en el servicio de eventos y su utilidad para el servicio de información sobre presencia.
- *REFER* (*RFC* 3515). Se utiliza para transferir llamadas y contactar con recursos externos.
- *PRACK* (*RFC 3262*). *SIP* define respuestas provisionales a solicitudes para proveer información sobre el progreso de la petición que se está procesando. Para garantizar la fiabilidad de estas transacciones los clientes retransmiten sus mensajes provisionales a partir de un algoritmo de *back-off*. Los mensajes *PRACK* (*PRovisional ACKnowledge*) se usan para detener estas retransmisiones.
- *PUBLISH* (*RFC* 3903). Funciona exactamente igual que el mensaje *SUBSCRIBE* anteriormente citado y se utiliza para subscribirse a eventos de otros usuarios.
- *UPDATE* (*RFC* 3311). Este mensaje permite a un cliente actualizar parámetros de una sesión (como *códecs* utilizados), funcionando como un *re-INVITE*. La ventaja de éste mensaje es que permite intercambiar este tipo de información antes de que el *INVITE* inicial sea completado.

Estos mensajes tienen asociadas una serie de respuestas definidas con unos códigos de estado, de forma que puede saber si la llamada ha tenido éxito o ha fallado, incluido el estado del servidor. Estos códigos son:

- 1xx. Provisional (petición recibida, continuando el procesamiento de la petición).

- 2xx. Éxito (acción recibida con éxito, entendida y aceptada).
- 3xx. Redirección.
- 4xx. Fallo en el cliente (la petición contiene una sintaxis errónea y no puede ser cumplida por el servidor).
- 5xx. Fallo en el servidor (el servidor falla a pesar de que la petición es aparentemente correcta).
- 6xx. Fallo global (la petición no puede ser realizada en ningún servidor).

Además de cada uno de los códigos de estado se añade una descripción breve del significado de la respuesta. La información asociada a cada una de éstas puede ser consultada de forma breve en [6].

3.6.2.2 Cabeceras *SIP*

Se distinguen cuatro tipos de cabeceras:

- Cabeceras generales. Aplicadas a peticiones y respuestas.
- Cabeceras de entidad. Definen información sobre el tipo del cuerpo del mensaje y su longitud.
- Cabeceras de petición. Permite que el cliente incluya información de petición adicional.
- Cabeceras de respuesta. Permite que el servidor incluya información de respuesta adicional.

Todos los tipos de cabeceras definidas se pueden consultar en [4].

3.6.2.3 Cuerpo de los mensajes *SIP*

Los contenidos del cuerpo de un mensaje varían dependiendo del tipo de solicitud o respuesta *SIP*. Para las solicitudes, el método *BYE* nunca incluye un cuerpo, mientras que los métodos *INVITE*, *ACK* y *OPTIONS* lo utilizan para codificar los mensajes del protocolo *SDP*.

Para las respuestas, el cuerpo transporta información adicional relativa a la codificación del dada.

3.6.2.4 Protocolo de Descripción de la Sesión

El protocolo de descripción de la sesión (*Session Description Protocol - SDP*), diseñado por el *IETF* se emplea para caracterizar la sesión que se negocia con *SIP*.

Mediante *SDP*, los extremos pueden indicar sus capacidades multimedia y decidir qué flujos multimedia compondrán la sesión, la manera en que se establecerá, los tipos de medios multimedia correspondientes a dichos flujos (audio, video, etc.) y qué *codecs* son soportados y se desean emplear para cada uno, así como la configuración específica de los *codecs* anunciados.

Los mensajes *SDP* incluyen:

- Nombre y tipo de la sesión.

- Tiempo que la sesión está activa.
- Medios de los que consta la sesión.
- Información para recibir los medios (direcciones, puertos, etc).

Información que va transportada en forma de texto usando el conjunto de caracteres ISO 10646 en la codificación UTF-8, dentro de los mensajes *SIP*.

3.6.3 Ejemplo de Funcionamiento

En puntos anteriores, donde se detalla la funcionalidad de los diferentes tipos de servidores que se definen dentro del protocolo *SIP*, ya se han expuesto una serie de ejemplos donde se aprecia el intercambio de mensajes realizado para acceder a cada uno de los servicios.

Sin embargo, no se ha mostrado ningún ejemplo sobre el intercambio de mensajes existente en una comunicación convencional entre dos clientes de una red *SIP*. Para ilustrar este proceso se va a proceder a exponer dos casos típicos donde se separan las fases de establecimiento (azul), mantenimiento (verde) y liberación (rojo) de la conexión.

- Llamada interna. Llamada dentro de un mismo dominio *SIP*. El diagrama es el siguiente:

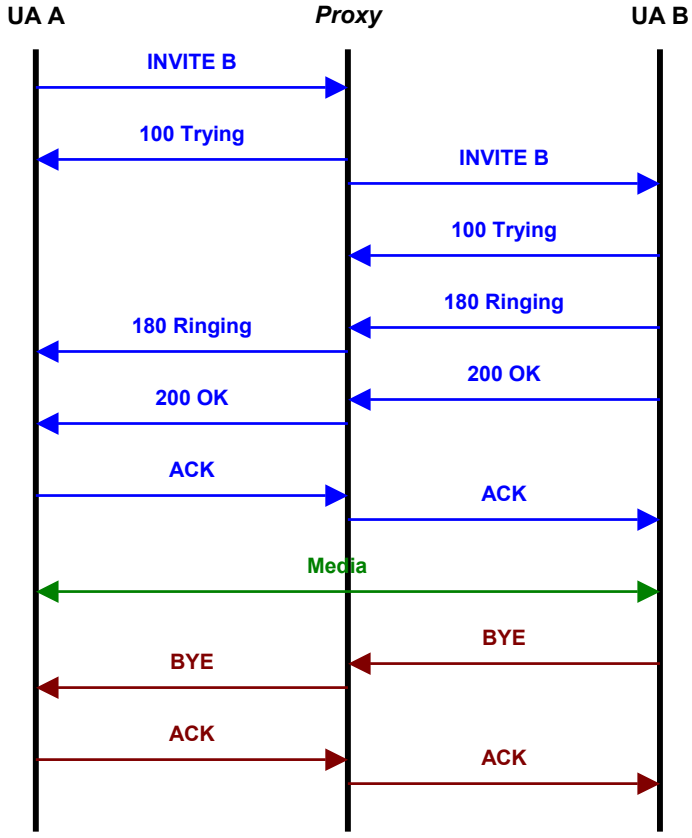

Figura 3-19: Intercambio de mensajes *SIP* en un mismo domino

- El agente de usuario de *A* enviará con destino el *UA B* una petición de *INVITE*, con el objetivo de establecer una comunicación con él. Este mensaje también contiene el paquete *SDP* describiendo las capacidades multimedia del terminal llamante.
- Esta petición llega al *proxy*, que determina dónde se encuentra B y realiza dos acciones:
  - En primer lugar envía un mensaje de *100 trying* (intentando) que informa al agente de usuario que ésta petición ya le ha llegado al *proxy* y que, por tanto, no es necesario que vuelva a retransmitirla.
  - El *proxy* contacta con su servicio de localización y envía la petición generada en *A* hasta su destino.
- El *UA B* realiza dos acciones:
  - Devuelve al *proxy* un respuesta de tipo *100* para que no reenvíe la petición.
  - Acepta la llamada 200/OK y envía el mensaje al llamante (*UA A*). Este mensaje contiene también un paquete *SDP* describiendo las capacidades multimedia del terminal llamado.
- El *UA A* reconoce la respuesta del *UA B*.
- Se procede la comunicación propiamente dicha.
- El *UA B* decide terminar la comunicación, por lo que envía una petición de tipo *BYE* al *UA A*.
- Éste notifica que la conexión ha terminado, y se finaliza la transacción.

- Llamada externa. El diagrama de la página siguiente muestra el establecimiento de una comunicación cuando llamante y llamado no se encuentran bajo el dominio del mismo servidor *proxy* y la llamada tiene que ser encaminada a través de un servidor de redirección.
  - El agente de usuario (en concreto su *UAC*) del iniciador de la llamada envía un mensaje *INVITE* al servidor *proxy* para comunicarse con otro determinado usuario.
  - Como el extremo receptor de la comunicación no está bajo el mismo dominio que el iniciador, la petición de éste es dirigida a un servidor de redirección, que contacta a un servidor de localización (paso que se ha obviado en el ejemplo anterior) para que éste proporcione la ubicación de la parte llamada.
  - Una vez el servidor *proxy* conoce la ubicación del destinatario le reenvía la petición del usuario hasta el servidor *proxy* del dominio al que pertenezca el llamado. Además confirma al servicio de localización que ha recibido su respuesta.
  - El *proxy* de red local al llamado realiza los mismos pasos que el de la red llamante.
  - El llamado confirma su intención de responder y el llamante acepta.

- Se procede a la comunicación de multimedia (voz, video, mensajería instantánea, etc.)
- El llamado inicia la terminación de la conexión.
- El llamante confirma el fin.

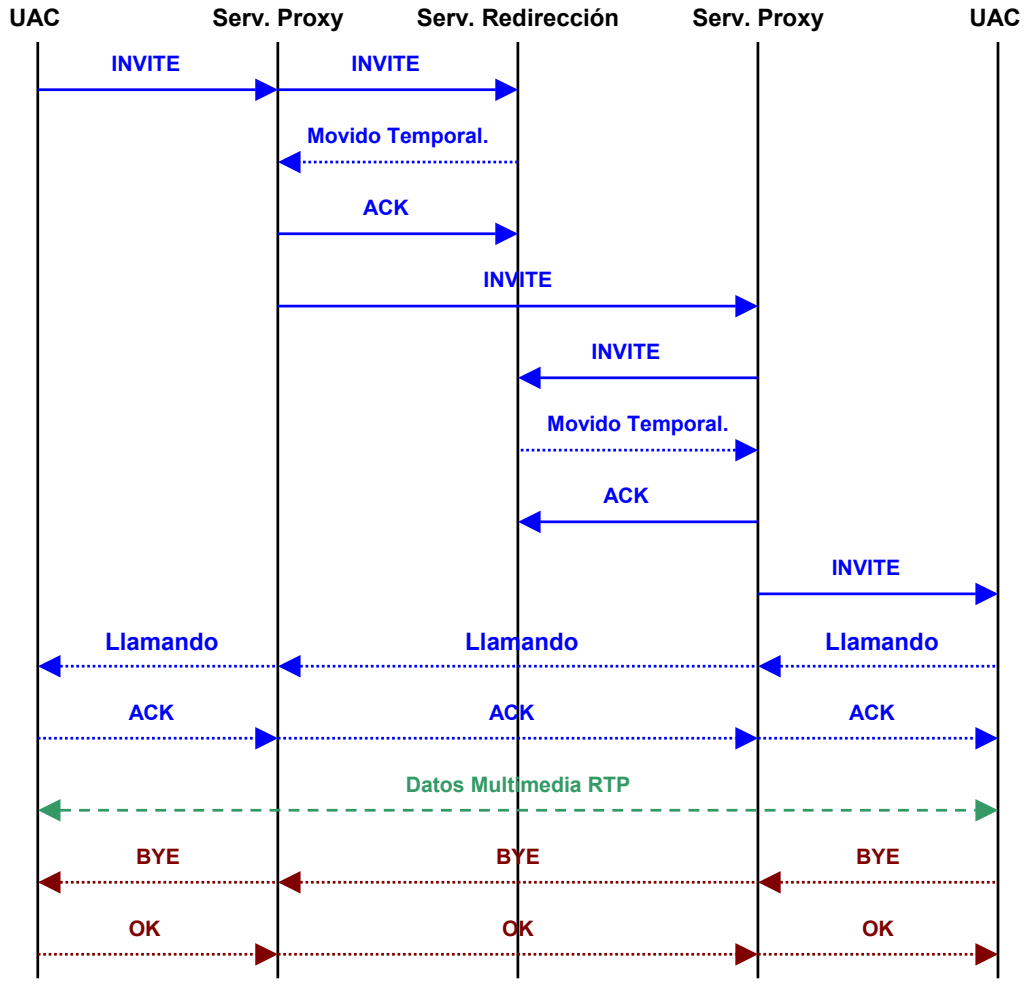

Figura 3-20: Ejemplo de comunicación *SIP* con redirección

Como nota a la ilustración anterior cabe destacar que se puede obviar el paso de la consulta del *proxy* local al llamado y el paso de los mensajes a través de él realizando la conexión de forma directa. Se pueden observar un mayor número de ejemplos en [10] y [11].

3.7 Comparativa

Una vez detallados los protocolos más extendidos en la provisión del servicio de telefonía *IP* es conveniente establecer de una forma lo más gráfica posible las ventajas y desventajas que se asocian a cada uno de ellos en comparación con el resto de las opciones disponibles. La siguiente tabla muestra las características generales de los protocolos / estándares que han sido explicados a lo largo de este capítulo.

	H.323	SIP	MCGP/H.248/MEGACO
Cuerpo de Estandarización	ITU	IETF	MGCP/MEGACO – IETF; H.248 – ITU
Arquitectura	Distribuida	Distribuida	Centralizada
Control de llamada	Gatekeeper	Servidor Proxy / Redirección	Agente de llamada / MGC
Extremos	Gateway, terminal	Agente de usuario	Gateway
Transporte de señalización	TCP / UDP	TCP / UDP	MGCP – UDP; Megaco/H.248 – Ambos
Servicios Suplementarios	Provistos por los extremos o el control de llamada	Provistos por los extremos o el control de llamada	Provisto por el agente de llamada

Tabla 3-3: Comparación de las características básicas de H.323, *SIP* y *MGCP / H.248 / Megaco*

En cuanto a la determinación de utilizar uno u otro ya se expresó que las soluciones centralizadas son las más apropiadas para las redes de los operadores de telefonía convencionales mientras que las distribuidas se adaptaban mejor al modelo de control del servicio sobre una red de datos.

Tomando en cuenta la premisa anterior bastará con establecer una comparación entre los dos estándares de telefonía sobre sistemas de tipo distribuido, la cual se ofrece en la siguiente tabla:

H. 323	*SIP*
Robusto pero consume más tiempo durante el establecimiento de la llamada	Simple, escalable y extensible.
Requiere alrededor de doce paquetes para establecer la llamada	Requiere alrededor de cuatro paquetes para establecer la llamada
Requiere *TCP* y *UDP* durante el establecimiento de llamada	Básicamente funciona sobre *UDP*. La fiabilidad se consigue a través de retransmisiones. También soporta *TCP*.
Provee control de flujo durante una sesión	No puede proveer control de flujo
Tiene capacidades de intercambio más elaboradas (H.245)	Mínima capacidad de intercambio pero suficiente para telefonía IP
Sólo puede utilizar los *códecs* definidos en la recomendación H.323	El uso de *SDP* permite a los agentes de usuario comunicarse utilizando cualquier *códec*
Provee una unidad de control para conferencias multipunto	No es necesario este elemento para conferencias *multicast SIP*.
Arquitectura fija. El despliegue es complejo y costoso en tiempo	Arquitectura escalable. Fácil de implementar

Tabla 3-4: Comparativa de las características básicas de H.323 y *SIP*

Después de la comparativa anterior y atendiendo a lo descrito en este capítulo se va a decidir la elección de *SIP* como protocolo para desplegar el servicio de *VoIP*. Esta elección se basa en las siguientes características:

- Establecimiento más rápido de conexión y menor ancho de banda requerido.
- Flexibilidad de ampliación de funcionalidades.
- Flexibilidad en la comunicación con todo tipo de *códecs*.
- Compatibilidad con nuevos productos y servicios.

En los puntos restantes de este capítulo se analiza un problema que es común a la todos los protocolos de señalización explicados hasta este momento, los conflictos con las técnicas *NAT*. Además se exponen las características de un protocolo de señalización diseñado específicamente para superarlos.

3.8 Conflictos de los Protocolos de Señalización con las Técnicas NAT

En puntos anteriores se han detallado las características fundamentales de los protocolos de señalización pero se ha obviado una debilidad que atañe a todos ellos y que no se suele encontrar demasiado extendida en la literatura que explica sus características fundamentales, que es el hecho de la difícil compatibilidad de éstos con las técnicas *NAT* (*Network Access Translation*). Estas técnicas se aplican en los extremos de casi todas las redes actuales y son claves en el transporte del tráfico a través de *Internet*.

Las técnicas *NAT* se usan para ahorrar direcciones *IP* y consisten principalmente en traducir direcciones *IP* y números de puerto privados a direcciones *IP* públicas cuando se genera tráfico desde la red privada a la pública. Esto permite servir las necesidades de una corporación relativamente grande con un número limitado de direcciones *IP*.

Cada dispositivo en la red privada tiene su propia dirección *IP*. El tráfico enviado a un dispositivo de la red pública será dinámicamente asignado a un número de puerto específico en la dirección pública. Así, el *router* que realice el *NAT* mantendrá una tabla que enlaza las direcciones privadas y los números de puerto a otros números de puerto asociados a direcciones públicas, pero habrá sido una condición indispensable que el iniciador de la conexión se encuentre en la red privada.

Lo que se va a exponer a continuación ocurre también para protocolos como *H.323* o *MGCP* pero se tomará el ejemplo de *SIP* para detallar el proceso.

Los mensajes *SIP* que se intercambian entre los clientes contienen detalles sobre el direccionamiento IP y los puertos que los Agentes de Usuario desean utilizar la intercambiar los flujos de información multimedia. De este modo, cuando los clientes intentan usar esas direcciones privadas para enviar o recibir tráfico *RTP*, la conexión falla porque los patrones definidos no son encaminables.

Las propuestas actuales para resolver este problema son:

- *Universal Plug and Play (UPnP).*
- *Simple Transversal of UDP Through NATranslation devices (STUN)*.

- Gateway de capa de aplicación.
- Configuración manual.
- Técnicas de túnel.
- Proyección automática de canal (*ACM / Automatic Channel Mapping*).

En los siguientes puntos se explican las características fundamentales de cada solución.

3.8.1 Universal Plug and Play (UPnP)

Esta tecnología se utiliza fundamentalmente en el ámbito doméstico y se encuentra principalmente propulsada por *Microsoft*. Esta técnica permite a las aplicaciones de los clientes descubrir y configurar componentes de red que están equipados con el *software UPnP*.

La necesidad fundamental de la telefonía IP es descubrir y usar la dirección IP y el puerto externos que le sean asignados por el *NAT*, tanto para la señalización como para los flujos de tráfico.

Una vez que dicha información es conocida, el cliente puede utilizarla en la señalización para establecer la llamada. Esto asegura que la llamada usa direcciones y puertos públicos además de la conectividad extremo a extremo.

El problema principal de esta técnica es que aún existe un número reducido de clientes que soportan el protocolo, aunque este número se encuentre en expansión, y que su compatibilidad con los *firewalls* es aún reducida.

3.8.2 Simple Transversal of UDP Through Network Address Translators (STUN)

Este protocolo ha sido diseñado para permitir a los clientes determinar si se encuentran bajo un *NAT* o un *firewall y*, en caso afirmativo, determinar su tipo y ayudarlos a encaminar correctamente sus paquetes. Ha recibido mucha atención como técnica del *IETF*, pero aún mantiene la desventaja de no ser totalmente compatible con todos los *NATs*, no funcionando bajo *NATs* de tipo simétrico, que son los más comunes.

Las características fundamentales de este protocolo son:

- *STUN* permite a un dispositivo encontrar su dirección IP pública y el tipo de servicio *NAT* detrás del que se encuentra.
- *STUN* opera en el puerto 3478 *TCP* y *UDP.*
- *STUN* no es aún soportado por los sistemas de *VoIP* de forma amplia.
- *STUN* necesita ser asociado a un dominio DNS.

Este protocolo distingue entre dos elementos:

- Cliente *STUN*. Entidad que genera peticiones *STUN* para determinar la dirección y puerto a utilizar, puede ser ejecutada en un sistema final tal como el PC de un usuario o un servidor de conferencias. Una vez que los

clientes tienen la información completa ya pueden construir los mensajes necesarios para el establecimiento de la llamada.

- Servidor *STUN*. Entidad que recibe las peticiones STUN y genera las respuestas correspondientes, que informan a los clientes del espacio de direcciones públicas y de los puertos que pueden ser usados para una sesión en particular. Usualmente estos servidores se encuentran situados dentro de la red pública.

Si atendemos a los diferentes tipos de *NATs* que podemos encontrar tenemos:

- *NATs* en los cuales la dirección y el puerto internos son siempre proyectados a la misma dirección y puerto externos. De este modo, cualquier *host* externo puede enviar un paquete al *host* interno, enviando un paquete a la dirección externa.
- *NATs* con las mismas premisas anterior pero en los que es necesario que el *host* interno inicie las conexiones para que se le asigne la dirección y puerto adecuados en la tabla *NAT.*
- *NAT*s simétricos, en los que todas las peticiones desde la misma dirección IP interna y puerto hacia un destino específico (dirección IP y puerto), son convertidos a la misma dirección externa y puerto. Si el mismo *host* envía un paquete con la misma dirección origen y puerto pero a un destino distinto, se realiza una conversión diferente. De este modo, sólo los *hosts* externos que reciban un paquete pueden enviar una respuesta al *host* interno.

En el último tipo de éstos, como la señalización se envía de forma independiente a los mensajes de datos, ocurre que existe una traducción distinta para los mensajes de señalización y para los mensajes de transporte de datos. Entonces, como se usa un puerto distinto para el tráfico saliente puede inducir al sistema a determinar que los mensajes intercambiados durante el inicio de la conexión son incorrectos, lo que hace que la conexión falle.

Asimismo *STUN* confía en el hecho que una vez un puerto saliente ha sido traducido, cualquier tráfico que aparezca desde cualquier parte de la red, con cualquier dirección IP origen, podrá usar la traducción en sentido inverso. El problema de este enfoque es que convierte a los *NATs* en servidores susceptibles a ataques por rastreo de puertos y crea problemas de seguridad.

Estos problemas no vienen por un mal diseño del protocolo, sino que se derivan de la falta de conductas estandarizadas en los *NATs*. El resultado de esta falta de estandarización ha sido la proliferación de servicios cuyo comportamiento es altamente impredecible, extremadamente variable e incontrolable.

STUN hace “lo mejor que puede” en estos entornos hostiles pero la solución sigue siendo la introducción de *NATs* con comportamientos estandarizados. Sin embargo, hasta que esto ocurra, *STUN* provee una buena solución a corto plazo dadas las condiciones bajo las cuales tiene que operar.

3.8.3 Gateway de Capa de Aplicación

Este elemento no es más que un *firewall/NAT* mejorado que es capaz de identificar los mensajes de información y su relación con los flujos de datos. De este modo hogeneizan la señalización y las cadenas multimedia en cuanto a direcciones IP y puertos.

Como desventajas de esta técnica encontramos que requiere el reemplazamiento o actualización de los equipos *NAT* que se estén utilizando y provee una configuración más compleja que la de los equipos a los que complementa o sustituye.

3.8.4 Configuración Manual

En estos casos el *NAT* es configurado manualmente con las traducciones necesarias para cada cliente. En este método es necesario que el cliente tenga una dirección *IP* fija junto con unos puertos también fijos por los que transmitir y recibir los datos y la señalización.

El problema de este tipo de soluciones es que únicamente son apropiadas para redes pequeñas donde se conozca muy bien cómo configurar el *NAT*.

3.8.5 Técnicas en Túnel

Este método se basa en crear un túnel tanto para los datos como para la señalización que traspase el *NAT*. Este método requiere dos servidores, uno en la parte privada y otro en la pública entre los que crear el túnel.

Así, el servidor externo modifica la señalización para reflejar los puertos correctos de escucha del cliente, lo que permite al sistema realizar y recibir llamadas de VoIP.

Las desventajas de esta solución vienen de los riesgos que aparecen por la vulnerabilidad que pueda tener servidor externo, que provee una forma fácil de alcanzar la red privada; y por el incremento de los retardos, debido a la necesidad de tratar los paquetes, que puede provocar una reducción en la calidad de la voz.

3.8.6 Proyección Automática de Canal

En esta aproximación se utiliza un elemento que actúa como un proxy doble, que controla la señalización y el tráfico multimedia de forma simultánea.

Si se está utilizando señalización *SIP* se puede ver el proxy de señalización como un *B2BUA* (*Back-to-Back User Agent*), actuando como punto de tránsito de todos los mensajes de señalización que se generen desde y hasta los clientes, incluidos los intercambiados con el servidor *SIP*. De esta forma se provee una visibilidad y un control completos del establecimiento de la llamada, que son dirigidos al *proxy*, realizando cambios en los mensajes para habilitar el intercambio correcto de información con el servidor *SIP*.

