

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN  
UNIVERSIDAD POLITÉCNICA DE CARTAGENA



*Proyecto Fin de Carrera*

## **Estudio y Desarrollo de Soluciones VoIP para la Conexión Telefónica entre Sedes Internacionales de una misma Empresa.**



AUTOR: Pedro Sánchez Sánchez  
DIRECTOR: Cristina López Bravo  
TUTOR: Pablo Pavón Mariño

Febrero / 2006

# Índice de contenido

1	Introducción.....	6
2	Fundamentos Teóricos.....	7
2.1	Introducción.....	7
2.2	Comunicaciones de voz sobre redes IP.....	7
2.2.1	Historia.....	7
2.2.2	Escenarios.....	7
2.2.3	Ventajas e Inconvenientes.....	9
2.2.4	Transporte de la Voz en redes IP.....	9
2.3	Protocolo H.323.....	12
2.3.1	Los Componentes de la Arquitectura H.323.....	13
2.3.2	Control y Señalización de H.323.....	15
2.3.3	Llamadas y Envío de Mensajes H.323.....	17
2.4	Protocolo SIP.....	19
2.4.1	La Pila de Protocolos de SIP.....	20
2.4.2	Elementos de una Arquitectura SIP.....	20
2.4.3	Mensajes SIP.....	21
2.4.4	Funciones de SIP.....	22
2.4.5	Ejemplo de Operación SIP.....	27
2.5	SIP frente a H.323.....	27
2.6	Codecs.....	28
3	Descripción de las Herramientas.....	29
3.1	GNU/Linux Kubuntu 5.10 Brezzy Badger.....	29
3.2	C-Kermit.....	30
3.3	SER (SIP Express Router).....	30
3.4	Liphone.....	31
3.5	Router Cisco 1761 con VIC FXS de 2 Puertos y VIC FXO de 2 puertos.....	32
3.5.1	Router Cisco 1761.....	32
3.5.2	VIC FXS y VIC FXO.....	33
3.6	Gateway GSM BossLan.....	33
4	Desarrollo del proyecto.....	35
4.1	Descripción del problema. Soluciones propuestas.....	35
4.2	Configuración del Puesto del Trabajo. Llamada entre dos Puertos FXS del Router Cisco.....	39
4.2.1	Preparación del Puesto de Trabajo en el Laboratorio.....	39
4.2.2	Instalación de C-Kermit. Mensajes de Inicio y Configuración Básica del Router.....	40
4.3	Llamada entre Puertos FXS de la misma Tarjeta. Primer Acercamiento a los Dial-Peers.....	45
4.4	Pasarela desde y hacia la RTC mediante GSM y H.323.....	47
4.5	Pasarela desde y hacia la RTC mediante GSM y SIP. Inclusión de LAN.....	50
4.5.1	SIP Express Router (SER). Instalación y Configuración.....	52
4.5.2	Liphone. Instalación y Configuración.....	58
4.5.3	El Router Cisco 1761 como agente de usuario. Cambios en los dial-peers SIP.....	67
4.5.4	El Proceso de una Llamada SIP.....	70
5	Análisis de Viabilidad del Proyecto de Inversión.....	78
5.1	Descripción del Proyecto.....	78
5.2	Datos Económicos del Estudio.....	78
5.2.1	Datos Utilizados en la Proyección de los Estados Contables.....	78
5.2.2	Financiación del Proyecto.....	78
5.2.3	Amortización y Tarifas de las Distintas Migraciones.....	81
5.3	Conclusiones del estudio.....	87
6	Conclusiones y Líneas Futuras.....	88
7	Bibliografía y Referencias.....	89
7.1	Bibliografía.....	89
7.2	Referencias.....	89