El proxy multimedia será el encargado de realizar el tránsito entre los flujos *RTP/RTCP* existentes que intercambian los agentes de usuario. Este elemento asegura una visibilidad y control completos de las cadenas de que contienen la información de voz, lo que sirve para proporcionar calidad de servicio y puede usarse para propósitos de tarificación. Del mismo modo, resulta un punto de aislamiento entre los clientes de la red interna y cualquier usuario no autorizado que desea provocar un ataque de denegación del servicio.

La principal desventaja de esta aproximación es que el servidor *SIP* se encuentra en la red pública, es susceptible de ataques, por lo que se ha de prestar especial atención a las cuestiones de seguridad.

3.8.7 Inter-Asterisk Exchange

Los problemas detallados anteriormente han hecho que se aborde el diseño de protocolos que no adolezcan de los mismos. Uno de estos protocolos es *Inter-Asterisk Exchange* (*IAX*), se puede definir como un protocolo *peer to peer* especialmente diseñado para proveer la señalización y el transporte de información necesarios en las llamadas de voz sobre IP [12].

Los principales objetivos de este protocolo son minimizar el ancho de banda que ocupa el tráfico de control y proveer transparencia a la utilización de técnicas *NAT* en la red.

Para conseguir estos objetivos multiplexa la señalización (secuenciación e información temporal) y el transporte de múltiples cadenas de información multimedia sobre una asociación *UDP* entre dos *hosts* y no utiliza el protocolo *RTP* (*Real-Time Transport Protocol*). De este modo, como la señalización y el transporte de información se realizan sobre el mismo puerto *UDP*, *IAX* no sufre los problemas derivados del *NAT* asociados a *SIP* o *H.323*.

Para identificar las distintas conversaciones añade un número de 15 bits (Calling Number), donde el valor cero está reservado y sirve para indicar que el destino es desconocido, es decir, que no se conoce su identificador.

En cuanto a su formato de mensajes, *IAX* es un protocolo binario, diseño que se realiza para mejorar la eficiencia en el uso de ancho de banda, especialmente en llamadas individuales.

Los mensajes de este protocolo se denominan tramas. Cada flujo de información de los que se generan se compone principalmente de Mini Tramas *IAX* (*Mini Frames*), que contienen una simple cabecera de 4 *bytes* y que permiten enviar la información con el mínimo *overhead*; suplementándose por Tramas Completas (*Full Frames*) de forma periódica, que incluyen la información de sincronización y la confiabilidad.

Existen dos tipos de información de control que se intercambia entre extremos usando Tramas Completas: Tramas de Control y Tramas de Control IAX.

- Las Tramas de Control proveen el control de las sesiones establecidas, como por ejemplo las de los dispositivos conectados a un extremo *IAX*.
- Las Tramas de Control *IAX* se usan para dirigir las interacciones de *IAX* que son normalmente independientes del tipo de extremos.

Para ilustrar mejor el funcionamiento de este protocolo se va a proceder a explicar el intercambio de mensajes en el establecimiento de una comunicación entre dos *hosts*. Ahora imaginemos que el *host A* quiere establecer una conversación con el *host B*:

- Para comenzar la conversación el *Host A* envía un mensaje *NEW* al *Host B*.
- Éste responde inmediatamente con aceptando el mensaje (*ACCEPT*) indicando al *Host A* que ha recibido la petición y está comenzando a servirla.

- El *Host A* reconoce (ACK) el mensaje *ACCEPT* al *Host B*.
- Una vez que comienza a sonar el terminal en el *Host B* éste envía un mensaje *RINGING* al *Host A*.
- Que A reconoce a través de un *ACK*.
- Finalmente, cuando se coge el teléfono en el extremo del *Host B* éste envía un mensaje *ANSWER* al *Host A* indicando que el establecimiento de llamada ha sido completado.
- A partir de este punto comienza la comunicación de voz *full-dúplex*.

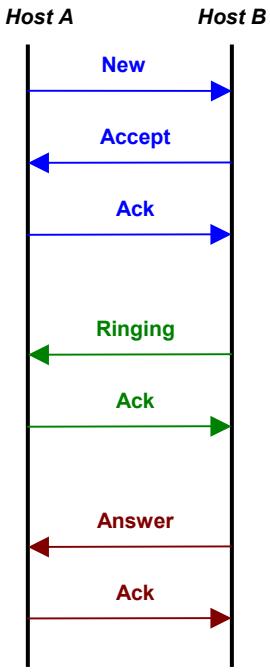

Figura 3-21: Establecimiento de llamada utilizando *IAX*

Con respecto a la implantación de este protocolo en una instalación real cabe destacar que existen múltiples terminales de usuario *software* que son compatibles con el mismo, algunos de ellos de libre distribución. Sin embargo, el hecho de que no sea aún un estándar extendido, sino un protocolo propietario de *Asterisk*, lo hace incompatible con productos de otros fabricantes, principalmente *gateways*, lo que dificulta su aplicación en infraestructuras donde se desea tener interconexión con la red telefónica básica.

En los capítulos siguientes de este proyecto fin de carrera se detallan los pasos y componentes necesarios para llevar a cabo una instalación totalmente funcional de un servicio de telefonía IP. No se utilizará *IAX* para dicha instalación, sino el protocolo *SIP*, asumiendo las limitaciones que éste tiene en cuanto a su incompatibilidad con técnicas *NAT*.

Capítulo 4
Desarrollo del Servicio de *VoIP*

4.1 Introducción

En capítulos anteriores se han detallado los parámetros fundamentales sobre los que se ha de desarrollar un servicio completo de *VoIP*; los cuales incluyen el ancho de banda, el protocolo de señalización y la configuración de los *gateways*.

De este modo, y atendiendo a los elementos definidos en los diferentes protocolos de señalización, se muestran las distintas opciones de *software* barajadas, sus características fundamentales y su configuración para acceder al servicio de telefonía *IP*. Los pilares sobre los que descansa la elección de una opción u otra vienen dados por la relación entre las prestaciones y el coste económico del *software*, así como la adaptación de ambos a la instalación que se pretende realizar y al sistema operativo disponible en el ordenador de cada usuario.

Unas vez definidas todas las opciones existentes y establecida una comparativa entre ellas, queda como tarea para el siguiente capítulo la elección entre las diferentes alternativas así como el planteamiento de una guía de instalación y funcionamiento que permita conocer y explotar al máximo las funciones de los componentes elegidos.

Cabe destacar que la búsqueda de alternativas queda limitada por ciertas características propias de la red y equipos donde se pretende instalar el servicio, en este caso los de la Escuela Técnica de Ingeniería de Telecomunicación (ETSIT). Es por ello por lo no se barajan ciertas opciones que si se deberían tener en cuenta en otros tipos de instalación, ya fuera por los parámetros de la red o del servidor.

4.2 Protocolo de Señalización

4.2.1 Introducción

El protocolo de señalización elegido para llevar a cabo esta instalación ha sido *SIP* (*Session Initiation Protocol*), tal y como se explicó en el capítulo anterior. Por consiguiente, se hacen imprescindibles tres elementos para prestar un servicio de *VoIP*:

- El servidor *SIP*, que puede añadir en un solo *software* las funcionalidades de registro, redirección y *proxy* junto con otras características asociadas más avanzadas.
- Los clientes *SIP* que proveen el acceso al servicio a los usuarios; cuyas funciones no sólo se resumen en el intercambio de señalización sino que también incluyen la captura y conversión de la voz del usuario a paquetes de datos y la transformación de los paquetes de datos recibidos con información de voz en señal audible a través de los altavoces.
- El *gateway*, que permite la interconexión con la red telefónica convencional, transformando el flujo de paquetes de datos en señales analógicas y viceversa.

A continuación se procede a discutir las características de las opciones que se han considerado.

4.2.2 Servidor SIP

Este elemento es el que aporta la inteligencia a la red *SIP* y realiza funciones tanto básicas (registro, redirección y *proxy*) como avanzadas (subscripción a eventos, presencia, configuración de planes de marcación, asociación a otras aplicaciones de mensajería, etc.).

Resulta común encontrar en un solo paquete todos los servicios básicos definidos en *SIP*, esto facilita la elección de una u otra aplicación atendiendo a las características adicionales proporcionadas. Esta elección se hará no sólo tomando en cuenta las funcionalidades definidas para el propio protocolo, sino teniendo en cuenta otras como facilidad de comunicación con la RTB (red telefónica básica) o personalización de aspectos tales como el control de acceso.

Existen múltiples paquetes de *software* que pueden ser estudiados ([13] y [14]) de entre los cuales, después de realizar un análisis exhaustivo atendiendo a la disponibilidad de programas de demostración, se tomaron en cuenta las siguientes opciones:

- *Fomine Real Time Communitacion Server* [15].
- *PBXnSIP* [16].
- *Swyx Server* [17].
- *CommnuniGate Pro Server* [18].
- *Ondo SIP Server* [19].

Como punto en común de todas las aplicaciones anteriores se encuentra el hecho de que están disponibles para sistemas operativos tipo *Windows*, parámetro crítico puesto que el servidor de la ETSIT funciona sobre un sistema operativo de tipo *Windows 2000 Server*, luego desde un principio de ha descartado cualquier *software* que no se pueda obtener para este sistema operativo.

Además, se pretende personalizar la gestión de la autenticación, de forma que ésta se pueda realizar de forma externa al propio paquete de *software*. Por consiguiente, sólo quedan como opciones disponibles *CommuniGate Pro* y *Ondo SIP Server*.

En los siguientes puntos se provee una exposición de las características fundamentales de ambas aplicaciones.

4.2.2.1 CommuniGate Pro Server

CommuniGate Pro Server es el único producto que comercializa la compañía *Stalker Software* desde su fundación en 1993. Actualmente existen más de 4000 instalaciones de esta aplicación que se estima que dan servicio a más de 26 millones de usuarios en todo el mundo.

Este software no es exactamente un servidor *SIP*, sino un servidor de mensajería. Su objetivo fundamental es permitir la transferencia de correo electrónico, pudiendo trabajar con *SMTP, IMAP, POP, Webmail* y añadiendo funcionalidades de otros protocolos tales como *LDAP*. En la siguiente figura se muestra la página de configuración principal.

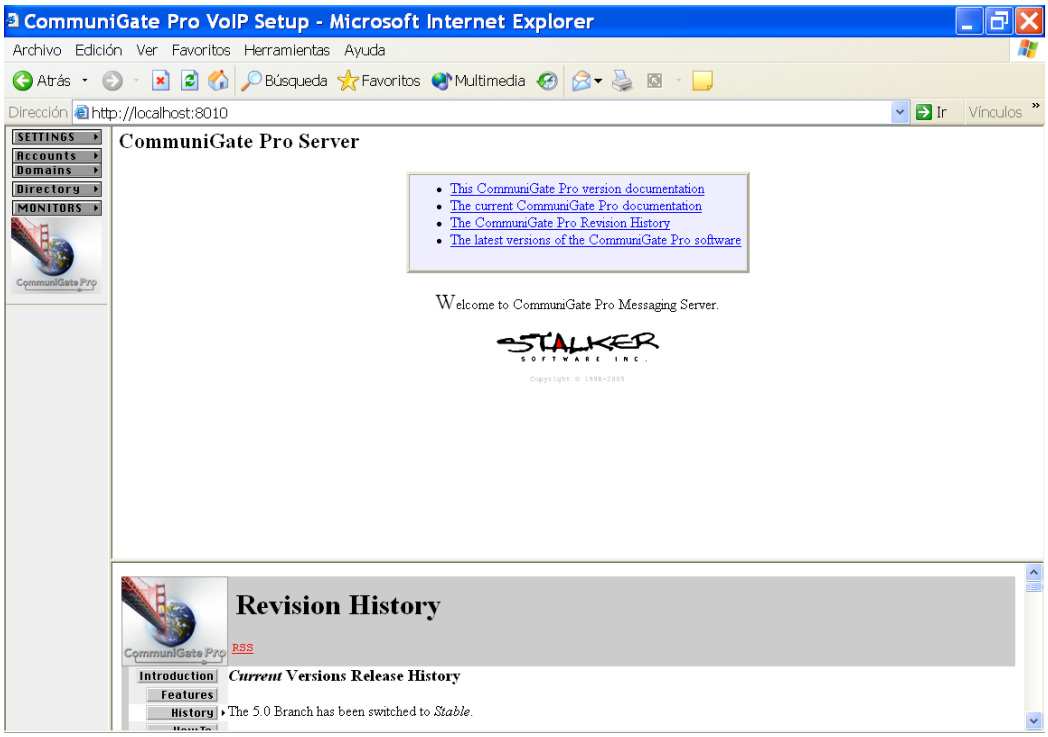

Figura 4-1: Página de acceso a la configuración de *Stalker CommuniGate Pro*

Sin embargo, en las últimas versiones se ha añadido a la transmisión de correo electrónico el soporte al protocolo *SIP*, con el objetivo de permitir comunicaciones corporativas a través de clientes *software* tipo *Windows Messenger* (otros terminales *SIP* de usuario también son compatibles, se comentan las características de algunos de ellos en puntos posteriores de este capítulo).

Los principales servicios que aporta esta aplicación son:

- Administración remota mediante interfaz gráfico o línea de comandos.
- Configuración una lista de servidores de correo.
- Multiplataforma. Existen versiones de esta aplicación para *AIX, BeOs, FreeBSD, HP/UX, IRIX, MackOS, Microsoft Windows (Win9x/XP/NT/200x), OpenBSD, OS/2, OS/400, Red Hat, Sun Solaris, SuSe, Tru64* y *Unix Ware*.

- Escalabilidad. Aunque se comience con una licencia para 50 usuarios en un único servidor cabe la posibilidad de realizar ampliaciones sin que afecte a los usuarios del sistema. Se proporciona un sistema dinámico de *clustering* para implementaciones de gran tamaño que necesitan balanceo de carga y redundancia para soportar hasta 5 millones de cuentas activas.
- Seguridad. De acuerdo a la documentación es totalmente compatible con *SSL* (*Secure Sockets Layer*) y se permite la implementación de autenticación externa.
- Fiabilidad. Aplicación de la regla de los cinco nueves implementando *clustering*, de forma que *Stalker* garantiza a sus clientes servicio en el 99,999 % del tiempo, es decir, 5 minutos de caída por año, al igual que se aplica para el servicio telefónico tradicional.
- Abundancia de documentación, tanto a través de la ayuda del propio programa como a partir de correo electrónico o listas de correo.

Si se decide instalar la aplicación, se aprecia que este proceso es sencillo (al menos en entornos *Windows*). Para acceder al interfaz de configuración bastará con abrir un navegador web con la *URL* http://localhost:8010 en la misma máquina donde se encuentra instalado el *software* o http://Direccion_IP_servidor:8010 desde cualquier otra. En cuanto a las características de configuración, se observa que existen principalmente cuatro divisiones, a la administración del servidor en sí, la de las cuentas y dominios, la del directorio y la monitorización del servicio.

El acceso a los modos de configuración suele estar protegido por contraseña, permitiéndose la definición de distintos tipos de administradores con diferentes niveles de acceso a la configuración del sistema.

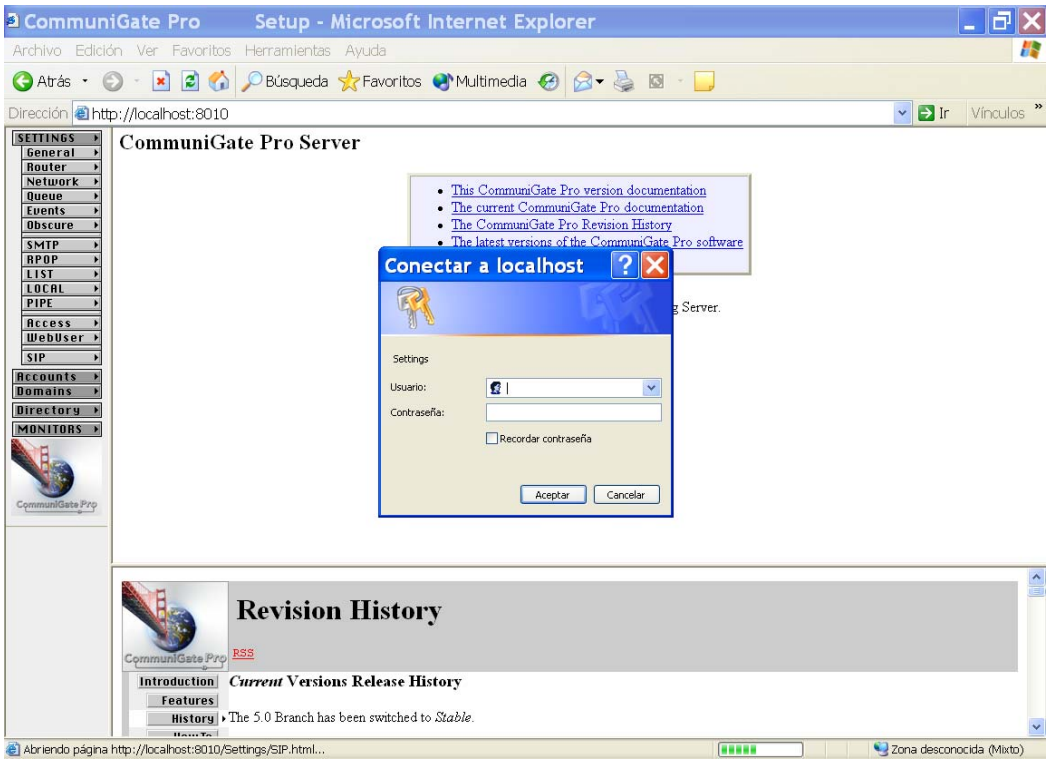

Figura 4-2: Petición de contraseña para acceder a la configuración

Para acceder a la configuración se crea la cuenta '*postmaster*' de forma automática durante la instalación, la contraseña se puede obtener consultando el fichero "*Directorio_de_Instalación\Accounts\postmaster.macnt\account.settings*". Esta cuenta no es de un solo uso sino que se mantiene activa de forma indefinida y con acceso completo a todos los parámetros de configuración del programa; a partir de la misma se pueden crear cuentas para otros usuarios con iguales o menores privilegios, por ello se considera importante cuidar la seguridad en el acceso a la misma.

Si ahora se centra el estudio de esta aplicación en la parte de soporte a la *VoIP* tenemos:

- Transporte de la comunicación sobre *TCP* y *UDP*.
- Comunicaciones seguras soportando el protocolo *TLS* (*Transport Layer Security*).
- Posibilidad de integración de varias aplicaciones de este tipo para comportarse como *media proxies* en condiciones de comunicación con *firewalls* o *NATs* (*Network Address Translation*).
- Posibilidad de creación de múltiples dominios bajo el mismo servidor para comunicaciones.
- Control de acceso de los usuarios.
- Provisión de mecanismos de presencia.
- Reenvío de llamadas hacia *gateways* de proveedores del servicio.

Sin embargo, no existe una forma clara y sencilla de crear planes de marcación, únicamente se plantea esta posibilidad para contactar con el *gateway* y tampoco se encuentra muy desarrollada; si bien ya se anuncia que es uno de los campos de trabajo actuales.

Junto con esta desventaja no debemos olvidar que esta aplicación no es específicamente un servidor *SIP*, sino que implementa uno de manera interna y una cantidad de funcionalidades más de las que no se va a sacar partido.

La licencia disponible cuando se obtiene el producto no es limitada en funcionalidad pero si en tiempo, expira a los 3 meses. El precio de la misma depende del número de usuarios (cuentas de correo) que vaya a tener tal sistema y comienza en los 25 usuarios por $699.00 [20].

4.2.2.2 Ondo SIP Server

La otra opción barajada como servidor *SIP* es *Ondo SIP Server*. La instalación de esta aplicación es sencilla y también se configura a partir de un interfaz web (requiriendo un usuario y una contraseña para acceder a los diferentes menús de configuración), lo que facilita su administración.

Las funciones que incluye esta aplicación son única y exclusivamente las relacionadas con el manejo de señalización *SIP* para el establecimiento de comunicaciones de *VoIP*. Es decir, este *software* provee las tres funciones básicas (*proxy*, redirección y reenvío) definidas para los servidores *SIP*, obviando otras como los servicios de presencia o la integración con cualquier tipo de servidor de mensajería instantánea.

En cuanto a las características propias inherentes a esta aplicación se encuentran las siguientes:

- Funcionamiento a partir del mismo instante de la instalación del *software*, prescindiendo de funciones como la autenticación de usuario. Este parámetro es configurable y se encuentra desactivado en un principio.
- Visión de los usuarios registrados instantáneamente y de ciertas características soportadas por los clientes *hardware* o *software* con los que estos acceden al servicio.
- Lista de sesiones. Que permite conocer todas las llamadas establecidas en un intervalo determinado.
- Configuración de planes de marcación. Configuración de acciones y respuestas según el tipo de mensaje, origen y destino de una conversación, modificación de direcciones, etc.
- Autenticación. Se puede optar por un acceso libre (sin contraseña) o por uno con autenticación. Ésta se puede dar tanto en el acceso de usuario al sistema (petición *REGISTER*) como al establecerse una llamada (petición *INVITE*) así como de forma interna o externa al servidor *SIP*. La introducción de los usuarios y sus claves respectivas se puede realizar a partir de un interfaz gráfico, mediante la importación de los nombres de usuario y contraseñas de un fichero separado por comas (CVS) o programando un *plug-in* que devuelva a la aplicación la contraseña para los usuarios desde una ubicación externa.
- Creación de ficheros de registro de llamadas pudiendo elegir el tiempo que se desea que estos sean guardados en memoria.
- Menú de configuración sencillo.

A continuación se va proceder a realizar un recorrido general por la aplicación con el objetivo de ofrecer una visión más completa de la funcionalidad de la misma.

Figura 4-3: Pantalla de acceso a *Ondo SIP Server*

El primer paso para acceder a la configuración del sistema es introducir un usuario y una clave que, por defecto es "*sa*". Una vez introducido estos parámetros se accede a la pantalla siguiente:

Figura 4-4: Pantalla de estado del servidor SIP

En ésta se puede parar y reiniciar la aplicación según se estime necesario. Principalmente es conveniente su utilización debido a que los cambios en la configuración no tienen efecto hasta que no se reinicia el servidor, lo que se permite en este caso es parar y reiniciar el servidor sin tener que cerrar su interfaz gráfico.

Para acceder a cada una del resto de las opciones que se ofrecen para esta aplicación hay que pulsar sobre su nombre en la barra superior, se informa del punto actual donde se encuentra el usuario porque la palabra identificativa se pone en color naranja.

De este modo, si vamos navegando a través de las diferentes opciones de izquierda a derecha tenemos:

- *Status*. Refleja todos los parámetros en los que se basa la configuración del mismo.
- *Registered*. Muestra el nombre y las características asociadas a los clientes *software* y *hardware* de cada uno de los clientes.
- *Sessions.* Mantiene un registro con todos los intentos de llamada realizados (completados o no).

- *Dial Plan*. Se pueden definir reglas para personalizar el encaminamiento de las llamadas, añadir o quitar caracteres del nombre de usuario, etc.
- *Authentication.* En esta pestaña se puede introducir manualmente el identificador y la contraseña de todos los usuarios a los que se permita acceder a la aplicación. Como este trabajo puede ser tedioso también se puede conseguir este efecto importando los mismos a través de un fichero separado por comas. Esta opción se deshabilita al instalar la autenticación externa.
- *Log.* Guarda todas las llamadas realizadas ordenadas por día. Se puede gestionar el tiempo que se desea que tales informes se desea que estén disponibles.
- *Config*. Esta es la pestaña de configuración del sistema. Se dará más información en párrafos consecutivos.
- *Logout*. Sirve para salir de la configuración del sistema sin detener la aplicación, de forma que cualquier usuario que desee acceder deberá de nuevo teclear el usuario y contraseña de acceso.

En la figura siguiente se muestra el asistente de configuración al usuario de los parámetros de configuración de sistema.

Figura 4-5: Pestaña de configuración de sistema de *ONDO SIP Server*

Como se aprecia en la figura anterior el menú de configuración se encuentra totalmente ordenado atendiendo a los diferentes parámetros de configuración. La navegación por las distintas opciones permiten configurar el *NAT Transversal*, el mensaje a partir del cual se desea realizar la autenticación (*REGISTER* o *INVITE*), los timeouts, los puertos *RTP* y el usuario y contraseña de acceso a la configuración del sistema.

Como servicios añadidos se destaca:

- La facilidad de configuración, debida a un interfaz muy intuitivo.
- La disponibilidad de documentación suficiente añadida a la gran calidad del soporte técnico.
- El coste de la licencia, cuyo coste es de $200 pero que se puede obtener de forma gratuita para usos no comerciales.

Para más información sobre esta aplicación y su configuración se recomienda consultar el siguiente capítulo, donde se explican detalles de configuración avanzados en la misma para poner en marcha el servicio de telefonía *IP* en la ETSIT de la UPCT.

4.2.2.3 Conclusión

Como se desprende del último párrafo del punto anterior se ha elegido *OnDO SIP Server* como servidor *SIP* para el servicio a desarrollar. Si realizamos una comparativa exhaustiva atendiendo a los servicios expuestos en los dos puntos anteriores tenemos:

CommuniGate Pro Server	Ondo SIP Server
Aplicación de mensajería integrada	Servidor *SIP* individual
Transporte de la señalización a través del protocolo *TCP*	Transporte de la señalización a través del protocolo *UDP*
Definición de diferentes dominios y cuentas de usuario para el mismo servidor.	Necesidad de definir un nombre en un servidor *DNS* para cada uno de los dominios que a los que se pretenda dar acceso con el servidor *SIP*
Posibilidad de gestión de la autenticación de forma externa.	Posibilidad de gestión de la autenticación de forma externa.
Mecanismos de presencia	Visualización de las características de los clientes de los usuarios registrados. Tipos de mensaje soportado y temporizador de registro.
Dificultad en la configuración de planes de marcación	Facilidad de configuración de planes de marcación
Amplia disponibilidad de documentación	Soporte muy eficiente
Licencia con actualizaciones para dos años	Licencia gratuita para fines no comerciales (uso personal y fines educativos).

Tabla 4-1: Comparación de las características principales de *CommuniGate Pro Server* y *Ondo SIP Server*

Una vez detalladas las ventajas e inconvenientes de cada opción frente a la otra, la decisión de instalar la aplicación *ONDO* se motiva en los siguientes pilares:

- Aplicación exclusiva de gestión de conversaciones *SIP*. Al contrario que *CommuniGate Pro* que es un servidor de mensajería completo.
- Disponibilidad de configuración de un plan de marcación con una documentación suficiente. Tarea complicada en *CommuniGate Pro*, siendo una de las mejoras pendientes.
- Mantenimiento de ficheros de registro.
- Personalización de parámetros como temporizadores y puertos de una forma mucho más simple e intuitiva que *CommuniGate Pro* y sin afectar a otros servicios.
- Coste de la licencia. Potencialmente gratuita para la ETSIT.

Sin embargo, la adopción de esta opción también implica que no se va a proveer el servicio de presencia. La no provisión de este servicio se puede ver técnicamente como un inconveniente puesto que el usuario llamante no puede conocer, a priori, si está disponible el llamado. Sin embargo, este inconveniente no supone un problema grave ya que en el sistema telefónico actual no existe dicho servicio, por lo que los usuarios no perciben su falta.

4.2.3 Cliente SIP

4.2.3.1 Introducción

Este elemento es el que se deberá instalar en los ordenadores personales de todos los individuos que deseen acceder al servicio de telefonía *IP*. Las funciones principales de este tipo de *software* ya fueron explicadas en el capítulo anterior pero pueden ser resumidas en proveer al usuario la posibilidad de realizar y recibir llamadas desde su *PC* de la misma forma que comprueba su correo electrónico o realiza otras tareas.

En este punto se va a proceder a la comparación de una serie de aplicaciones que proveen este servicio. La decisión de utilizar una u otra vendrá dada por el sistema operativo del usuario, las características adicionales de cada aplicación y el coste de las mismas.

En los siguientes apartados se estudia el funcionamiento de las siguientes aplicaciones:

- *Windows Messenger* (*Microsoft*).
- *Phoner* (*Heiko Sommerfeldt*).
- *PhonerLite* (*Heiko Sommerfeldt*).
- *X-Lite* (*Xten Networks, Inc*).

Además de estos clientes SIP *software* también se han probado otros, que han sido descartados por diferentes motivos. Ejemplos de este tipo de aplicaciones son:

- *Firefly,* debido a los numerosos problemas que presenta en su registro en los servidores *SIP* y el gran consumo de recursos en las máquinas donde se ejecuta.
- *MyJabber*, debido al hecho de que es un producto del que es necesario pagar licencia.
- *SIPPS Free* (*Ahead Software AG*), debido a que, una vez hecha la instalación hay ocasiones en las que el programa no arranca, luego no se estima conveniente ofrecer una aplicación cuya fiabilidad no está asegurada.

Como nota a añadir destacar que los clientes que han sido analizados no son los únicos disponibles a través de una descarga gratuita en *Internet*. Sin embargo, sí que es cierto que abarcar todas las opciones disponibles y comprobar su compatibilidad con el servidor *SIP* elegido supone un trabajo enorme cuyo fruto es incierto. Por eso en este proyecto fin de carrera se proponen una serie de opciones con una funcionalidad que se considera suficiente para proveer el servicio.

4.2.3.2 Microsoft Windows Messenger

Esta es la aplicación que más ha impulsado el crecimiento del protocolo *SIP* en su uso a nivel residencial.

Antes de continuar detallando sus características simplemente se ha de destacar que *Windows Messenger* no es lo mismo que *MSN Messenger* [21]. *Windows Messenger* posee un enfoque corporativo, que permite integrarlo con *Microsoft Exchange*, *Microsoft Live Communication Server* o cualquier otro servidor *SIP* para proporcionar un servicio de comunicaciones a una compañía o una red. En cambio, *MSN Messenger* no permite personalizar o caracterizar los parámetros de conexión al servicio, luego accede únicamente al servicio proporcionado por *Microsoft*. Por ello, si se está utilizando *Windows XP* y se quiere un servicio de mensajería personalizado, se deberá usar *Windows Messenger*.

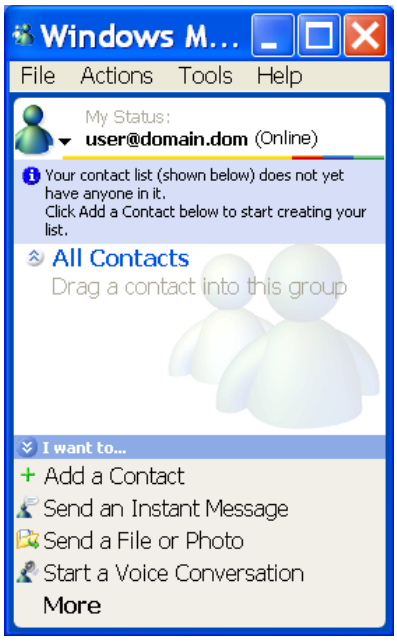

Figura 4-6: Interfaz de usuario de *Windows Messenger*

En la figura anterior se puede apreciar el interfaz de usuario de *Windows Messenger* en idioma inglés para el usuario *user*, que se conecta a un servidor *SIP* situado en la máquina *domain.dom*.

Entre las funcionalidades que ofrece este programa destacan:

- Identificación de llamada.
- Conversaciones de audio y video.
- Servicio de mensajería instantánea.
- Servicio de transferencia de archivos.
- Servicios de presencia, dependiente del *SIP Server* instalado.
- Compartición de aplicaciones y funciones pizarra compartida.
- Interfaz personalizable.
- Bloqueo de contactos.
- Integración completa con aplicaciones como *Microsoft Exchange*, *Microsoft Live Communication Server, Microsot Outlook Express* o la asistencia remota de *Windows XP*.

El principal inconveniente de este cliente *software* es la limitación que sólo funciona bajo *Windows XP*.

4.2.3.3 Phoner

Es una aplicación concebida para permitir a un usuario con una tarjeta RDSI en su computador realizar llamadas telefónicas desde el mismo utilizando la red telefónica básica. La conexión a este tipo de interfaces se realiza utilizando las funcionalidades que proporciona la interfaz *CAPI* (*Common IDSI Application Program Interface*) [22].

Además, desde la versión 1.66 permite realizar llamadas de *VoIP* a cualquier cliente *SIP*. En la figura siguiente se puede apreciar el interfaz gráfico de esta aplicación.

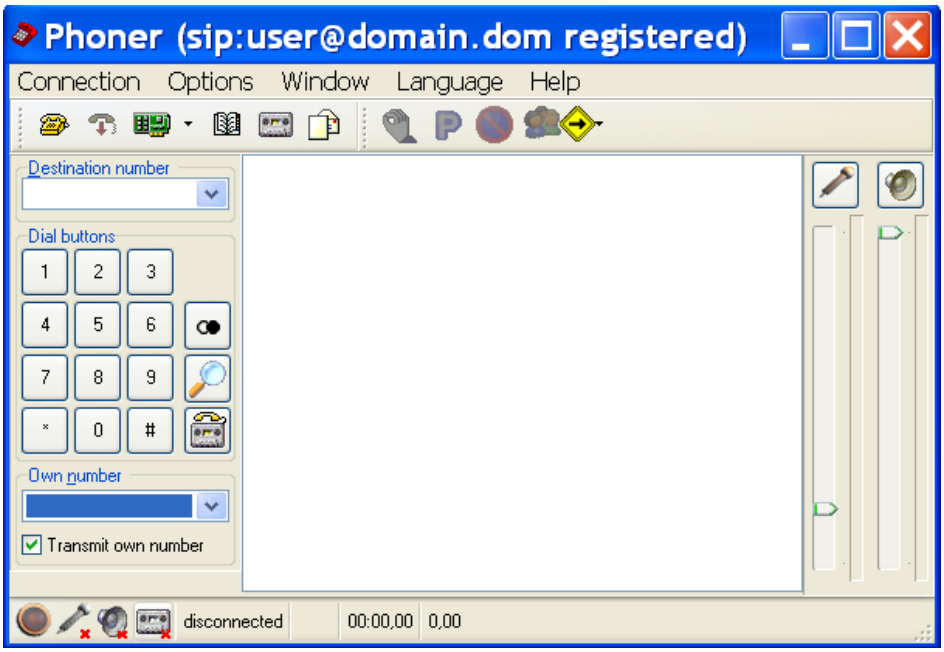

Figura 4-7: Interfaz de usuario de *Phoner*

Las características principales de este producto son:

- Identificación del llamante.
- Agenda telefónica (asignación del nombre el llamado al identificador).
- Importación de contactos de *Microsoft Outlook*.
- Tonos individuales de llamada (asignados por llamante).
- Envío de SMS a teléfonos móviles.
- Envío de correos electrónicos usando un cliente SMTP integrado.
- Grabación de la llamada actual.
- Múltiples llamadas.
- Transferencia de llamada.
- Llamada a tres.
- Servidor web integrado.
- Soporte de *VoIP* para conexiones *SIP*.
- Códecs soportados: G.711 (leyes a y u), *GSM, iLBC y Speex*.

- *STUN* soportado para conexiones a través de *NAT* o *firewall*.

Además de estas características destaca:

- Aplicación para sistemas operativos de tipo *Windows* (95/98/ME/NT40/2000/XP).
- Los recursos computacionales requeridos bajos.
  - Mínimo un *Pentium* o *AMD* a 100 MHz.
  - 32 MB de RAM.
- Cliente de libre distribución.

Además, de esta aplicación se extrae otra, *PhonerLite*, cuyas características fundamentales se exponen en el siguiente punto.

4.2.3.4 PhonerLite

Se trata de una versión de *Phoner* específica para *VoIP*. Comparte gran parte del código con la aplicación anterior, por lo que integra sus ventajas principales. En la siguiente figura se puede observar el interfaz gráfico de este cliente *SIP*.

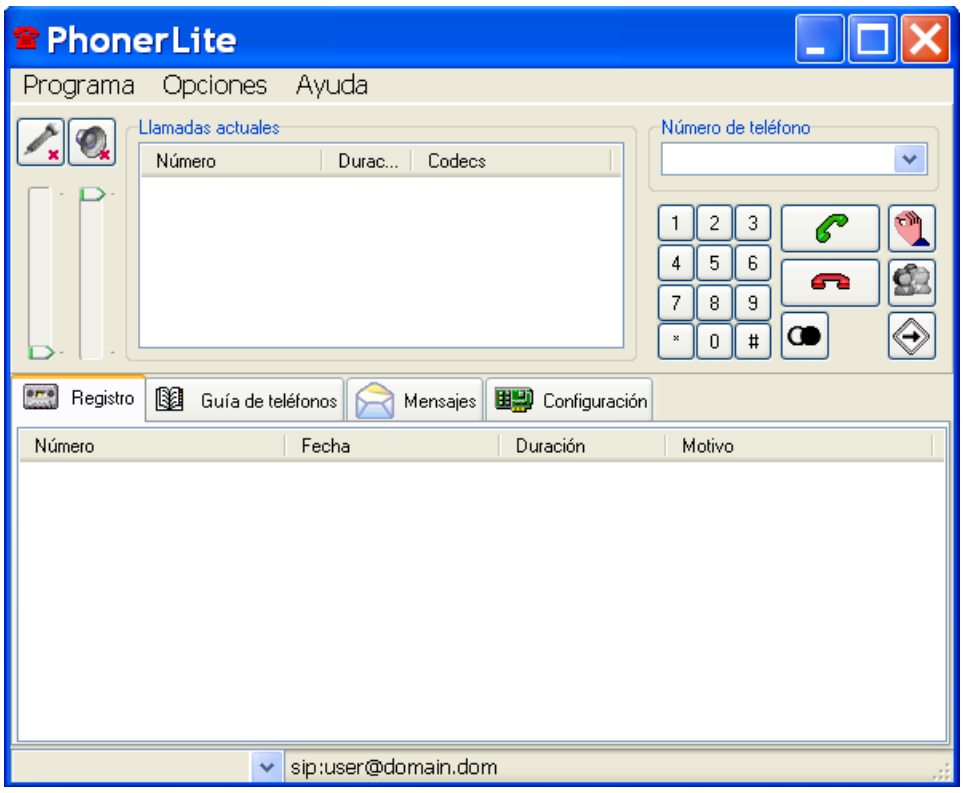

Figura 4-8: Interfaz de usuario de *PhonerLite*

Entre las características principales que incorpora esta aplicación se encuentran las siguientes:

- Interfaz en castellano.
- Identificación del llamante.
- Registro de llamadas. Permite conocer el destino de las llamadas realizadas con anterioridad.
- Agenda telefónica. Permite asociar un número a un nombre y llamarlo desde la propia guía.
- Mensajes. Posibilita enviar mensajes instantáneos a otros usuarios.
- Permite configurar la respuesta automática de llamadas.
- Existe la posibilidad de realizar tareas de depuración, lo que permite conocer mejor el progreso de las llamadas desde un punto de vista técnico así como las causas del fallo en las mismas.
- Tiene integrado un cliente *STUN* para permitir conexiones que atraviesen *firewalls* o *NAT*s en los términos que soporta dicho protocolo.

Los requisitos técnicos de los equipos son análogos a los de la aplicación anterior y resulta también sólo válida en sistemas operativos tipo *Windows*. Sin embargo, se recomienda utilizar ésta debido a que evita al usuario tener que lidiar con parámetros que no son propiamente del servicio de *VoIP* y que pueden causar confusión, así como ralentizar el equipo.

4.2.3.5 X-Lite

Este cliente *SIP* tiene un interfaz gráfico como el que se muestra a continuación:

Figura 4-9: Interfaz de usuario de *X-Lite*

A continuación se va a proceder a detallar las características más interesantes de este terminal *software* [23]:

- Disponible en versiones para *Windows, Linux* y *Mac OS X*.

- Mantenimiento de llamada.
- Registro simultáneo en múltiples proxys *SIP*.
- Muestra de la identificación de llamante (identificador *SIP*).
- Temporizador de llamada.
- Función de silenciar.
- Regulación del nivel del micrófono y altavoces.
- Duración e identificación de la última llamada.
- Registro de llamadas recientemente realizadas y recibidas.
- Agenda (obtenible y exportable en ficheros CSV).

Además cabe destacar que es un terminal de fácil configuración.

4.2.3.6 Conclusión

Para determinar cuál de los terminales telefónicos (clientes *SIP*) es el más adecuado según las necesidades que se plantee cada cliente, se va a proceder a realizar un comparativa entre todos en la siguiente tabla.

Característica	Windows Messenger	Phoner	Phoner Lite	X-Lite
Sistema Operativo	Windows XP	Todos Windows	Todos Windows	Windows, Linux y Mac OS X
Idioma del Interfaz	Inglés	Inglés	Español	Inglés
Temporizador de Duración de la Llamada	Si	Si	Si	Si
Códec Soportado	G.723.1, G.722.1 ITU standard, GSM6.10 y DVI4, G.711 (ITU standard).	G.711 (leyes a y u), *GSM, iLBC y Speex*	G.711 (leyes a y u), *GSM, iLBC y Speex*	G.711 (leyes a y u), *GSM, iLBC y Speex*
Identificación de Llamada	Si	Si	Si	Si
Agenda de Contactos	No	Si	Si	Si
Transferencia de Archivos	Si	No	No	No
Mensajería Instantánea, Servicios de Presencia, Pizarra Compartida y Videoconferencia	Si, dependiendo del servidor *SIP*	No	No	No
Multiconferencia				
Integración con Aplicaciones de envío de Correo Electrónico	Si	Si	No	No
Integración de Cliente STUN	No	Si	Si	Si
Cifrado de la Conversación	No	No	No	Si

Tabla 4-2: Comparativa de las características de los diferentes *SIP User Agents*

Atendiendo a la tabla anterior los usuarios deberán elegir cuál es el terminal que más se ajusta a sus necesidades. En este proyecto fin de carrera se recomienda usar el cliente *SIP PhonerLite* debido a que integra las funciones básicas de acceso al sistema y posee un interfaz en español, evitando al usuario trabajar y configurar funciones más avanzadas que pueden, por incompatiblidad con terminales de otros tipos ser usadas un número mínimo de veces.

Para usuario de sistemas operativos tipo *Linux* el terminal recomendado es X-Lite. El modo de configuración de todos los *SIP User Agents* estudiados se adjunta como apéndice al final de este documento.

4.3 Gateway

Una parte indispensable para permitir una conexión total entre la telefonía IP el servicio telefónico tradicional es la inclusión de un *gateway*. Las funciones de este elemento se resumen en la conversión de formatos (señalización y datos) desde la red de paquetes a la de telefónica conmutada y viceversa.

Existen multitud de opciones al adquirir un *gateway* debido a la gran extensión que está teniendo la *VoIP*; por ello es necesario determinar cuales son las características del escenario real sobre el que se ha de implantar el servicio.

Atendiendo a este criterio surgen las siguientes posibilidades:

- Conexión de los teléfonos analógicos o digitales convencionales para realizar el transporte de la información a través de una red de paquetes, de forma que este elemento sustituye la centralita tradicional.
- Conexión a una centralita para realizar el transporte de las llamadas que ésta gestiona.
- Conexión a la red de circuitos con el objetivo de finalizar las llamadas que han sido transportadas a través de la red de paquetes.

Estos escenarios determinan el número de puertos que es necesario que tenga el equipo (no es lo mismo utilizarlo como concentrador que como terminador de llamadas) así como otras características asociadas a la señalización y tipo de interfaz (*FXS/FXO*).

Normalmente estos equipos se componen de varios interfaces telefónicos (*FXS* si proporcionan el tono a los equipos telefónicos o *FXO* si no proporcionan ningún tipo de señal) y un número mucho menor de interfaces hacia la red de paquetes, que normalmente suele ser *Ethernet*.

Atendiendo a la instalación que se ha de realizar dentro de la ETSIT de la UPCT, tal y como se especificó al comienzo de este capítulo, se estima que el *gateway* más adecuado para proveer el servicio es del tercer tipo de los anteriores.

Por ello, se cree conveniente que este equipo posea un número de puertos extensible. Así, se podrá comenzar con un número de puertos *Ethernet* pequeño y un número de puertos de voz bajo (realizando un dimensionamiento a la baja del sistema) para incluir mayor cantidad de conexiones a medida que se compruebe que el sistema comienza ser más utilizado (*Ley de Metcalfe: "la utilidad de una red es proporcional al cuadrado de sus usuarios*", o, dicho de otro modo, los usuarios perciben mayor utilidad a utilizar ciertos servicios cuantos más usuarios tengan éstos).

En el siguiente capítulo se explica con detenimiento cuáles son las características finales de diseño y de servicio del sistema, así como sus parámetros de configuración.

Capítulo 5
Implantación del Servicio de *VoIP*

5.1 Introducción

En el capítulo anterior a éste se han mostrado las opciones disponibles para implantar un sistema de *VoIP* atendiendo a los tres elementos básicos del sistema, los clientes *SIP*, el servidor y el *gateway*. De todos los componentes expuestos se han estudiado ventajas e inconvenientes así como, en algunos casos, se ha delimitado la opción más conveniente.

Sin embargo, de forma previa a la instalación del servicio de telefonía IP, es necesario tener un conocimiento de la infraestructura de telefonía y datos sobre la que se ha de sustentar el mismo. El tipo de red de ambas clases existente será el que determinará la forma y el tipo de instalación.

No es lo mismo, tal y como se explicó en el capítulo anterior, realizar una instalación de VoIP para el transporte de la voz a nivel troncal, manteniendo la infraestructura corporativa de voz convencional asociada a la centralita telefónica, que migrar completamente a una red de paquetes, sustituyendo los terminales analógicos y digitales por teléfonos IP o aplicaciones instaladas en los equipos de usuario.

Es por ello que en este capítulo se comienza revisando las características de las redes de voz y datos que dan servicio al escenario donde va a funcionar el sistema, la Escuela Técnica de Ingeniería de Telecomunicación (ETSIT). Una vez conocidos estos parámetros se podrá determinar qué elementos habrán de utilizarse a nivel básico, apareciendo además nuevos requisitos adicionales que obligarán a la inclusión de otros no contemplados en las características básicas iniciales.

Además de lo expuesto en el capítulo anterior, en los siguientes puntos se podrá obtener información no sólo de los distintos elementos a instalar, que ya fue básicamente expuesta en el capítulo anterior, sino también sobre cómo han de configurarse para proveer telefonía IP de forma real. Además, se añaden los pasos realizados como consecuencia de los requisitos que no son propios de la instalación del sistema, como la autenticación de los usuarios, y otros desarrollados para facilitar el acceso o mejorar la seguridad.

5.2 Características del Equipamiento Existente

5.2.1 Características de las Redes de Voz y Datos de la ETSIT

La primera de las consideraciones que se deben tener cara a la implantación de un servicio es el conocimiento profundo de la infraestructura sobre la que se ha de soportar. En este caso, el escenario elegido para la prestación de este servicio es la Escuela Técnica de Ingeniería de Telecomunicación, que se sitúa en el Cuartel de Antiguones.

El servicio de datos se proporciona a través de una red *Ethernet* cableada utilizando cable *UTP* categoría 6, que tiene su punto de concentración en cuatro armarios, situados en cada una de las esquinas del edificio.

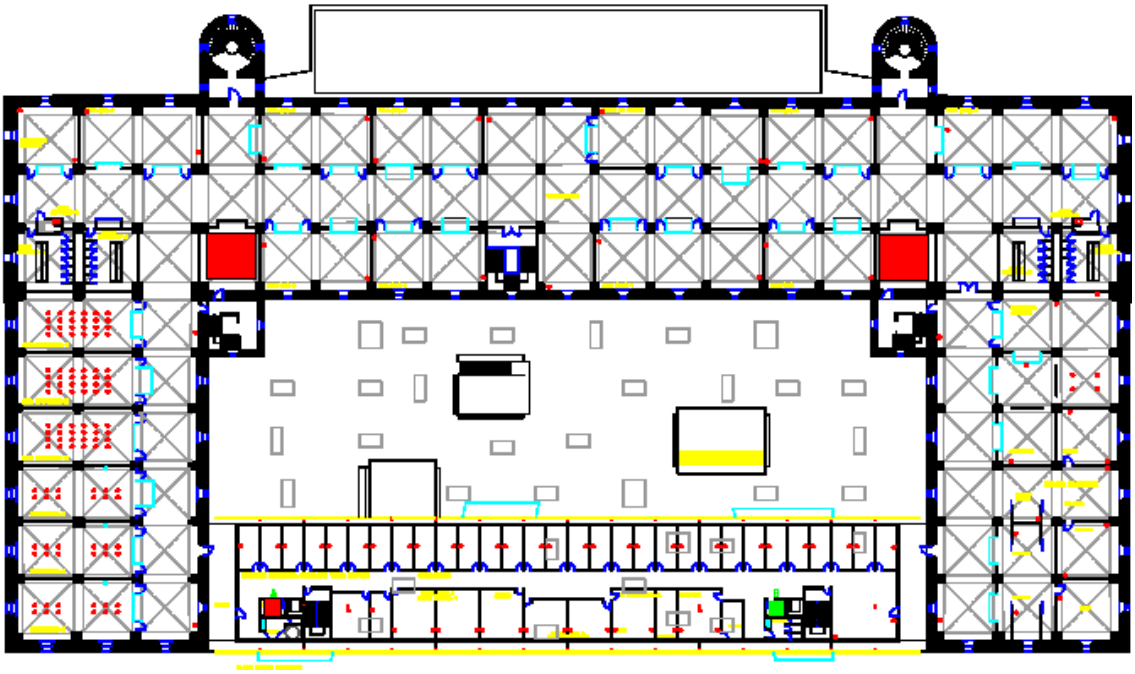

Figura 5-1: Disposición de los tres armarios de comunicaciones de la primera planta del Cuartel de Antiguones

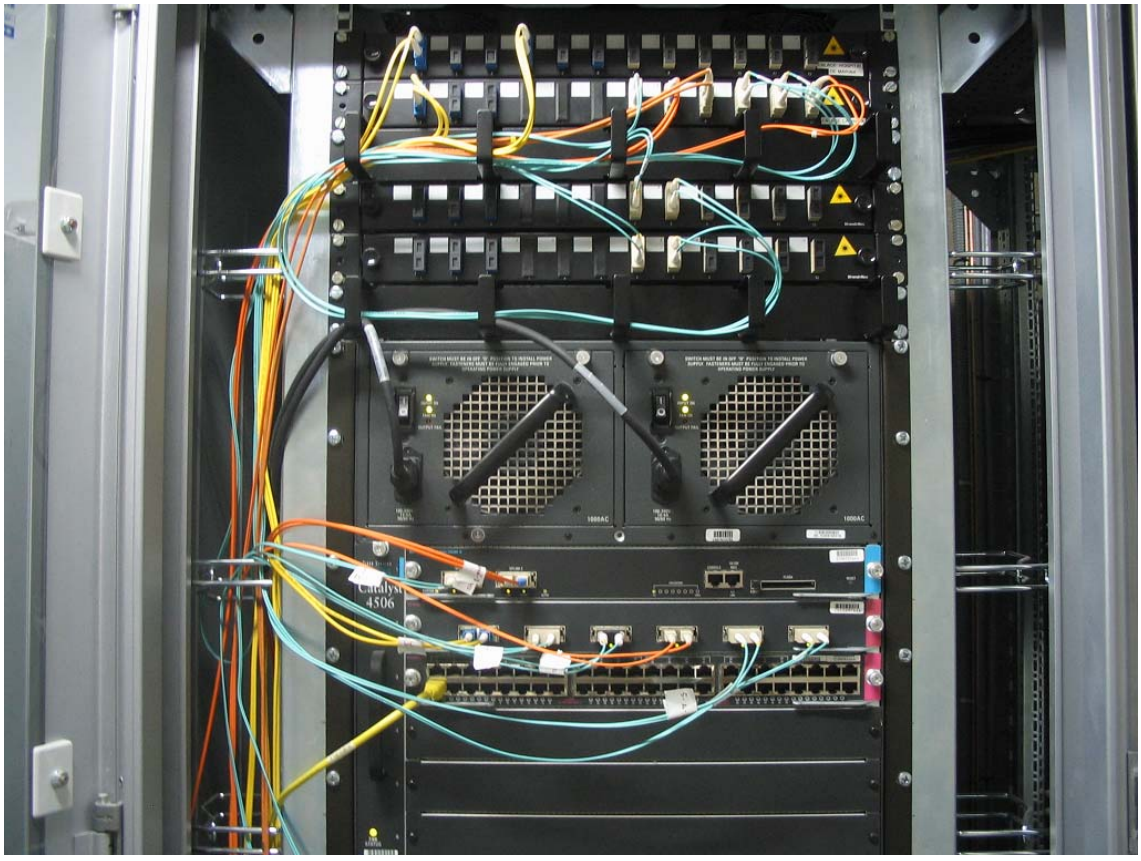

Figura 5-2: Armario de cableado principal del Cuartel de Antiguones

La tecnología de los equipos de interconexión es de la marca *Cisco*, en concreto *switches* en *stack Catalyst 2950* de 48 puertos *Ethernet 10/100* para los armarios secundarios y un *Catalyst 4506* en el armario principal. Con respecto a la unión entre los diferentes armarios especificar que se realiza a través de cableado de fibra óptica a velocidad de 1 Gbps. Sobre esta topología de red física se sustenta red de la ETSIT, que utiliza dos direcciones IP de clase C (212.128.44.0 y 212.128.45.0).

En cuanto al servicio telefónico general se soporta por dos centrales telefónicas *Alcatel* 4400, ubicadas en el edificio de la Facultad de Ciencias de la Empresa (Campus Alfonso XIII) y en el Antiguo Hospital de Marina, interconectadas a través de los circuitos de 2 Mbps con 30 canales vocales cada uno, utilizando una tecnología propietaria de Alcatel (no son enlaces primarios).

De este modo, ambas centralitas se comportan como un único sistema telefónico, con 1000 extensiones analógicas y 96 extensiones digitales, equipado con 3 interfaces para accesos primarios (en la actualidad 2 en servicio, con un primario con Telefónica de España S.A. y otro con MoviStar). Las extensiones digitales, sistema propietario de ALCATEL, son convertibles en RDSI S0 (Accesos básicos RDSI) mediante conversores específicos existentes.

El servicio telefónico de la ETSIT, se ofrece a través de la conexión del mismo a la centralita ubicada en el Antiguo Hospital de Marina, mediante de dos cables urbanos de 200 pares telefónicos cada uno. La topología de la interconexión se muestra en la siguiente ilustración.

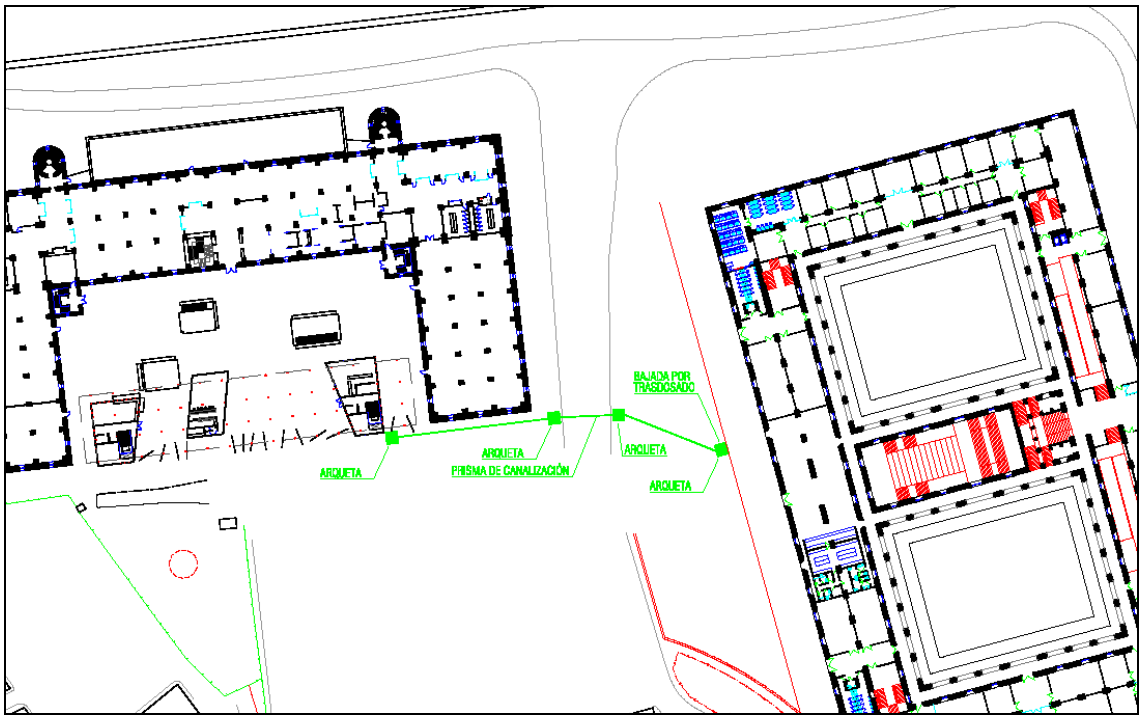

Figura 5-3: Interconexión entre los edificios de Antiguones y del Antiguo Hospital de Marina

Una vez expuestas las principales características en cuanto a la infraestructura de voz y datos existente se deben extraer dos conclusiones de forma principal:

- El cableado de datos es UTP categoría 6 y la tecnología de acceso es *Ethernet* a una velocidad de 100 Mbps.
- El acceso telefónico se realiza desde el Antiguo Hospital de Marina, luego cualquier llamada entrante o saliente al sistema de *VoIP* se cursará a través de dicha centralita.

5.3 Servidor SIP

Las funciones requeridas a esta aplicación son las básicas de cualquier servidor *SIP,* si bien es imperativo de esta instalación la posibilidad de personalización de una de ellas: la autenticación de los usuarios en su proceso registro en el sistema.

Además de esta funcionalidad básica, también resulta indispensable contar con servicios avanzados de centralita como la configuración de planes de marcación, y con otras funciones auxiliares como la posibilidad visualizar el histórico de llamadas realizadas.

Como ya fue expuesto en el capítulo anterior, el servidor *SIP* elegido es *Ondo SIP Server,* pues puede funcionar sobre *Microsoft Windows 2000 Server* y desarrollar todas las necesidades funcionales. Por ello, en este punto únicamente se va a exponer cómo se ha de configurar el mismo para proveer los servicios requeridos.

5.3.1 Autenticación

El *SIP Server* elegido permite su utilización con o sin registro en el mismo. Si se decide establecer un servicio sin registro previo, no se podrá restringir el uso del sistema a nadie, sin embargo, esta función es muy útil en tareas de depuración de la instalación, ya que permite determinar si el fallo está en el proceso de autenticación o si es independiente a él (mala configuración de algún otro parámetro de la conexión entre el agente de usuario y el servidor).

En cambio, para proveer un servicio real es imperativo autenticar a los usuarios. Este servidor SIP permite realizar el proceso cuando le llegan dos tipos de peticiones distintas, *REGISTER* e *INVITE*. Evidentemente, la opción más segura sería verificar al usuario cada vez que desea establecer una conversación, es decir, cada vez que envía un mensaje *INVITE*, sin embargo, la imposibilidad de registrar el *gateway* elegido ha reducido la autenticación a la petición *REGISTER*, enviada cuando se desea iniciar el uso del sistema.

Utilizando la autenticación en una petición o en otra, la aplicación escogida para actuar como servidor permite la creación de un programa auxiliar en lenguaje *Java* que actúa como *plug-in*. Este código tiene como requisito único entregar al servidor un nombre de usuario y una contraseña para que éste pueda validarlos con los que le envía el usuario que se pretende registrar. Para el servicio desarrollado, este procedimiento se realiza en dos fases.

La primera de estas fases consiste en el registro del usuario en el servicio, que se realiza a través de una página web creada al efecto y la segunda fase viene determinada por la lectura de la contraseña asociada al nombre de usuario.

5.3.1.1 Proceso de Registro en el Servicio

La iniciativa de registrar al usuario en el servicio, en lugar de acceder al directorio LDAP de la UPCT, viene dada por cuestiones tanto de seguridad como de rendimiento:

- Las cuestiones de seguridad se motivan en el hecho de que no es admisible que se pueda extraer la contraseña del profesorado desde los servidores de la UPCT, puesto que esto puede crear debilidades en el sistema de seguridad.
- Las cuestiones de rendimiento se centran en el hecho de que minimizar el número de mensajes intercambiados disminuye la latencia en el proceso de registro y lo hace independiente del estado del servidor de directorio global.

Figura 5-4: Interfaz gráfico principal de la página de registro en el servicio de telefonía IP de la ETSIT

Es ello por lo que se ha considerado adecuado crear una página web donde los profesores, al registrarse, dejen constancia de su nombre de usuario y contraseña de forma local, para que el servidor *SIP* pueda obtener de forma local tales parámetros. A continuación se procede a explicar cómo se gestiona dicha suscripción al sistema. Para un mejor entendimiento del proceso seguido en la programación de la web se recomienda consultar el Apéndice A.

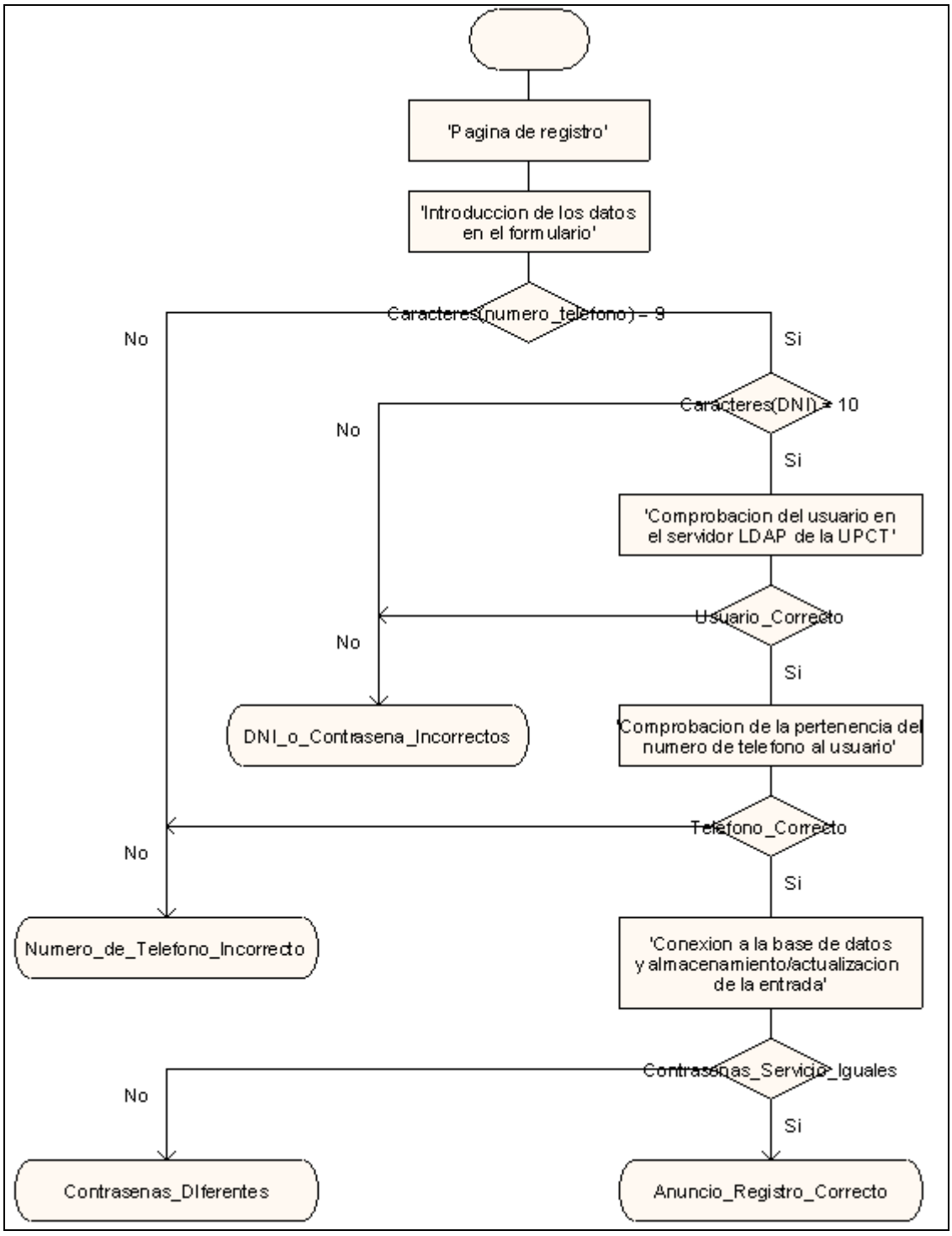

Figura 5-5: Diagrama de flujo del proceso de registro en el servicio de telefonía IP

Téngase en cuenta que el lenguaje elegido para realizar los procesos que se exponen a continuación ha sido *PHP*. Para aumentar la comprensión del proceso que se va a mostrar a continuación se recomienda seguir el diagrama de flujo de la figura 5-5.

1. El usuario se conecta a través de su navegador web a la página www.teleco.upct.es/telefonia-ip y se muestra un interfaz como el de la figura 5-4.
2. En esta página hay un formulario. Este formulario pide:
    a. *Número de teléfono*: las nueve cifras que componen el número de teléfono que tenga el profesor, es decir, una cadena del tipo 968328872.
    b. *DNI*: Número del documento nacional de identidad, su utilidad se explica en el siguiente punto.
    c. *Contraseña de Correo*: Es la contraseña global de la red Campus que utiliza el usuario, por ejemplo, para acceder a su correo electrónico de la UPCT.
    d. *Contraseña Servicio VoIP*: Contraseña que se desea tener en este servicio.
    e. *Confirme Contraseña*: Se pide al usuario que reintroduzca la clave que desea tener en el servicio de telefonía IP para confirmar que la primera que introdujo no fue errónea.
3. Una vez introducida toda la información y pulsado el botón *Registrar* que se encuentra bajo el formulario. El proceso comienza comprobando en el directorio de la UPCT que el usuario verdaderamente es un profesor de la misma y que posee el número de teléfono con el que se desea registrar.
4. Estos datos se obtienen realizando una consulta *LDAP* al servidor de directorio de la UPCT, intentando establecer una unión entre el *DNI* y la contraseña de red Campus introducida. Si se puede establecer esta unión significa que el DNI introducido es de un profesor de la UPCT. Sin embargo, aún queda que verificar que el usuario posea realmente el número de teléfono con el que se desea registrar. En caso contrario se muestra una página indicando el error (figura 5-6).

Figura 5-6: Aviso de fallo en la introducción del DNI o la contraseña de red Campus

5. Para confirmar su número de teléfono basta con generar una búsqueda que asocie el número de teléfono con el DNI introducido. Contando el número de coincidencias se puede determinar si el usuario ha introducido éste correctamente. En caso contrario, también se muestra una página de error (figura 1-7).

Figura 5-7: Aviso de fallo en la introducción del número de teléfono con el que el usuario pretende registrarse en el servicio

6. Una vez confirmados la validez del usuario y la del número de teléfono que éste introduce, se deberá verificar que las contraseñas introducidas para el servicio de telefonía IP son iguales.
    a. En tal caso se conecta una base de datos *Microsoft Access* (consultar apéndice D para conocer cómo se ha de instalar el *driver* de acceso a esta base de datos) y:
        i. En caso afirmativo se actualiza la entrada introduciendo la contraseña y DNI del usuario que ha realizado el último registro. Este paso está especialmente pensado para profesores que comparten el número de teléfono. En este caso se recomienda tener una contraseña en común para permitir a ambos acceder al servicio, a pesar de que no puedan utilizarlo de forma simultánea.
        ii. En caso negativo, se crea una nueva entrada en la tabla para el dicho usuario.
        iii. Muestra una página de éxito del registro (figura 5-8).

Figura 5-8: Registro exitoso, el usuario ya tiene su identificador y su contraseña almacenadas

    b. En caso contrario se accede a una página informativa del error (figura 5-9).

Figura 5-9: Aviso de error al introducir la contraseña del servicio

Además de las páginas mostradas también se incluye una página de ayuda y otra de descargas, que pretenden completar la anterior, proporcionándole el soporte documental necesario así como facilitando a los usuarios del servicio la obtención de un terminal adecuado y comprobado.

5.3.1.2 Acceso al Nombre de Usuario y Contraseña

Como quedó expuesto al comienzo de este apartado, esta es la segunda fase del proceso de personalización de la autenticación de los usuarios. *ONDO SIP Server* permite instalar un *plug-in* escrito en lenguaje Java (versión 1.4 o posterior). Si se atiende a la figura que se puede observar en [24] y el proceso que se explica, la autenticación utilizando el servidor elegido se realiza de la siguiente forma:

Figura 5-10: Proceso de autenticación externa utilizando ONDO SIP Server

1. El usuario envía una petición *SIP* de tipo *REGISTER* o *INVITE* al servidor.
2. Cuando llega esta petición al servidor y el *plug-in* se encuentra correctamente instalado (ver apéndice B), invoca al método *lookup()* del mismo.
3. Dicho método busca en el repositorio indicado la contraseña asociada al usuario introducido y la devuelve al servidor (mediante el objeto *UserRecord*).
4. El servidor comprueba si la clave que le ha proporcionado el usuario es igual a la obtenida del repositorio.
5. En caso afirmativo registra al usuario y en caso negativo le devuelve un mensaje rechazando el registro.

El programa que gestiona la autenticación debe implementar un interfaz llamado *UserDir*, que incluye los siguientes métodos:

- *init()*: tareas de inicialización.
- *close()*: tareas de cierre.
- *lookup()*: tareas de búsqueda.
- *append()*: añadir un usuario.
- *remove()*: borrar un usuario.
- *getCount()*: obtención del número total de usuarios.

De estos métodos, el único al que se ha dotado de funcionalidad en este proyecto fin de carrera es el método *lookup()*, mientras que a los otros han sido cubiertos incluyendo cláusulas de tipo *return* junto con el argumento de salida esperado. El método *lookup()* es llamado por el *SIP Server* siempre que este desea autenticar un usuario y tiene como parámetros de entrada:

- *Username*: el nombre con el que se pretende registrar el usuario.
- *Method*: el método que invoca dicha petición (*REGISTER* o *INVITE*).
- *Destination*: la dirección *SIP URI* del destino de la petición.
- *Authinfo:* el campo *Proxy-Authorization* del mensaje *INVITE* o el campo *Authorization* del *REGISTER*.

Como parámetros de salida, debe ofrecer de forma obligatoria, un objeto de tipo *UserRecord*, que contiene, básicamente la información:

- *Username:* nombre el usuario a verificar.

- *Password:* la contraseña obtenida del repositorio en texto plano. Esto significa que si ésta se encuentra cifrada se debe descifrar antes de devolverla al *SIP Server*.

El siguiente diagrama (figura 5-11) detalla la funcionalidad implementada para el método *lookup()*, el código del mismo se adjunta como Apéndice C.

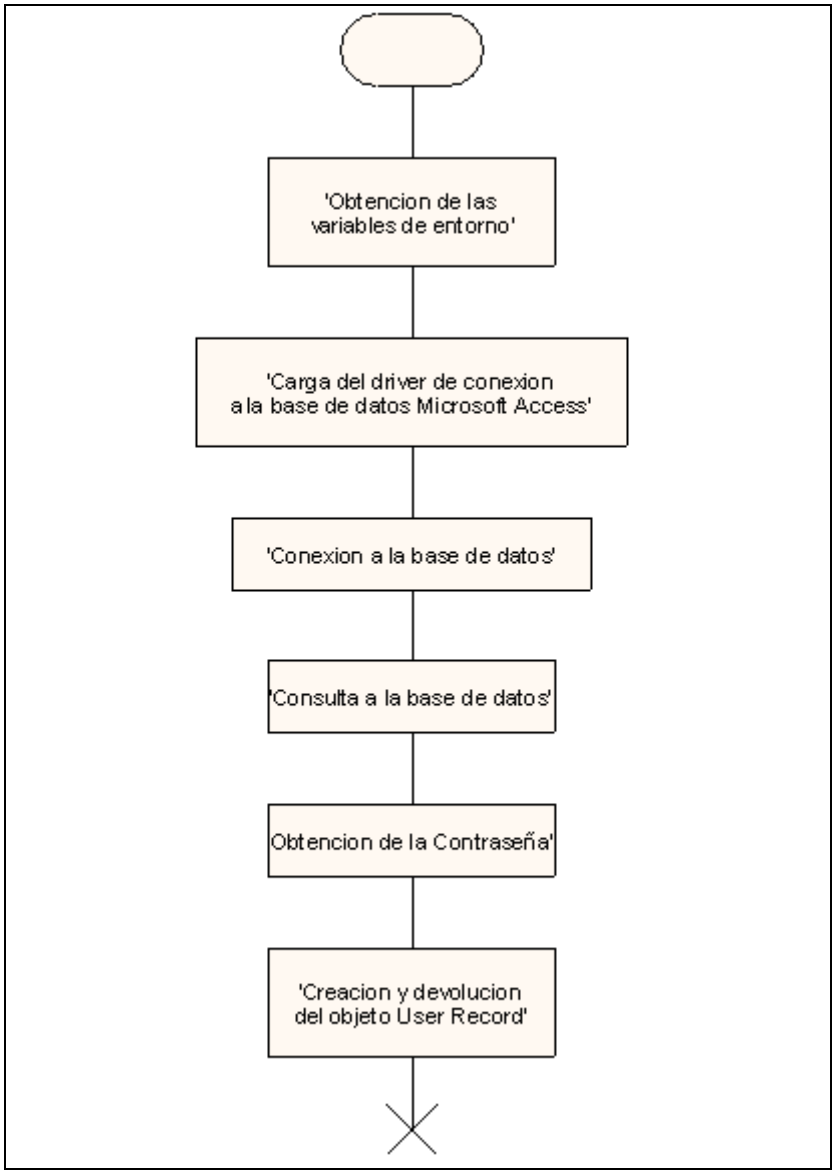

Figura 5-11: Diagrama de funcionamiento de la función *lookup()* del *plug-in* de autenticación

Hasta ahora se ha explicado el diseño del *plug-in* programado, obviando las modificaciones que se introducen el servidor *SIP*. El proceso de instalación del *plug-in* se incluye en el apéndice B.

5.3.2 Configuración de los Planes de Marcación

Un plan de marcación permite a los diferentes usuarios de un sistema de telefonía IP comunicarse con destinos fuera y dentro del mismo, atendiendo a los números marcados.

Esta funcionalidad viene heredada de la que ya poseían las centralitas telefónicas y que permite, por ejemplo, discernir entre llamadas de internas en una misma organización y externas. En el caso planteado en este proyecto fin de carrera los planes de marcación se utilizan para discernir entre los siguientes tipos de llamadas:

- Llamadas que tienen como origen los terminales instalados en los ordenadores de usuario
    - Llamadas desde un terminal *SIP* a otro terminal *SIP*. Para alcanzar a un usuario en este sistema es necesario marcar los nueve dígitos que componen su número de teléfono (9683XXXXX).
    - Llamadas desde un terminal *SIP* a una extensión telefónica interna de la UPCT. Este tipo de llamadas se establecen marcando los seis números del destino que queremos alcanzar (3XXXXX). Es decir, si queremos llamar al teléfono analógico de la ETSIT (968325313) desde una extensión en un PC deberemos marcar 325313.
    - Llamadas desde un terminal *SIP* a un teléfono analógico externo a la red telefónica de la UPCT. En este caso se deberá anteponer el '0' al número de nueve cifras con el que se desee contactar.
- Llamadas desde un terminal externo a un terminal *SIP.*
    - Llamada desde una extensión de la UPCT a un terminal *SIP.* Para estos usuarios será suficiente con marcar el '8800' desde su extensión analógica, esperar a recibir tono y marcar las seis cifras que componen el número del usuario (3XXXXX). De este modo, para llamar a cualquier profesor a su extensión *VoIP* se deberá marcar '8800'+'3XXXXX', según sea su número de teléfono.
    - Llamada desde un teléfono conectado a la RTC a un terminal *SIP*. En este caso los usuarios deberán marcar el '968338800', que le da acceso a la línea desde la que se accede al *gateway* para después marcar el número de seis cifras del usuario al que desean llamar (3XXXXX). Es decir, para llamar a cualquier profesor a su extensión *SIP* se marcará: '968338800' + '3XXXXX'.

Una vez expresadas todas las posibilidades de conexión entre usuarios del sistema y usuarios del sistema telefónico convencional, el servidor *SIP* deberá discernir entre los diferentes tipos de llamadas, con el fin de determinar cuales de ellas se dirigen hacia otros usuarios registrados en el servicio y cuáles deben ser enviadas al *gateway*.

La configuración de los planes de marcación se completa con las alternativas que ofrece el *gateway*, que serán descritas en el posteriormente en este mismo documento. En la siguiente imagen se puede observar la configuración realizada teniendo en cuenta un escenario con un *gateway* en la dirección IP 192.168.50.12.

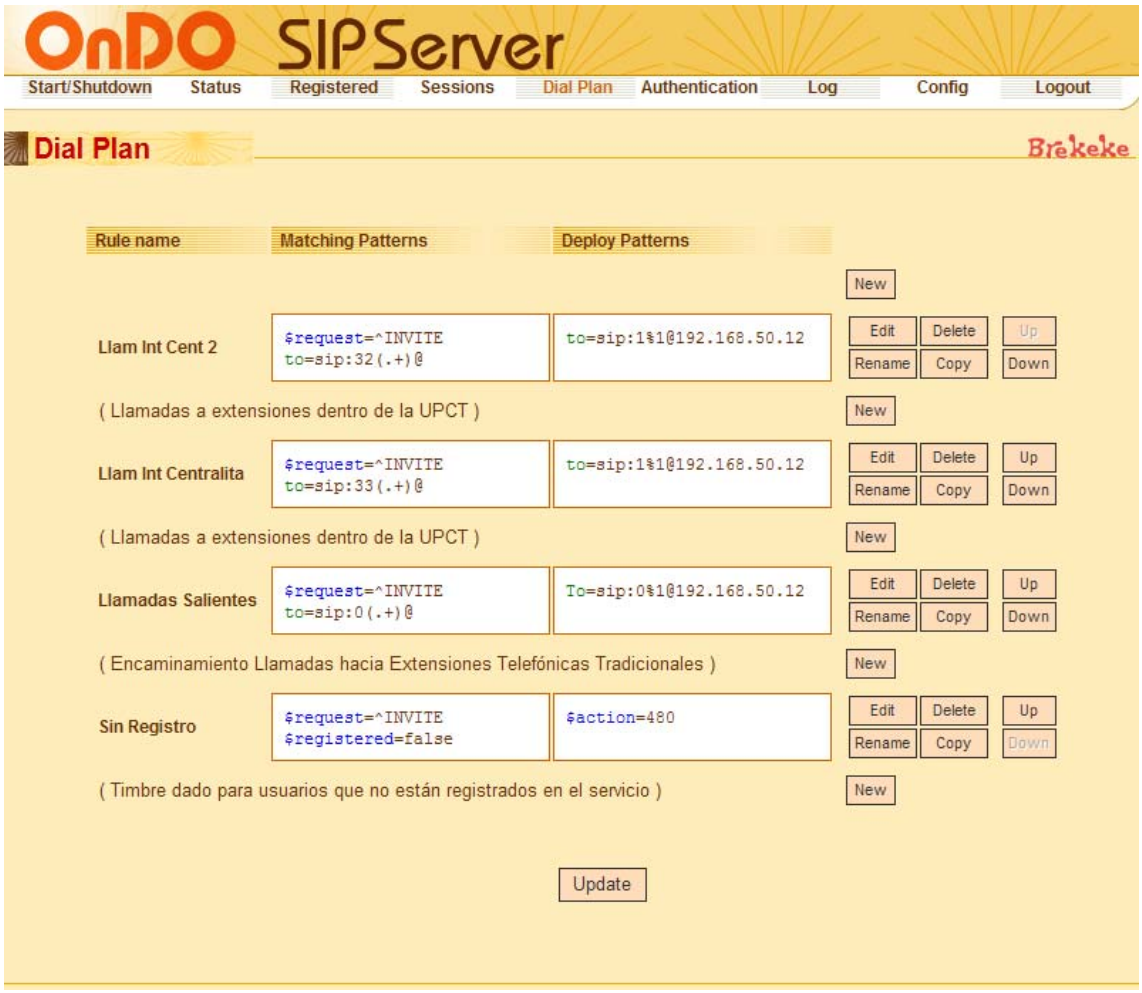

Figura 5-12: Planes de marcación configurados

Como se aprecia en la figura anterior, para dar cabida a todas las necesidades expuestas al comienzo de este punto resulta necesaria la configuración de cuatro planes de marcación, que determinan qué llamadas son externas dependiendo de la dirección de destino. A continuación se explica el funcionamiento de estas cuatro reglas:

- *Llam Int Cent 2.* Esta regla, cuando recibe un mensaje *SIP INVITE*, comprueba si la dirección destino comienza por 33. Si se da este caso es porque el usuario ha marcado un número de seis cifras únicamente, luego esta llamada es externa a una extensión de la UPCT. Lo que se realiza en esta regla es sustituir el 32 por un 1 (cambio requerido por el plan de marcación programado en el gateway), para facilitar la marcación al *gateway* y se reenvía a este.
- *Llam Int Centralita.* Caso análogo al anterior, pero atendiendo a que el comienzo de la dirección destino es 32.
- *Llamadas Salientes.* Este tipo de llamadas son las que tienen como primer dígito marcado el cero en los mensajes de establecimiento de la llamada, en este caso lo único que se hace es reenviarlas al *gateway* del sistema.

- *Sin Registro*. Este regla se da para peticiones que se dirigen hacia usuarios que no están registrados. En este caso se devuelve un tono que indica que el llamado no está disponible temporalmente.

Para más información sobre configuración de planes de marcación utilizando ONDO SIP Server se recomienda la referencia [25].

5.4 Clientes SIP

Los clientes SIP también requieren una configuración determinada. Establecer correctamente los parámetros de la conexión es una tarea clave para llevar a cabo un registro adecuado que permita establecer llamadas de forma adecuada.

Por ello en los apéndices E, F, G y H de este proyecto fin de carrera se muestra al usuario cómo se han de configurar los mismos para obtener un acceso satisfactorio al servicio.

En este punto, sin embargo, se va revisar el proceso de autenticación de los clientes en el servidor SIP, como método de seguridad para evitar la suplantación de los usuarios.

5.4.1 Proceso de Autenticación de los Clientes en el Servidor SIP

Los mecanismos de seguridad que se aplican a una transacción *SIP* siguen el modelo de capas y pueden comenzar aplicando *IPSec* a nivel de red y *TLS* (*Transport Layer Security*) a nivel de transporte, de forma previa los mecanismos que provee *SIP* propiamente.

Sin la aplicación de técnicas de seguridad a nivel de red y de transporte los usuarios deberán confiar en el mecanismo de seguridad que provee *SIP* a nivel de aplicación, provisto a través de un servicio de Autenticación por Compendio (*Digest Autenticación*).

En general, el proceso de autenticación intenta evitar que usuarios maliciosos se registren en nombre de otros y accedan a los servicios destinados a éstos, o usen recursos no autorizados. Este problema es verdaderamente importante cuando entran en juego procesos de tarificación.

Durante el intercambio de señalización entre el agente de usuario *SIP* y el servidor, un atacante podría suplantar a un usuario imitando la identidad real del mismo. La autenticación provee un mecanismo para asegurar que ambos extremos de la comunicación son legítimos.

El proceso de autenticación puede envolver sólo al cliente con respecto al servidor o puede ser un proceso mutuo, de forma que ambos extremos verifican su identidad recíprocamente. El proceso básico es el primero de los enunciados y será el que se explique en este punto. Los principales puntos donde se precisa a autenticación son:

- Registro
- Establecimiento de la sesión / llamada
- Modificación de la sesión / llamada

- Terminación de la sesión / llamada

De cualquier modo, el proceso de autenticación sigue el esquema mostrado en la siguiente ilustración:

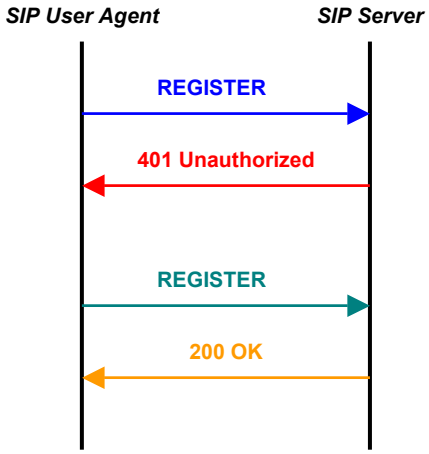

Figura 5-13: Proceso básico de registro de un cliente *SIP* en un servidor

Tal y como se aprecia en el diagrama anterior, el intercambio básico de mensajes se realiza de la siguiente forma:

1. El agente de usuario *SIP* envía una petición de registro al servidor.
2. El servidor devuelve una respuesta de tipo 401 (no autorizado) o 407 (autenticación requerida). En esta respuesta el servidor rellena el campo *WWW-Authenticate* de la cabecera *SIP* e incluye en él un *nonce* (cadena de caracteres dependiente de la implementación del servidor).
3. El cliente *SIP* genera un nueva petición *REGISTER*. En ella crea utiliza el campo *Authorization* y en él incluye, además de los datos proporcionados por el servidor en el mensaje anterior, una respuesta. Dicha respuesta es una función *hash* md5 (el algoritmo también se especifica en la cabecera) del nombre de usuario, la contraseña, *nonce* proporcionado por el servidor, el método *SIP* y la *URI* solicitada.
4. Esta respuesta se verifica en el servidor, que debe reconstruirla para poder verificarla. Este proceso lo hace a partir del *nonce* de la respuesta del punto 2, el nombre de usuario que el agente de usuario pretende registrar, la contraseña que el servidor le tiene asociada a ese terminal y los otros parámetros a los que tiene acceso a partir de la cabecera *SIP* del mensaje.
    a. En caso afirmativo autentica al usuario.
    b. En caso negativo se rechaza al cliente *SIP* que haya pretendido registrarse.

Como se puede apreciar, la principal ventaja de este procedimiento es que permite autenticar a un usuario sin que éste tenga que enviar su contraseña en texto claro. Simplemente basta con que éste envíe la respuesta construida a partir de los

parámetros proporcionados por el servidor y que éste último elemento construya la misma respuesta a partir de la información que posee. Para más información sobre este proceso se recomienda consultar [26] y [27].

5.5 Gateway

Cisco Systems, como empresa líder en el sector de la venta de equipos de telecomunicación entendió, a finales de los 90, que se abría, con la telefonía *IP*, un nuevo negocio y decidió transformar esta oportunidad en una fortaleza.

Por ello diversificó sus productos, de forma que se pudieran añadir a sus routers tarjetas que proporcionasen acceso a nuevos servicios análogamente a las que ya existían para dar interconexión con los diferentes tipos de tecnologías *LAN* y *WAN* existentes.

Esto proporciona una forma rápida de expandir el nuevo mercado de la *VoIP* utilizando varios pilares de forma fundamental:

- Crecimiento de la capacidad de las redes *LAN* y *WAN* actuales. El aumento en el ancho de banda disponible y la mejora en los equipos de conmutación, así como la aparición de mecanismos de calidad de servicio, permite añadir nuevos servicios donde el tiempo es un parámetro clave.
- El abaratamiento del ancho de banda. Existen tecnologías que proporcionan alta capacidad a precios cada vez más reducidos. Esto crea un doble efecto expansivo ya que promociona la creación de redes de alta capacidad en entornos de oficina y residenciales, al mismo tiempo que surgen y se extienden servicios que demandan cada vez un mayor caudal.
- Provisión de crecimiento modular. Se puede comenzar con un router de gama media como sistema para permitir la conectividad entre sucursales e ir insertando tarjetas a éste a medida que se vayan desarrollando nuevos servicios. Esto permite una adaptación total a los requisitos de la empresa donde se desean instalar.

5.5.1 Servicios Provistos por Gateway

Un grupo de estos *routers* modulares lo compone la serie Cisco 2800, que ha sido el elegido para el escenario de esta instalación. Construidos sobre la serie 2600, añaden características cuyo fin es mejorar el comportamiento, la disponibilidad y la fiabilidad. De este modo, se diversifican hacia la mejora en cuestiones importantes actualmente, como son:

- Seguridad.
- Integración de servicios de voz.
- Servicios de *Wireless LAN*.
- Interfaces de alta densidad.

Así, este tipo de routers provee soporte (en ciertos casos a través de tarjetas extensoras o actualizaciones del IOS) para [28]:

- Dos puertos 10/100 *Fast Ethernet* integrados.
- Estructura modular.
- Servicios avanzados a tasas de múltiples T1/E1 ó xDSL.
- Seguridad:
  - Cifrado en placa.
  - Soporte de hasta 1500 túneles *VPN* (*Virtual Private Network*) con el módulo AIM-EPII-PLUS.
  - Control de Admisión a Red (*Network Admission Control - NAC*).
  - Prevención de la intrusión a través de soporte del *Cisco IOS Firewall*.
- Voz:
  - Soporte de conversaciones analógicas y digitales.
  - Soporte opcional de correo de voz.
  - Soporte opcional de *Cisco CallManager Express*.

Figura 5-14: *Cisco 2800 Integrated Services Router*

Como se ha expresado anteriormente, algunas de estas funcionalidades se consiguen a través de tarjetas que se adquieren por separado. Los módulos de provisión telefónica proporcionan la traducción de la señalización y el formato de los datos (de paquetes a una señal analógica de voz y viceversa), permitiendo conectar los *routers* tanto a la RTB como a equipos de telefonía privados (centralitas telefónicas) o a faxes [29]; soportando además la utilización de distintos protocolos de señalización como H.323, *MGCP* o *SIP* y diferentes tecnologías de transporte (*Frame Relay* y *ATM*).

Entre los módulos a añadir exitentes se distingue entre *VIC* (*Voice Inteface Card)* y *VWIC (Voice WAN Interface Card*). La principal diferencia está en que las tarjetas *VIC* permiten conexiones simultáneas de menor escala y se usan para entornos más reducidos que las *VWICs*

Entre las tarjetas *VIC* soportadas existe la posibilidad interfaces *FXS* (*Foreign Exchange Station*), *DID* (*Direct Inward Dial*), *FXO* (*Foreign Exchange Office*), *E&M* (*Ear and Mouth*) ó RDSI básicos, etc.

Estas tarjetas se usan con el *DSP* (*Digital Signal Processor*) para tarjetas de Fax y Voz *PVDM2*, de forma que se pueden proporcionar desde 4 hasta 120 canales

según el tipo y número de módulos incorporados. Las compañías pueden seleccionar el número mínimo de *PVDM2* dependiendo de los canales de voz que requieran y escalar el resultado conforme vayan creciendo. Así, estos *DSP* pueden ser configurados a través del software para una complejidad alta, media o flex. La última de las configuraciones es la incluida por defecto y en este modo, el interfaz de red negocia dinámicamente el códec apropiado para la llamada dependiendo de los *PVDM2* disponibles.

En el escenario donde se realiza la instalación, el equipo que dará servicio a la ETSIT, se ha añadido un interfaz *VIC2-4FXO*, que permite conectar el router a cualquier centralita o incluso a la red telefónica básica.

En nuestro caso se conectará a la centralita de la Universidad Politécnica de Cartagena, actuando únicamente como un *gateway* que permite a cualquier usuario del servicio de *VoIP* contactar con otro usuario que se encuentra en la RTB. Entre las características que puede aportar este equipo como *gateway VoIP* se encuentran:

- Compresión de las cabeceras *RTP* (*cRTP* / *Compressed Real-Time Protocol*).
- Control de admisión de llamadas.
- Mecanismos avanzados de calidad de servicio (*QoS*).
- Soporte a múltiples protocolos de señalización: *H.323*, *MGCP* y *SIP*. A los que hay que añadir *Cisco CallManager Express* 3.1.
- Soporte de múltiples códecs: G.711, G.729, G.729a/b, G.723.1, G.726, G.728, *GSM, GSM-EFR, GSM-ER*.
- Cancelación de eco.
- Supresión de silencios a través de detección de actividad en la voz (*VAD* / *Voice Activity Detection*).
- Soporte de identificación del llamante.
- Configuración de grupos de salto.
- Configuración de planes de marcación.
- Configuración de soluciones *IVR* (*Interactive Voice Response*).
- Etc.

5.5.2 Configuración Realizada

El fichero de configuración que detalla los parámetros configurados se añade en el apéndice I de este documento. De cualquier modo, en este punto se pretende acercar al lector a las principales características de las que se beneficia el servicio de telefonía *IP* implantado.

Los servicios configurados en el mismo son:

- Planes de marcación.
- Expansión de números.
- Lista de preferencia de códecs.

- Grupos de salto (si un puerto de la tarjeta está ocupado desborda al siguiente).
- VAD (*Voice Activity Detection*).

A continuación se explica en qué consisten los parámetros más representativos de los configurados: los planes de marcación y la expansión de números.

5.5.2.1 Planes de Marcación

Un plan de marcación describe el número y el patrón de dígitos que un usuario marca para alcanzar un número de teléfono particular, y queda formado por todos los dígitos se introducen en el sistema, independientemente de su utilidad final.

Los planes de marcación deben cumplir las normas que imponen las redes telefónicas a las que se conectan los *gateways*, puesto que deben permitir alcanzar destinos que se encuentran en éstas; a excepción de redes puramente privadas.

Los planes de marcación se configuran a través de sentencias más pequeñas denominadas *dial peer*s. Estas sentencias proporcionan una configuración similar a rutas estáticas, ya que definen el camino que han de seguir las llamadas dependiendo del llamante o del llamado. Los atributos definidos en estos *dial peer* determinan los dígitos marcados que el *router* toma y reenvía a los dispositivos telefónicos. Se definen dos tipos de *dial peers*:

- *POTS*: Define las características de una conexión de red telefónica tradicional. Su función básica es marcar una cadena determinada en un puerto de voz específico del *router* local, normalmente el que se conecta a la RTB o a la centralita.
- *Voice-network*: Este tipo define las características de una conexión en una red de paquetes. En este caso se utiliza la cadena marcada para decidir el siguiente *router* o equipo al que enviar el paquete. Existen distintos tipos de *dial peers* para redes de paquetes dependiendo de la tecnología de red. Se distinguen los siguientes:
    - *VoIP (Voice over IP)*. Apunta a direcciones IP del *router*/servidor de destino que finaliza la llamada.
    - *VoFR (Voice over Frame Relay).* Apunta al *DLCI (data-link connection identifier*) de la interfaz desde la cual se general la llamada.
    - *VoATM (Voice over ATM)*. Apunta al circuito virtual *ATM* para la interfaz desde la cual se genera la llamada.

En nuestro caso sólo se van a configurar *dial peers* de tipo *POTS* y *VoIP*, puesto que únicamente tenemos dos tipos de redes relacionadas, la red *IP* desde la que toman el servicio de datos los profesores de la ETSIT y la telefónica tradicional, a través de la centralita telefónica de la UPCT.

Para configurar estos *dial peers* basta con entrar al modo de configuración global del *router Cisco* y escribir uno de estos dos comandos:

- Para *dial peers* tipo *POTS*:

  - `dial-peer voice n pots`
  - Donde n es el número de *dial peer* y lo elige el usuario
- Para *dial peers* tipo *VoIP*:
  - `dial-peer voice n voip`
  - Donde n es el número de *dial peer* y lo elige el usuario

Como nota destacar que no pueden existir dos *dial peers* con el mismo identificador. Una vez se definen éstos se deben incluir una serie de parámetros dentro de los mismos, que se explican a continuación tomando como base la configuración realizada en el *gateway* de este proyecto fin de carrera.

5.5.2.2 Dial Peer POTS

El siguiente fragmento está extraído del fichero de configuración del *gateway*, para más información consultar apéndice I.

```
dial-peer voice 1 pots
 destination-pattern ..........
 port 0/0/0:1
```

En el fragmento anterior se aprecian dos atributos especificados, *destination-pattern* y *port*. El primero de ellos es común a todos los *dial peer* y representa la cadena que debe coincidir para ser enviado a través del puerto que indica el parámetro *port*.

En caso anterior se indica que cualquier número de diez dígitos será enviado por el puerto 0 o 1 (si existe desbordamiento – grupo de salto) de la tarjeta *VIC* del *gateway*. Como nota destacar que el carácter '.' es un carácter de escape cuyo significado es cualquier número.

5.5.2.3 Dial Peer VoIP

El siguiente fragmento, al igual que el anterior, está extraído del fichero de configuración del *gateway*, para más información consultar apéndice I.

```
dial-peer voice 2 voip
 destination-pattern 9683.....
 voice-class codec 1
 session protocol sipv2
 session target ipv4:212.128.44.40
 session transport udp
 dtmf-relay h245-alphanumeric
```

En éste, los atributos tienen los siguientes significados:

- *destination-pattern:* la finalidad es la misma expresada para el punto anterior, la comprobación del número marcado para determinar si es el *dial peer* adecuado.
- *voice-class codec 1:* en los *routers Cisco* cabe la posibilidad de definir listas de preferencia de *códecs* cara a la negociación de los mismos con los clientes/terminales *software*. Este comando especifica la lista 1 de las definidas.
- *session protocol sipv2:* los *routers Cisco* traen como predefinida la utilización de señalización H.323 para la señalización de la llamada de *VoIP*, luego es imperativo la necesidad de especificarles *SIP* como protocolo de señalización en caso de no utilizar H.323.
- *session target ipv4:212.128.44.40:* en este caso se especifica el siguiente elemento de la red de paquetes al que deben reenviarse todos los paquetes, en nuestro caso será la dirección del *SIP Server*.
- *session transport upd:* protocolo de transporte especificado. En *SIP* cabe la posibilidad de definir tanto *TCP* como *UDP* como protocolos de transporte.
- *dtmf-relay h245-alphanumeric:* protocolo de gestión de los dígitos marcados.

Para más información sobre configuración de *dial peers* en *routers Cisco* se recomienda la referencia [30]. Únicamente añadir que la supresión de silencios está automáticamente configurada en el sistema, si bien cabe la posibilidad de modificar la configuración para aplicar técnicas más agresivas.

5.5.2.4 Expansión de Números

Esta funcionalidad es similar a la que realizan las centralitas telefónicas y puede ser comparada con los planes de marcación realizados en el servidor *SIP*. Se basa en, a partir de un patrón de números, añadir, quitar o modificar dígitos del mismo, de forma que varíe el número marcado.

Es especialmente útil cuando se combina con planes de marcación y en los *routers Cisco* se ejecuta antes que éstos, paso muy a tener en cuenta en la configuración de los mismos.

El comando para realizar esta tarea es:

```
num-exp 3..... 9683.....
```

El comando anterior convierte una cadena de seis dígitos que comienza por 3 a una cadena que empieza por 9683 más el resto de los 5 dígitos marcados al comienzo.

Capítulo 6
Conclusiones y Líneas Futuras de Trabajo

Este proyecto final de carrera nació como con el objetivo de dotar de un servicio de telefonía IP para la Escuela Técnica Superior de Ingeniería de Telecomunicación (ETSIT).

A nivel teórico, a lo largo del documento que finaliza en estas páginas, se ha revisado la mayoría de las consideraciones que han de tenerse en el proceso de diseño, desde problemas que afectan a la voz hasta los pasos que deberían seguirse para realizar una instalación satisfactoria.

Junto con éstas, se ha continuado exponiendo las características de los protocolos de señalización existentes sobre los que se sustenta cualquier servicio de *VoIP*, comparando los protocolos H.323 (*ITU*) y *SIP* (*IETF*), de los que se han expuesto ventajas e inconvenientes.

Sin embargo, en la segunda parte del documento, desaparece este componente teórico para dar protagonismo a lo que ha supuesto el mayor esfuerzo realizado; la implementación final del servicio.

Como se aprecia en los dos últimos capítulos, este componente se resume en, una vez elegido el protocolo de señalización, buscar, instalar y configurar los tres elementos básicos que componen una instalación: los terminales de usuario (*SIP User Agents* en nuestro caso), el servidor (*SIP Server*, que integra las funciones básicas de registro, *proxy* y redirección) y el *gateway* o pasarela.

Sin embargo, poner en funcionamiento un sistema de *VoIP* no es un proceso sencillo. Parte de las consideraciones van surgiendo a medida que se va desarrollando el proyecto, de forma que aparecen modificaciones que varían el diseño inicial. De hecho una de éstas ha llevado a crear la página web del servicio, cuya necesidad se justificó en el capítulo anterior, y a definir un esquema reglas de utilización para establecer las llamadas, derivado de la necesidad de configurar planes de marcación.

La principal fortaleza de este proyecto fin de carrera es que ofrece a cualquier técnico de sistemas un manual para proveer una solución corporativa del servicio de telefonía *IP* de mínimo coste, basándose en una plataforma *Windows*. Además esta implementación ha quedado abierta en su utilización a cualquier profesor que tenga un número de teléfono asociado en el directorio de la UPCT, no estando limitado únicamente a profesores de la ETSIT. Sin embargo, fortalezas como la economía presupuestaria se tornan en debilidad en otros aspectos, puesto que, como suele resultar común, la solución más económica no es la que aporta mejores características.

De hecho, se han obviado las consideraciones de calidad de servicio (*QoS*) (que se pueden revisar en [31]), las debilidades de seguridad en las comunicaciones (la única seguridad que se provee en este proyecto fin de carrera es la derivada de la autenticación básica *SIP*) y la prestación de servicios avanzados *SIP*, como pueden ser los asociados a la presencia de los usuarios. Para nuevas aproximaciones en esta infraestructura se propone mejorar, por orden de importancia:

- Los aspectos relacionados con el transporte de voz, implantando técnicas de calidad del servicio a nivel de enlace en los *switches* de la red de la ETSIT y evaluando bajo tráfico real su incidencia en las comunicaciones de voz.
- Mejorar la seguridad de las comunicaciones. Aplicando protocolos como *IPSec* a nivel de red o *TSL* a nivel de transporte que sean eficientes ante ataques activos.
- Ofrecer servicios avanzados, como los de presencia *SIP*.
- Minimizar la capacidad de la red de la que se hace uso, mejorando los códigos que se utilizan.
- Aplicar una solución ante los problemas que tienen tanto *SIP* como *H.323* para las llamadas que han de atravesar sistemas *NAT*, lo que proporcionaría conectividad *VoIP* a los laboratorios.
- Mejorar los procesos de almacenamiento de información de las comunicaciones, de forma que se puedan controlar en tiempo real y vía web las llamadas realizadas por cada usuario, indicando destino y duración y aplicando, en su caso, las tarifas correspondientes.

Para lograr estas mejoras resulta evidente que se deben realizar dos tareas fundamentales: programar un servidor y un cliente *SIP* que añadan las características deseadas y obtener el acceso a los códigos necesarios de transformación de voz, adquiriendo las licencias oportunas.

A más largo plazo queda la inclusión de nuevas tarjetas de voz en el *gateway* que amplíen la capacidad del mismo, así como la implantación del servicio de telefonía *IP* a nivel global en toda la Universidad Politécnica de Cartagena.

Apéndice A

Código PHP de Registro en el Servicio de Telefonía IP

El siguiente fragmento de código muestra cómo se ha programado la verificación de las contraseñas (a través del acceso al servidor LDAP) de red Campus y la comprobación de pertenencia del número de teléfono al usuario que se pretende registrar, así como el almacenamiento del número de teléfono, contraseña en el servicio del usuario y su DNI dentro de la base de datos *voip.mdb* de *Microsoft Access*.

El código que proporciona esta funcionalidad se presenta a continuación:

```php
<?php

/********************************* FIN DE SESIÓN ****************************************/
//Continuamos la session
session_start();
//Destruimos el array
$_SESSION = array();
//Destruimos la sesion
session_destroy();
/*********************** REGISTRO EN EL SERVICIO VoIP  ***********************************/
/*    Este fragmento de código administra los datos introducidos en el formulario para
   registarse en el servicio VoIP     */

//Si los campos están llenos se intenta autenticar al usuario
if(isset($_POST["num_telef"]) && isset($_POST["dni"]) && isset($_POST["contrasena_campus"]) && isset($_POST["contrasena_VoIP"])
        && isset($_POST["confirm_cont_VoIP"])){

   // Almacenamos el valor de cada campo del formulario en una variable
   $num_telef = $_POST["num_telef"];
   $dni = $_POST["dni"];
   $contrasena_campus = $_POST["contrasena_campus"];
   $contrasena_VoIP = $_POST["contrasena_VoIP"];
   $confirm_cont_VoIP = $_POST["confirm_cont_VoIP"];

   // Quitamos espacios antes y después de los datos introducidos
   $num_telef = trim($num_telef);
   $dni = trim($dni);
   $contrasena_campus = trim ($contrasena_campus);
   $contrasena_VoIP = trim ($contrasena_VoIP);
   $confirm_cont_VoIP = trim ($confirm_cont_VoIP);

   //Comprobamos número de caracteres de nº teléfono y DNI previo a conexion LDAP
   if(strlen($num_telef)==9){
        if(strlen($dni)==10){
        $_ok = true;
           //Conexión al servidor LDAP para verificar la contraseña y el passwd

           // La variable $ok comprueba si todos los datos están correctos y sirve para decidir si introducir los datos en la base de datos

           // Establecimiento de la conexión al servidor LDAP de la UPCT y verificación de la existencia del usuario y la clave
           $conn = @ldap_connect("212.128.20.213",389);
           if(@$bind_usr = ldap_bind($conn, "cn=".$dni.",dc=Usuarios,dc=upct,dc=es", $contrasena_campus))
           {
              $_SESSION['nombre'] = $dni;
              $_ok = true;
           }
           else
           {
              header("Location:fail1.php");
              $_ok = false;
           }
```

```
            //Comprobación del número de usuarios con ese número de teléfono, sólo habrá error en caso de 0 coincidencias
            switch($num_telefs){
                case 0:
                    $_ok = false;
                    header("Location:fail2.php");
                    break;
                default:
                    break;
            }

            //Comprobación de las contraseñas que han sido introducidas-> Si son distintas se muestra la página de error adecuada
            if($contrasena_VoIP != $confirm_cont_VoIP){
                header("Location:fail3.php");
                $_ok = false;
            }

            //Almacenamiento de los valores en la tabla de la base de datos
            if($_ok){
                $dtbs=odbc_connect('voip','','');
                //Comprobacíon de si existe la entrada en la base de datos para actualizarla con la contraseña y el usuario actuales.
                $sql_select="SELECT password FROM tabla WHERE id='".$num_telef."';";
                $result = odbc_exec($dtbs,$sql_select);
                $consult = odbc_result($result, 1);

                if($consult == "") { //Si no se ha encontado la entrada en la base de datos se introduce sin más
                    $sql="INSERT INTO tabla (id, password, dni) VALUES ('".$num_telef."','".$contrasena_VoIP."','".$dni."');";
                    $rs=odbc_exec($dtbs,$sql);
                }
                else{                   //Si está en la base de datos, se actualiza
                    $sql="UPDATE tabla SET password='".$contrasena_VoIP."', dni='".$dni."' WHERE id='".$num_telef."';";
                    $rs=odbc_exec($dtbs,$sql);
                }

                odbc_close($dtbs);                  // Cierre de la base de datos
                header("Location:registered.php");  // Muestra la página informativa de registro exitoso
            }
        }
        else  // Fallo en el DNI
            header("Location:fail1.php");
    }
    else // Fallo en el número de teléfono introducido
        header("Location:fail2.php");
}
else{
?>
   // Si no están todos los campos del formulario completos se muestra la página web de nuevo
<?
}
?>
```

Nota: Para que el código anterior funcione es necesario descomentar la línea *extensión=phpldap.dll* en el archivo *php.ini* del servidor web.

Apéndice B

Instalación del Plug-in Java en Ondo SIP Server

Las siguientes líneas explican los pasos que se han de realizar para instalar el plug-in de autenticación en *Ondo SIP Server*. Este ejemplo se basa en la existencia de una sola clase que contiene la funcionalidad (*Accounting.class*) que se encuentra en un paquete denominado *com.topict* (en el archivo java deberá existir la línea *package com.topict*).

1. Una vez creado y compilado el *plug-in* en Java, tendremos un archivo de extensión *.class*.
2. Para instalar dicho archivo se deberá insertar la clase creada dentro de la ruta de directorios:

   ONDO-INSTALL-FOLDER/webapps/proxy/WEB-INF/classes/com/topict
3. Una vez se tiene la clase en la ruta anterior. Se debe abrir el archivo *sv.properties* de:

   install_dir\webapps\proxy\WEB-INF\work\sv
4. Una vez abierto el fichero, se deberá de añadir/modificar en él la línea:

   net.usrdir.plugins=com.topict.Accounting
5. Guardar cambios en el archivo y salir.

Como se aprecia, el proceso de instalación es sencillo, si bien puede generar problemas si no se siguen con exactitud los pasos marcados en este documento. Además, se considera interesante mostrar cómo queda el panel de configuración del servidor SIP en este apéndice, de forma que se complemente con la información mostrada hasta ahora.

Cabe destacar que al instalar el programa de autenticación en el servidor *SIP* se realizan automáticamente todos los cambios que son necesarios en su configuración, por lo que solamente es necesario habilitar sobre qué tipo de mensaje se desea establecer la autenticación. Por ello se muestran los siguientes dos ilustraciones.

Ilustración 1: Configuración *SIP* del servidor seleccionada para realizar la autenticación al recibir una petición REGISTER

Ilustración 2: Configuración SIP avanzada, se aprecia que el parámetro *Thru Registration* está habilitado, tarea que se hace automáticamente al instalar el *plug-in*

Apéndice C

Código del Plug-in Java de Autenticación para Ondo SIP Server

El código que se muestra a continuación muestra el proceso por el cual se extrae la contraseña de una base de datos *Microsoft Access* y se entrega ésta al servidor *SIP* para que éste realice el proceso de autenticación.

```
package com.topict;          //Paquete en el que se incluye la clase
                             //para referenciarlo después desde Ondo SIP Server

import com.brekeke.common.*;
import com.brekeke.net.usrdir.*;


import java.sql.*;
import java.io.*;
import java.util.*;
import java.net.*;


public class Accounting implements UserDir
{

//VARIABLES DE CLASE

    Envrnmt env = null;
    Logging log = null;
    LogLevel loglevel = null;

// MÉTODOS DE LA CLASE

    public void init(Envrnmt env, Logging log) throws Exception
    {
        this.env = env;
        this.log = log;

        //Establecimiento del nivel de log
        loglevel = new LogLevel(env.getInt("net.userdir.loglevel.console",
                                LogLevel.LOG_LEVEL_SYSTEM),
                                env.getInt("net.userdir.loglevel.file",
                                LogLevel.LOG_LEVEL_EXCEPTION));

        log.println("Ldap: comienza la conexión",loglevel, LogLevel.LOG_LEVEL_SYSTEM);
    }


    public void close() //Realiza el cierre
    {
        // Se cierra el contexto principal de acceso a LDAP
    }

    public UserRecord lookup(String username, String method, String destination, String authinfo)
    throws Exception
    {
        // Variables del método
        String id=username;
        String password = new String();
        String driver = "sun.jdbc.odbc.JdbcOdbcDriver";
        Connection con = null;
        Statement stmt = null;

        // Objeto que hay que devolverle a ONDO SIP Server
        UserRecord  record = new UserRecord() ;
        record.username = id;
        // Ahora falta introducirle al objeto el parámetros password


// BÚSQUEDA DEL PASSWORD EN LA BASE DE DATOS

        //Dirección donde se encuentra la base de datos
        String sitiobase = "c:\\apache\\htdocs\\voip.mdb";
        String direcc = "jdbc:odbc:Driver={Microsoft Access Driver (*.mdb)};DBQ=" + sitiobase;

        //Cargamos el driver
        try
        {
            Class.forName(driver).newInstance();
        }
        catch (Exception e)
        {
            System.err.println("Falló la carga del driver.\n");
            record.password = null;
            return record;
        }
```

```
        //Abrimos la conexión a la base de datos
        try
        {
            con = DriverManager.getConnection(direcc);
            System.out.println("Conexión completa\n");
        }
        catch(Exception e)
        {
            System.out.println("Problemas conectando con la base de datos\n");
            System.out.println(e.getMessage());

            if (con != null)
            {
                try
                {
                    con.close();
                }
                catch(Exception e2)
                {
                    System.out.println("Conexión fallida, cerramos base datos\n");
                }
            }
        }

        // Realizamos la consulta deseada la base de datos
        try
        {
            String query = "SELECT password FROM tabla where id="+id;
            System.out.println(query);
            stmt = con.createStatement();
            ResultSet result = stmt.executeQuery("SELECT password FROM tabla where id='"+id+"'");

            System.out.println("Consulta realizada\n");

            // Mostrar los resultados obtenidos
            try
            {

                boolean moreRecords = result.next();
                if (!moreRecords)
                {
                    System.out.println("No hay registros");
                    password = null;
                }
                else
                {
                    System.out.println("Hay uno o más registros");
                    password = result.getString("password");
                    System.out.println(password);
                }
            }
            catch (Exception e)
            {
                System.out.println(e);
            }
        }
        catch(Exception e)
        {
            System.out.println("Error de consulta en la base de datos\n");
            password = null;
            e.printStackTrace();
        }
        finally
        {
            System.out.println("Cerrando conexiones\n");
            try {stmt.close();} catch (Exception e){}
            try {con.close();} catch (Exception e){}
        }
        record.password = password;
        return(record);
    }


// LOS SIGUIENTES MÉTODOS SE RELLENAN PARA COMPLETAR EL INTERFAZ
// PERO NO SON UTILIZADOS POR ONDO SIP SERVER
    public boolean append(UserRecord ur)
    {
        //Este método no es usado por ONDO SIP Server. Se provee
        //para que pueda utilizarse desde otras aplicaciones.
        //Añade un objeto UserRecord al directorio

        //Devuelve
        //  -> true si se añadió bien
        //  -> false si no tubo éxito
        return true;

    }

    public boolean remove(UserRecord ur) //Borra un usuario
    {
        //Mismas características que el anterior
        return false;
    }

    public boolean remove(String ur) //Borra un usuario
    {
        //Mismas características que el anterior
        return false;
    }

    public int getCount() //Obtiene el número total de usuarios
    {
        //Devuelve el número de usuarios en el directorio
        return 0;
    }
}
```

Apéndice D

Instalación de un Driver de Acceso a una Base de Datos *Microsoft Access* en *Windows XP*

Durante el capítulo 5 se ha explicado que el sistema de autenticación para el servicio de telefonía *IP* que describe este documento se basa en que el *SIP Server* tome la contraseña de la información que el usuario proporciona en su registro a través de la página web. Estos datos deben quedar almacenados en un repositorio, que podría ser un servidor externo, aunque en esta instalación se ha optado por una base de datos *Microsoft Access* que se encuentra en el mismo servidor de la ETSIT.

A este repositorio es necesario que se acceda por partida doble; en primer lugar, debe permitirse el acceso al mismo desde la web realizada en PHP para que se almacenen los parámetros de usuario representativos (número de teléfono, DNI y contraseña); y en segundo deberá permitirse que el *SIP Server* acceda para recuperar la contraseña y poder crear el *digest* con el que verifica al cliente.

En este apéndice se muestra cómo crear un *driver* de conexión a una base de datos *Microsoft Access* con nombre *voip.mdb*. Este proceso de detalla para *Windows XP* pero es análogo a *Windows 2000 Server* y se ordena según la siguiente secuencia:

1. Acceder al Panel de Control de *Windows XP*.
2. En él, hacer doble *click* en *Herramientas Administrativas.*

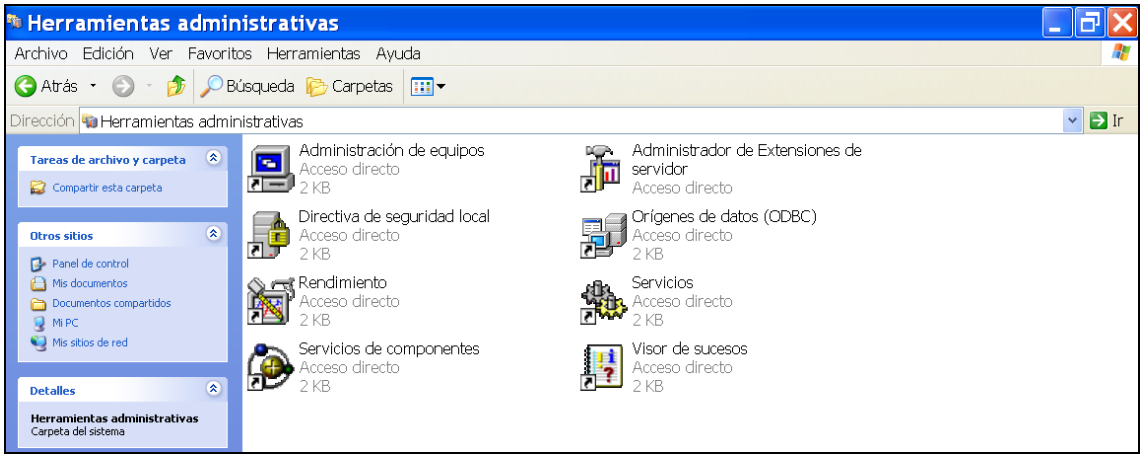

Ilustración 3: Opciones de Herramientas Administrativas en el Panel de Control

3. Acceso a *Orígenes de Datos* (*ODBC*).

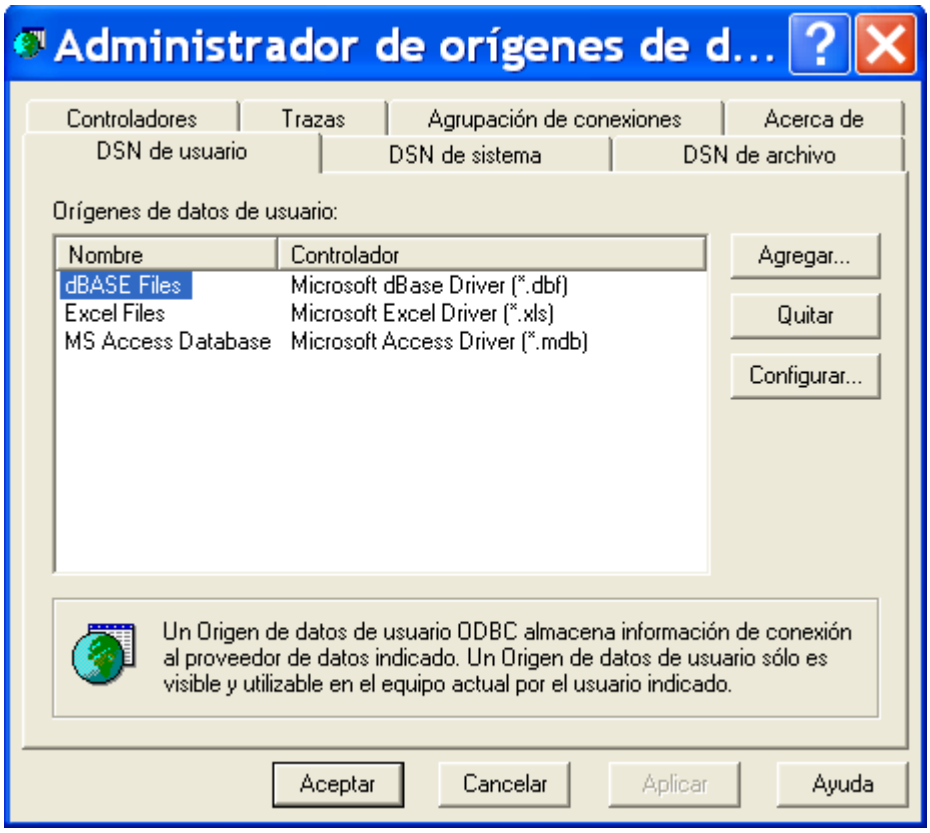

Ilustración 4: Administrado de orígenes de datos

4. Se pulsa sobre *Agregar...* y se elige un *driver para Microsoft Access (*.mdb)*. Después de esto se pulsa sobre *Finalizar*.

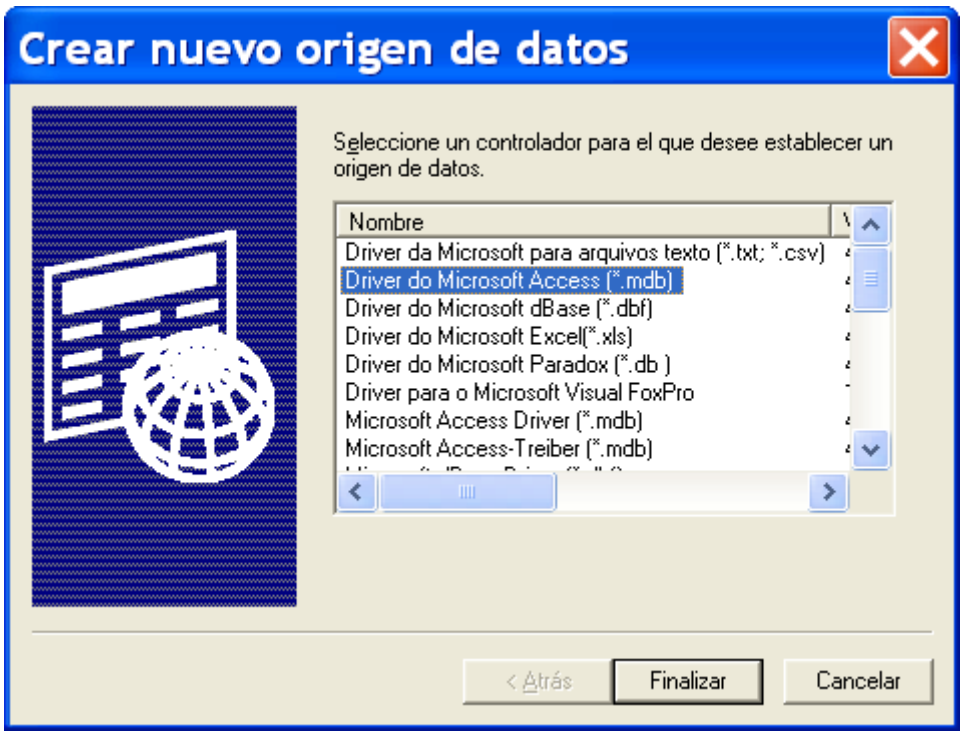

Ilustración 5: Creación de un origen de datos

5. Se crea la base de datos y se configura el driver ODBC tal y como muestra la siguiente figura.

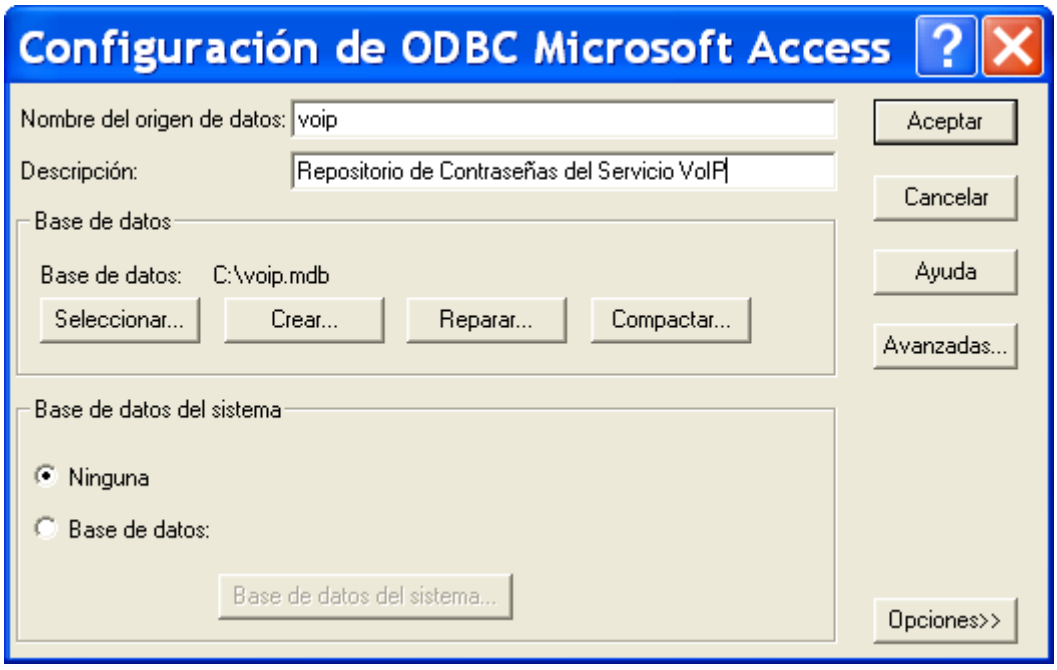

Ilustración 6: Configuración del *driver* ODBC para una base de datos *Microsoft Access*

6. Se pulsa *Aceptar* y la base de datos se añade a las que se encontraban previamente en el administrador de orígenes de datos.

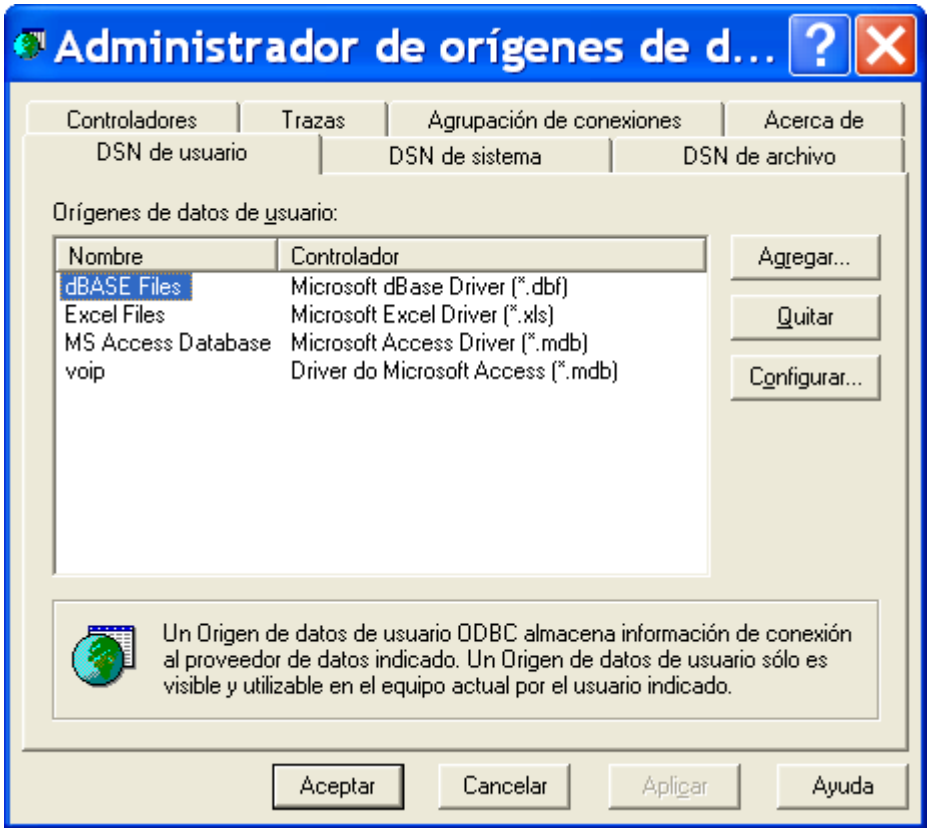

Ilustración 7: Figura que muestra el administrador de orígenes una vez finalizado el proceso de configuración del nuevo driver

Como se puede observar en la secuencia de ilustraciones, el nombre del *driver* añadido es en portugués. Esto es una peculiaridad del sistema operativo donde se ha realizado la creación del mismo, en cualquier caso el procedimiento es el mismo.

Apéndice E

Configuración y Uso de Windows Messenger

CONFIGURACIÓN

Las siguientes imágenes muestran el proceso de definición de parámetros necesario hasta obtener la configuración adecuada que ha de tener este *software* para realizar la conexión al servidor *SIP* instalado. Para realizar ésta se va a tomar como ejemplo un usuario no real del sistema que tiene el número 968328871, que cada usuario deberá cambiar por su verdadero número de teléfono de la UPCT.

Se comienza explicando todo el proceso.

1. Cuando se arranca el terminal se puede apreciar el siguiente interfaz.

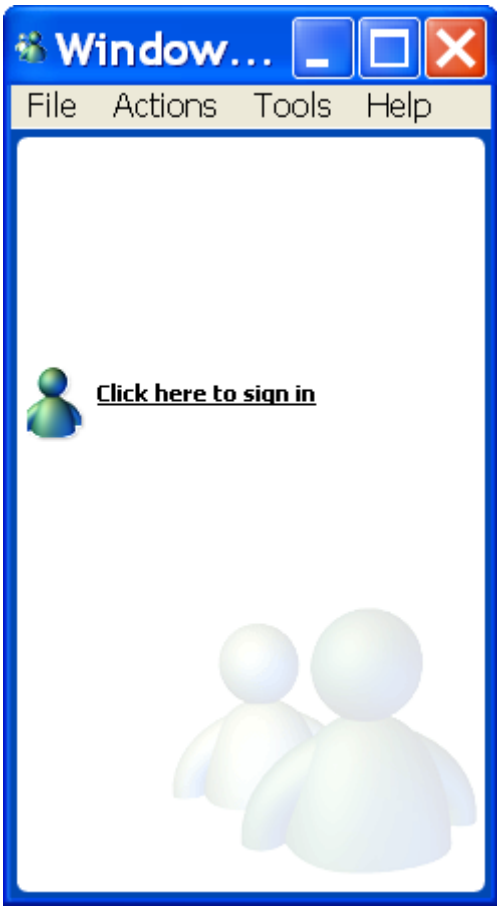

Ilustración 8: Interfaz de usuario de *Windows Messenger*

2. Para configurar el acceso dirigirse al menú *Tools/Options*.

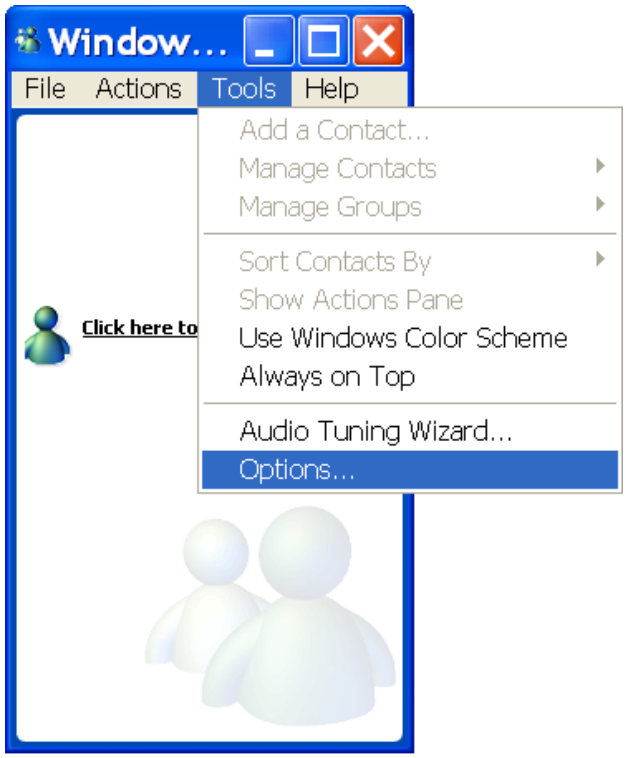

Ilustración 9: Acceso a la configuración del terminal

3. Una vez seleccionado *Options* aparecerá una pantalla en la cual, si presionamos sobre la etiqueta *Accounts*, nos ofrece la siguiente imagen.

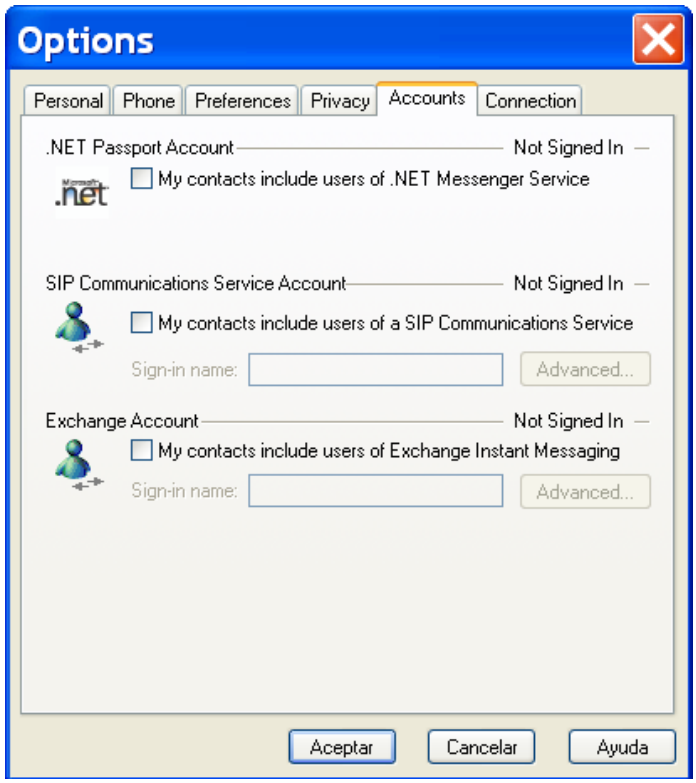

Ilustración 10: Panel de configuración *SIP*

4. Para configurar el sistema deberá seleccionarse en *SIP Communications Service Account* la opción *My contacts include users of a SIP Communications Service*. Una vez echo esto aparece una nueva ventana que debe configurarse de la siguiente forma:

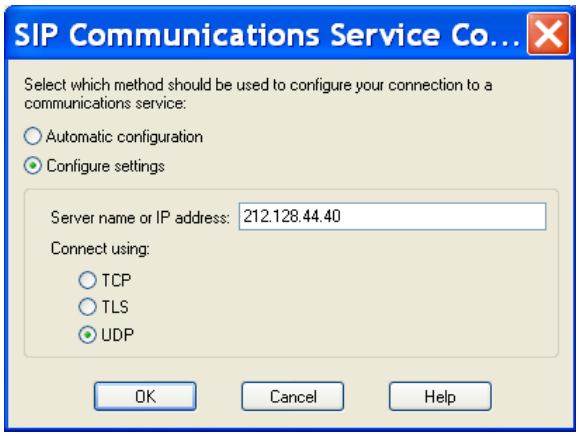

Ilustración 11: Configuración *SIP* del servicio

5. La ventana que aparece debe quedar configurada como se muestra en la ilustración anterior. Si bien aparecerá la como opción seleccionada *Automatic Configuration*, se deberá cambiar por *Configure settings* y seleccionar tanto el protocolo sobre el que va a funcionar *SIP* (en este caso *UDP*) y la dirección *IP* o *DNS* del servidor *SIP*.
6. Una vez terminado se pulsa OK y se vuelve a la ventana anterior, que ahora se debe configurar para acceder al servicio con una dirección SIP adecuada (número_telefono@DirecciónIP_servidorSIP), esto hace que tenga la forma la forma:

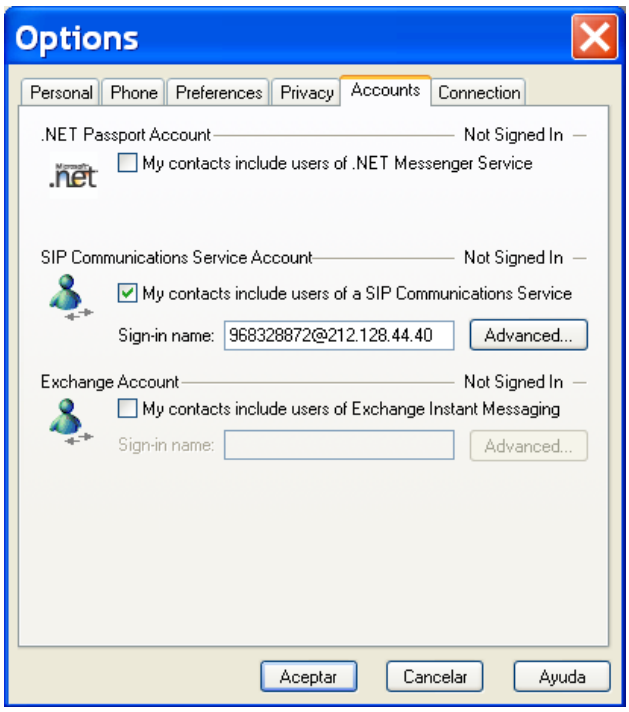

Ilustración 12: Selección del nombre de usuario

Todas las configuraciones anteriores se han hecho utilizando la dirección IP del servidor correspondiente si bien es igualmente válido tomar su nombre DNS (Pcteleco.upct.es ó www.etsit.upct.es). Esta es la configuración del sistema pero aún no se ha accedido al servicio, este punto se explica en el siguiente punto.

ESTABLECIMIENTO DE LA LLAMADA

Para establecer una llamada será necesario, en primer lugar, registrarse en el servicio. Es por ello por lo que se va a comenzar explicando los pasos necesarios para realizar esta primera tarea. Si usted ya conoce como registrarse en el programa, simplemente siga este apartado desde el punto 5.

1. El primer lugar será arrancar la aplicación. Como ya se le ha dado un nombre de usuario en la configuración del sistema realizada en el apartado anterior, mostrará un aspecto parecido al siguiente, dependiendo del número asignado a cada usuario.

Ilustración 13: Arranque de *Windows Messenger* ya configurado

2. Lo normal es seleccionar la opción *Click here to sign in*, pero se puede acceder con otro usuario pulsando *To sign in with a different account, click here*. Si se selecciona la primera de las opciones, aparece una pantalla como la siguiente que hay que rellenar atendiendo a los parámetros de cada usuario, pero siguiendo el patrón general.

Ilustración 14: Configuración de los parámetros del usuario para el registro

3. Una vez rellenados correctamente todos los campos, este cliente *SIP* se queda un intentando el registro, lo que muestra una pantalla que indica *sign in* (registrando).
4. Una vez terminado este registro, se muestra al usuario una pantalla como la siguiente:

Ilustración 15: Interfaz de *Windows Messenger* para un usuario registrado

5. A partir de este punto se procede a explicar el establecimiento propiamente dicho de una conversación. Para ello se debe seleccionar *Start a Óbice Conversation*. Aparecerá una pantalla de la que se ha de seleccionar la pestaña *Others* y se deberá introducir el número de la forma: nºdestino@IPServidor_SIP, tal y como muestra la siguiente ilustración.

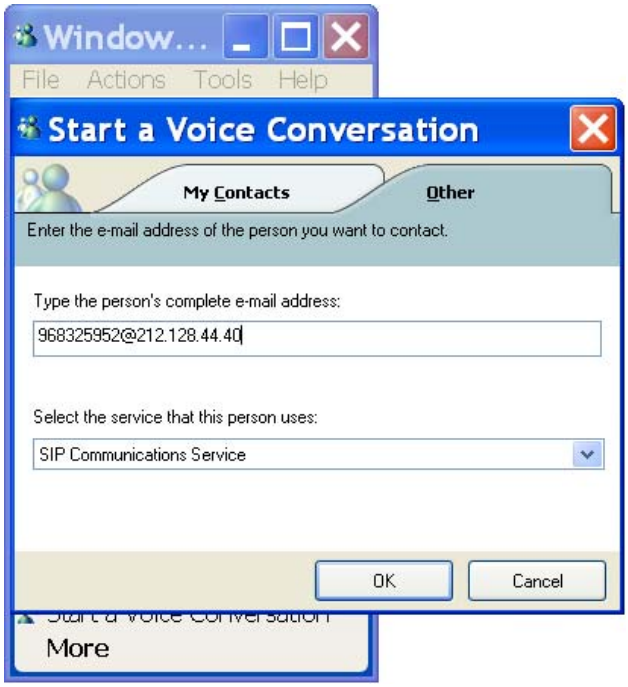

Ilustración 16: Realización de una llamada

6. Cuando se pulsa *OK* aparece una ventana como la siguiente, esperando que el otro extremo de la conversación acepte la llamada.

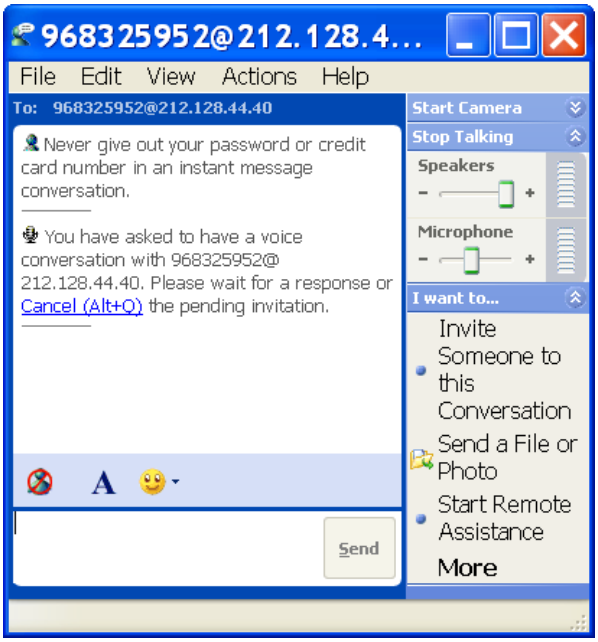

Ilustración 17: Interfaz del extremo llamante esperando que el llamado acepte la conversación

7. En el lado llamado, si se usa este cliente, aparece una ventana como la siguiente.

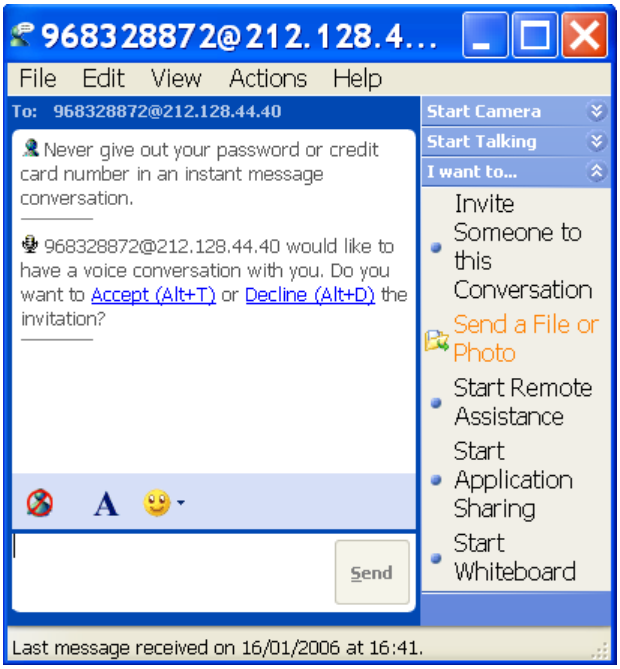

Ilustración 18: Interfaz del extremo llamado pidiendo que éste acepte o rechace la llamada

8. Una vez éste acepta, se queda en ambos terminales una imagen como la siguiente y ya es posible comenzar a utilizar los servicios avanzados de esta aplicación.

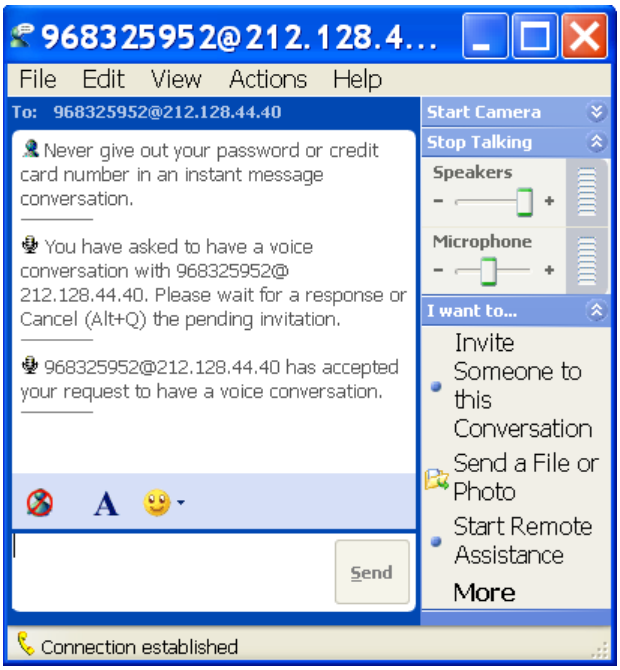

Ilustración 19: Interfaz de conexión establecida

9. Cuando uno de los extremos de la conversación desea finalizar, queda en ambos extremos una imagen como la siguiente:

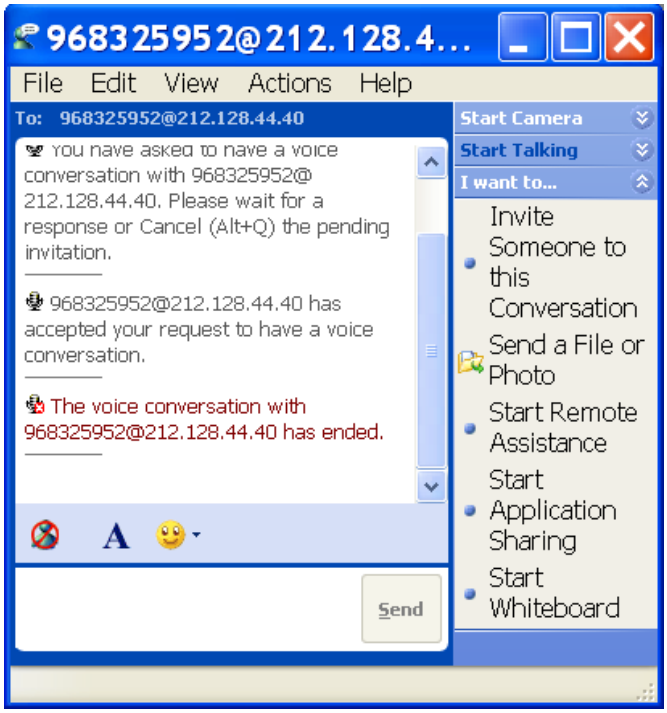

Ilustración 20: Finalización de la llamada

10. En este punto basta con cerrar la ventana y la llamada habrá finalizado

En este manual no se ha explicado como configurar los parámetros de sonido. *Windows Messenger* posee un interfaz bastante sencillo para hacerlo. Siguiendo los pasos del mismo dicha configuración no ha de revestir mayor problema.

Apéndice F

Configuración y Uso de Phoner

CONFIGURACIÓN

Las siguientes imágenes muestran el proceso de definición de parámetros necesario hasta obtener la configuración adecuada que ha de tener este *software* para realizar la conexión al servidor *SIP* instalado. Para realizar ésta se va a tomar como ejemplo un usuario no real del sistema que tiene el número 968328871, que cada usuario deberá cambiar por su verdadero número de teléfono de la UPCT.

Se comienza explicando todo el proceso.

1. Cuando se arranca el terminal se puede apreciar el siguiente interfaz.

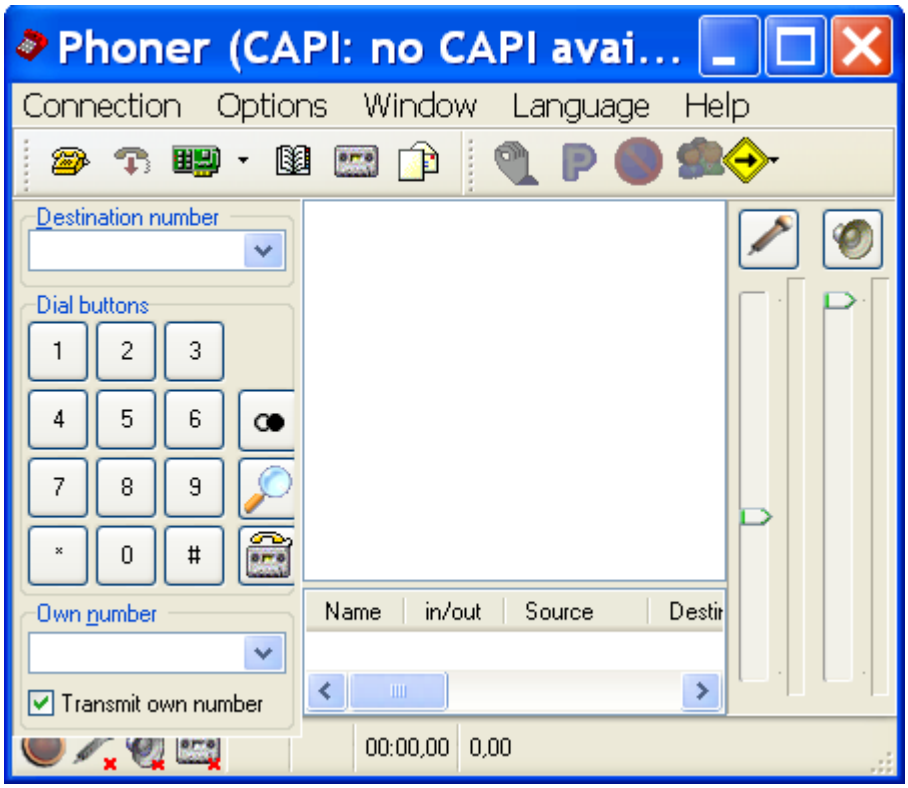

Ilustración 21: Interfaz de usuario de *Phoner*

2. Entonces, acceder al menú *Options*/*Communication* o pulsar sobre el icono que se asemeja a una tarjeta de red, se accederá a una ventana como la que se muestra a continuación.

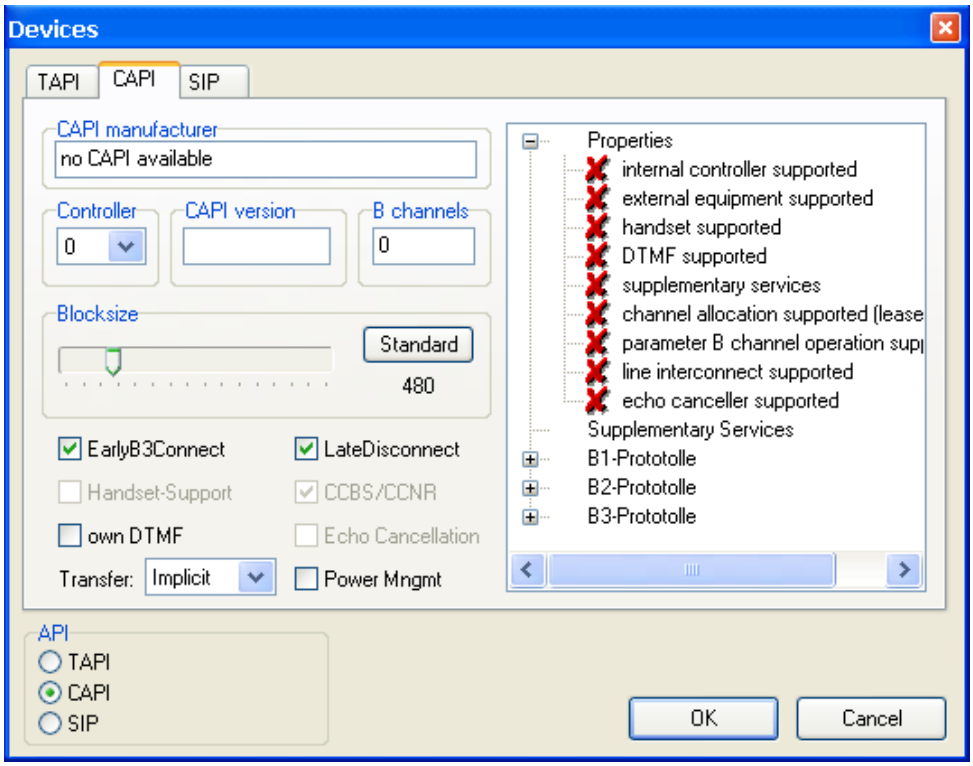

Ilustración 22: Interfaz de configuración del sistema

3. Sin embargo, no se configurará este terminal para ser usado con la funcionalidad *CAPI*, sino con la API *SIP* (ver parte inferior izquierda de la imagen anterior).

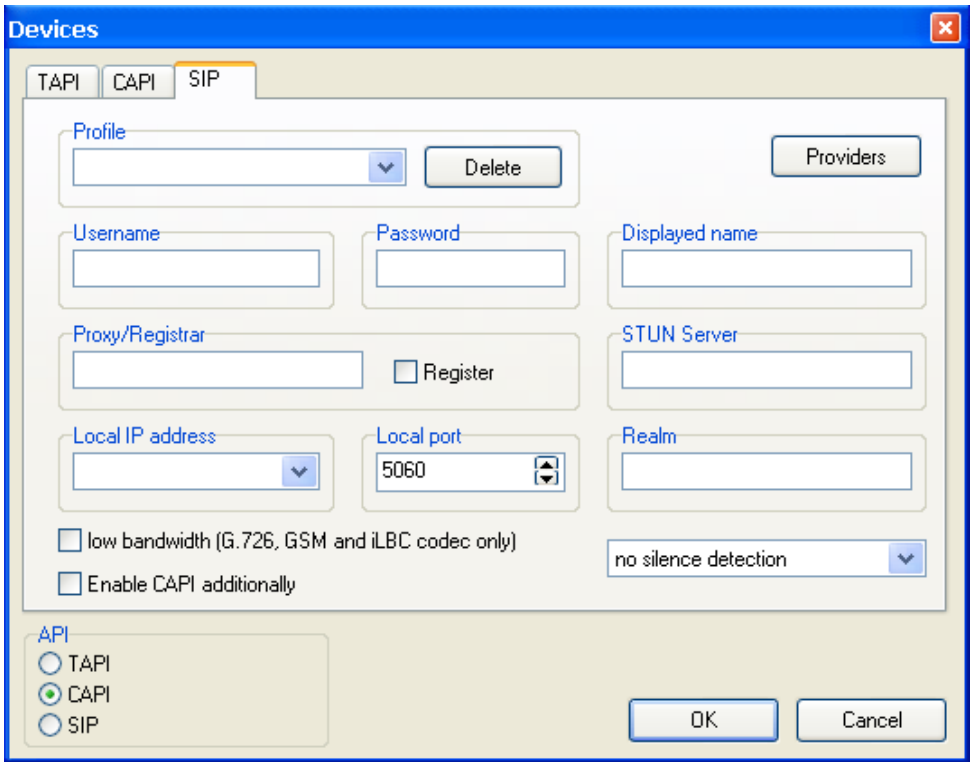

Ilustración 23: Panel de configuración *SIP*

4. En este punto, se deberá configurar el interfaz de la siguiente forma:

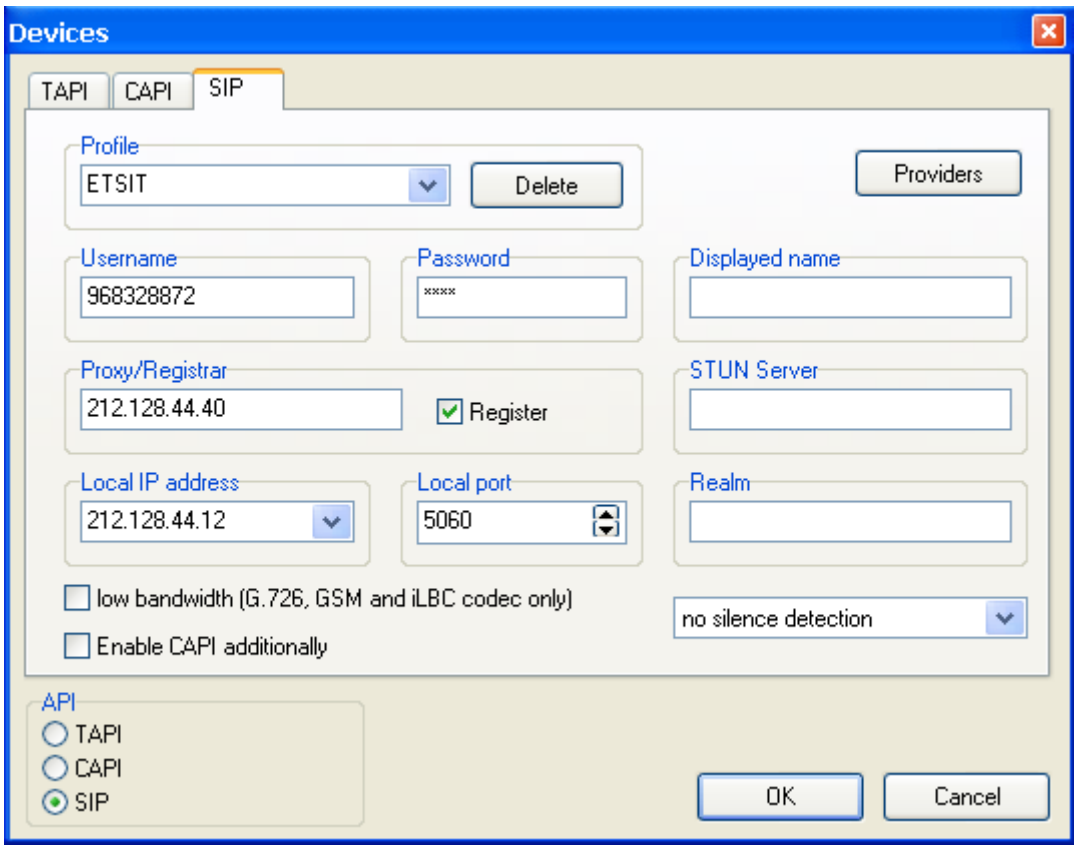

Ilustración 24: Configuración *SIP* de *Phone*r final para acceder al servicio

Además, si se desea se puede sustituir la dirección IP del servidor *SIP Proxy* por su nombre DNS, es decir, cambiar 212.128.44.40 por www.etsit.upct.es o Pcteleco.upct.es.

ESTABLECIMIENTO DE LA LLAMADA

Para establecer una llamada bastará con marcar el número deseado en el campo de texto *Destination Number* y pulsar sobre el botón etiquetado con la figura de un teléfono.

Apéndice G

Configuración y Uso de PhonerLite

CONFIGURACIÓN

Las siguientes imágenes muestran el proceso de definición de parámetros necesario hasta obtener la configuración adecuada que ha de tener este *software* para realizar la conexión al servidor *SIP* instalado. Para realizar ésta se va a tomar como ejemplo un usuario no real del sistema que tiene el número 968328871, que cada usuario deberá cambiar por su verdadero número de teléfono de la UPCT.

Se comienza explicando todo el proceso.

1. Cuando se arranca el terminal se puede apreciar el siguiente interfaz.

Ilustración 25: Interfaz de usuario de *PhonerLite*

2. Como se puede observar el interfaz está completamente en español. Para acceder a los parámetros de configuración bastará con pulsar sobre la pestaña etiquetada con este nombre.

Ilustración 26: Panel de configuración

3. En este punto, se deberá configurar el interfaz de la siguiente forma y presionar el botón guardar.

Ilustración 27: Configuración SIP de *PhonerLite* final para acceder al servicio

Además, si se desea se puede sustituir la dirección IP del servidor SIP Proxy por su nombre DNS, es decir, cambiar 212.128.44.40 por www.etsit.upct.es o Pcteleco.upct.es.

ESTABLECIMIENTO DE LA LLAMADA

Para establecer una llamada bastará con marcar el número deseado pulsando con el ratón sobre el número de en el interfaz de la aplicación o a través del teclado del PC, y presionar después sobre la tecla que muestra un auricular en verde.

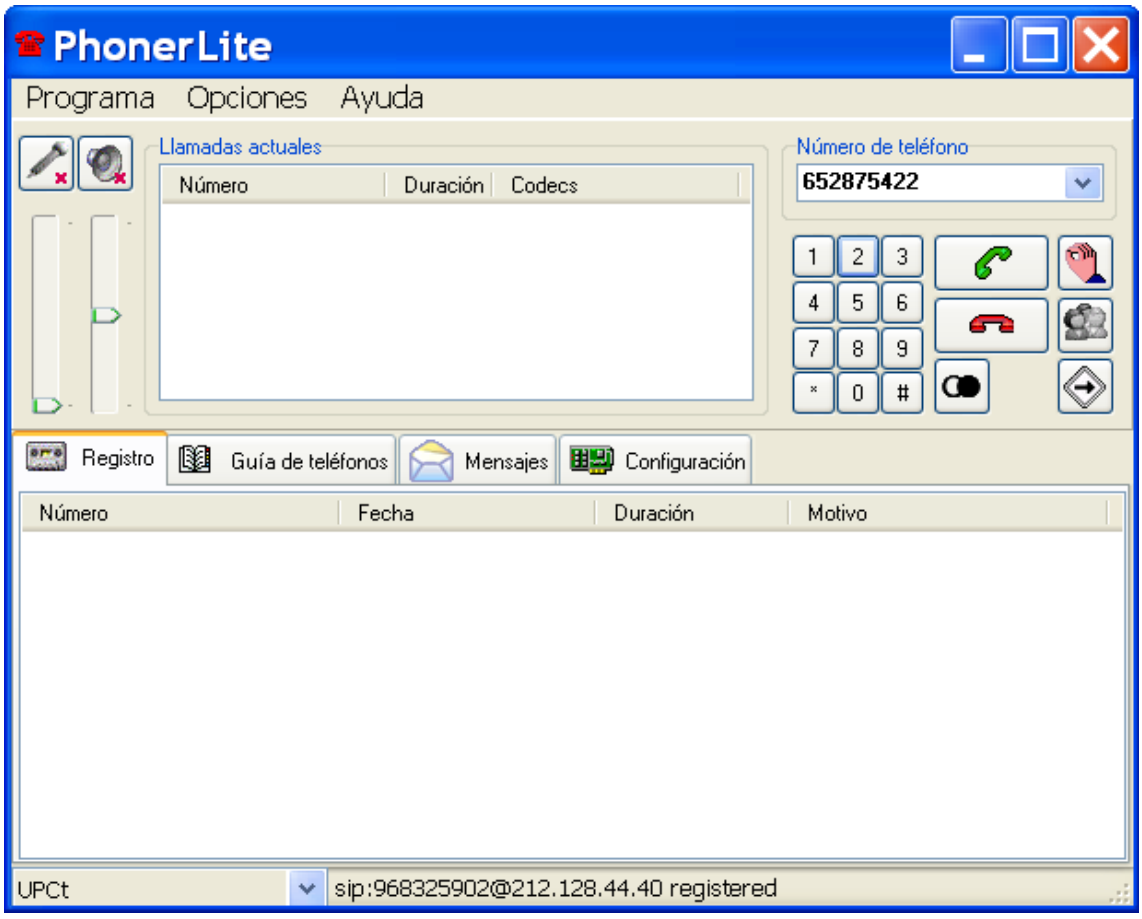

Ilustración 28: Realización de una llamada con *PhonerLite*

Apéndice H

Manual de Configuración y Uso de X-Lite

CONFIGURACIÓN

Las siguientes imágenes muestran el proceso de definición de parámetros necesario hasta obtener la configuración adecuada que ha de tener este *software* para realizar la conexión al servidor *SIP* instalado. Para realizar ésta se va a tomar como ejemplo un usuario no real del sistema que tiene el número 968328871, que cada usuario deberá cambiar por su verdadero número de teléfono de la UPCT.

Se comienza explicando todo el proceso.

1. Cuando se arranca el terminal se puede apreciar el siguiente interfaz. Una vez arrancado la primera acción que pide es la configuración de los parámetros de sonido, se recomienda no descuidar dicha configuración porque este programa es muy sensible a la misma. De cualquier modo se puede repetir este proceso en otro momento.

Ilustración 29: Interfaz de usuario de *X-Lite*

2. Pulsando sobre el botón indicado se accede al menú, el cual, por ser la primera vez que se arranca la aplicación dirige al usuario de forma directa a la configuración de los parámetros de la cuenta.

Ilustración 30: Acceso al menú de la aplicación

3. Las siguientes figuras muestran la ventana de parámetros que se vacía y rellena de forma adecuada para un usuario con un número de teléfono 968328872. Los parámetros a personalizar por parte del usuario con respecto a los mostrados en la ilustración son el nombre de éste (su número de teléfono en 9 cifras) y la contraseña.

Ilustración 31: Panel de configuración SIP

Ilustración 32: Panel de configuración SIP relleno con los parámetros del servicio de telefonía IP de la ETSIT

4. Una vez configurados todos los parámetros se cierra la ventana pulsando sobre el aspa de la parte superior derecha y se aprecia en el interfaz que el cliente se está registrando.

Ilustración 33: Proceso de registro del cliente en el servicio

5. Cuando finaliza este proceso se puede visualizar una ventana como la siguiente.

Ilustración 34: Usuario registrado

ESTABLECIMIENTO DE LA LLAMADA

Para establecer una llamada bastará con marcar el número deseado atendiendo a los criterios que se establecen en la página web del servicio, se podrá realizar pulsando con el ratón sobre cada uno de los números seleccionados o utilizando el teclado numérico. La siguiente figura muestra el proceso.

Ilustración 35: Marcación de un número

Como nota destacar que el interfaz no cambia su fondo a naranja de forma predeterminada cuando el cliente se registra en el servicio, sino que ha de realizarse de forma manual. Los pasos para conseguir cambiar el interfaz son:

1. Acceder al Menú
2. Pulsar sobre *User Settings*.
3. Seleccionar *Features*
4. Y cambiar la propiedad *Use Orange Background* a *Yes*

Apéndice I

Archivo de Configuración del Gateway

Las siguientes líneas muestran las características de la configuración del *gateway* del sistema de telefonía IP a través de la visualización del fichero de configuración del equipo. Para más información consultar el punto 5.5.

```
Building configuration...

Current configuration : 1578 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
!
no network-clock-participate aim 0
no network-clock-participate aim 1
no aaa new-model
ip subnet-zero
!
ip cef
!
voice-card 0
 no dspfarm
!
voice service voip
 redirect ip2ip
 sip
 no call service stop
!
!
```

```
voice class codec 1
 codec preference 1 g711ulaw
 codec preference 2 g711alaw
 codec preference 3 g729r8 bytes 40
 codec preference 4 g723r63 bytes 96
 codec preference 5 g726r16 bytes 80
!
!
interface FastEthernet0/0
 ip address (OMITIDO) 255.255.255.0
 duplex auto
 speed auto
 no shutdown
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip classless
ip http server
no ip http secure-server
!
!
control-plane
!
!
voice-port 0/0/0
!
voice-port 0/0/1
!
voice-port 0/0/2
!
voice-port 0/0/3
!
dial-peer voice 1 pots
 destination-pattern ..........
 port 0/0/0:1
!
```

```
dial-peer voice 2 voip
 destination-pattern 9683.....
 voice-class codec 1
 session protocol sipv2
 session target ipv4:192.168.50.10
 session transport udp
 dtmf-relay h245-alphanumeric
!
dial-peer voice 3 pots
 destination-pattern ....
 port 0/0/0:1
!
num-exp 3..... 9683.....
num-exp 1.... ....
sip-ua
 retry invite 3
 retry response 3
 retry bye 3
 retry register 3
 timers expires 300000
 sip-server ipv4:192.168.50.10
!
!
line con 0
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
!
end
```

Bibliografía

[1] Secretaría de Estado de Telecomunicaciones y para la Sociedad de la Información, "*Resolución de 30 de junio de 2005*".

http://www.cmt.es/busca/Serv/document/Legislacion_nacional/Basica_Telecomunicaciones/Numeracion/pdf/RE-05-08-18.pdf

[2] Implementation of QoS Provisioning System for Voice over IP

http://students.cs.tamu.edu/zbmai/publication/rtas_2002_wang.pdf

[3] H. Schulzrinne y J. Rosenberg, "*Signaling for Internet telephony*," en *International Conference on Network Protocols (ICNP)*, (Austin, Texas), Oct. 1998.

[4] Jonathan Davidson, James Peters, "*Fundamentos de Voz sobre IP*", Ediciones Cisco Press, 2001

[5] H.323 Protocols Suite

http://www.protocols.com/pbook/h323.htm

[6] SIP, Session Initiation Protocol

http://www.networksorcery.com/enp/protocol/sip.htm

[7] *Guide to Cisco Systems' VoIP Infrastructure Solution for SIP*

http://www.cisco.com/en/US/tech/tk652/tk701/technologies_configuration_guide_chapter09186a00800eadee.html#xtocid4

[8] IETF, "*Uniform Resource Identifiers (URI): Generic Syntax*", *RFC* 2396, 1998

http://www.faqs.org/rfcs/rfc2396.html

[9] SIP Messages and Methods Overview

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/rel_docs/sip_flo/preface.htm

[10] Scott Keagy, "*Integración de Redes de Voz y Datos*", Ediciones Cisco Press, 2001.

[11] Radvisión, "*Session Initiation Protocol (SIP)"*, 2005

http://www.sipforum.org/component/option,com_docman/task,view_category/Itemid,75/order,dmdate_published/ascdesc,DESC/subcat,0/catid,13/limit,10/limitstart,10/

[12] Inter-Asterisk EXchange (IAX) Version 2
http://www.cornfed.com/iax.pdf

[13] SIP Product List

http://www.pulver.com/products/sip/

[14] voip-info.org

http://www.voip-info.org/wiki/

[15] Fomine Real Time Communitacion Server

http://www.fomine.com/rtc-server.html

[16] PBXnSIP

http://www.pbxnsip.com/products.php

[17] Swyx Server

http://www.swyx.com/swyxware/product.html?product=18

[18] CommuniGate Pro

http://www.stalker.com/content/solutions.htm

[19] Brekeke Ondo SIP Server

www.brekeke.com

[20] CommuniGate Systems

http://www.stalker.com/content/pricing_single_server.htm

[21] Ejecutar Windows Messenger y MSN Messenger 5.0 en Windows XP

http://support.microsoft.com/?kbid=330117

[22] Phoner

http://www.phoner.de/index_en.htm

[23] Counter Path. User Guides & Product Specifications

http://www.xten.com/index.php?menu=UserGuides

[24] Ondo SIP Server. Authentication Plug-in Developer's Guide
http://www.brekeke-sip.com/download/oss/oss_authplugin_e.pdf

[25] Ondo SIP Server. Tutorial – Dial Plan
http://www.brekeke-sip.com/download/oss/oss_tutorial_dialplan_e.pdf

[26] RFC 2617. HTTP Authentication: Basic and Digest Access Authentication
http://www.faqs.org/rfcs/rfc2617.html

[27] Study of Digest Authentication for Session Initiation Protocol (SIP)
www.site.uottawa.ca/~bob/gradstudents/DigestAuthenticationReport.pdf

[28] Cisco 2811 Integrated Services Router
http://www.cisco.com/en/US/products/ps5881/index.html

[29] IP Communications Voice/Fax Network Module Datasheet
http://www.cisco.com/en/US/products/hw/modules/ps5365/products_data_sheet09186a0080191d41.html

[30] Voice Dial Plan Considerations
http://www.cisco.com/en/US/products/hw/routers/ps1904/products_configuration_guide_chapter09186a008007dc40.html

[31] Juan Antonio Martínez León, *"Implantación y Estudio de Técnicas de Servicios Diferenciados en Telefonía IP sobre Redes Ethernet"*, Proyecto Fin de Carrera, Octubre 2003.