## Índice de ilustraciones

Escenario 1.....	8
Escenario 2.....	8
Escenario 3.....	8
Escenario 4.....	9
Escenario 5.....	9
Formato de trama RTP.....	10
Tipo Sender Report.....	11
Pila de protocolos de H.323.....	12
Componentes de la arquitectura H.323.....	13
Señalización de H.323.....	15
Inicio de llamada H.323.....	17
Establecimiento del canal de control de H.323.....	18
Comienzo de la llamada H.323.....	18
Diálogo H.323.....	19
La pila de protocolos SIP.....	20
Mensajes para la resolución de direcciones con SIP.....	22
Establecimiento de la sesión con y sin gatekeeper.....	23
Negociación de contenidos.....	23
Cambio de parámetros de sesión.....	24
Terminación y cancelación de sesión.....	25
Control de llamada.....	25
QoS con SIP.....	26
Nuevos servicios SIP.....	26
Ejemplo de operación SIP.....	27
Una captura de Linphone en funcionamiento.....	31
Dos modelos de la serie 1700 de Cisco.....	32
Gateway GSM BossLan.....	33
Coste nulo entre delegaciones.....	35
Llamadas entre comerciales y oficinas.....	36
Llamadas entre comerciales y clientes de otros países.....	36
Pasarela VoIP H.323 desde y hacia la RTC vía GSM con teléfonos convencionales.....	37
Pasarela VoIP SIP desde y hacia la RTC vía GSM con teléfonos convencionales y ordenadores personales.....	38
Conexión del puesto de trabajo.....	39
Llamada entre oficinas simulada con un sólo router.....	45
Pasarela PSTN con H.323.....	47
Simulación de una LAN y teléfonos de una oficina.....	51
Conexión para la simulación con SIP.....	52
Pantalla principal de Linphone.....	60
Configuración de red en Linphone.....	61
Configuración del hardware de sonido en Linphone.....	62
Configuración del protocolo SIP en Linphone.....	63
Configuración de los codecs utilizados y su orden en Linphone.....	64
Pestaña de estado de Linphone.....	65
Pestaña para enviar tonos DTMF con Linphone.....	66
Pestaña de contactos de Linphone.....	67
Envío de mensajes de una llamada SIP entre todos los componentes de la arquitectura.....	77
Comparación Coste-Inversión de los tipos de migración.....	84
Gráfico del coste telefónico sin migración.....	84
Comparación entre pagos por préstamo de inversión, coste telefónico y ahorro con la migración	

total.....	85
Comparación entre pagos por préstamo de inversión, coste telefónico y ahorro con la migración parcial.....	85
Comparación entre ahorro y coste telefónico con migración total.....	86
Comparación entre ahorro y coste telefónico con migración parcial.....	86

## Índice de tablas

Comparativa de los codecs más comunes.....	28
Inversión de la migración total.....	79
Financiación de la migración total.....	79
Inversión de la migración parcial.....	79
Financiación de la migración parcial.....	79
Préstamo para la financiación de la migración total.....	79
Préstamo para la financiación de la migración parcial.....	80
Amortización del préstamo de la migración total.....	80
Amortización del préstamo de la migración parcial.....	80
Resumen de costes sin aplicar el PFC.....	81
Resumen de costes aplicando migración parcial.....	82
Resumen de costes de la migración total.....	83
Tiempos de amortización con financiación propia.....	87

# 1 Introducción.

Este Proyecto Fin de Carrera describe los métodos de estudio y desarrollo de la migración a VoIP de la empresa **SunConexion International Trade Company**. Esta empresa se dedica al negocio inmobiliario turístico residencial y ha planteado este proyecto previendo la apertura de dos sucursales, una en Alemania y otra en el Reino Unido, para captar el porcentaje de mercado extranjero que desea adquirir una vivienda en el levante español, por su clima privilegiado.

Tras la apertura de todas las delegaciones, la compañía estima tener un considerable gasto telefónico, por el alto importe de las llamadas internacionales así como el ingente tráfico generado por una atención al cliente de alta calidad desde pie de obra. Así, se estudiarán las diversas alternativas tecnológicas que posibiliten la eliminación o reducción de dicho coste telefónico y se presentarán soluciones desde el punto de vista técnico y el económico.

El documento se divide en 6 capítulos, de los que el primero es esta introducción.

El capítulo 2, "*Fundamentos Teóricos*", se encarga de revisar los fundamentos de las tecnologías utilizadas en el desarrollo del presente proyecto, describiendo los conceptos de las tecnologías de transporte de voz sobre redes de datos, su historia, ventajas e inconvenientes, centrándonos en la Voz sobre IP (VoIP) y sus diferentes escenarios de aplicación, así como los dos protocolos más utilizados, H.323 y SIP, para proporcionar este servicio.

En el tercer capítulo, "*Descripción de las Herramientas*", nos encargaremos de los equipos utilizados en las simulaciones de laboratorio. Hablaremos del sistema operativo utilizado en el PC y del *software* instalado ( C-Kermit, Linphone y el servidor SER), del *router* Cisco y el *Gateway* GSM.

El capítulo 4, "*Desarrollo del Proyecto*", trata detalladamente todas las infraestructuras y simulaciones realizadas en el laboratorio, así como la instalación y configuración de las herramientas descritas en el capítulo anterior.

En el capítulo 5, "*Análisis de Viabilidad del Proyecto de Inversión*", evaluaremos la solución propuesta desde un punto de vista económico. Este aspecto es vital para demostrar la rentabilidad del proyecto y que la inversión a realizar para la migración a VoIP de la empresa se puede amortizar a muy corto plazo.

Por último, el sexto capítulo se encarga de mostrar las conclusiones y líneas futuras del proyecto, relacionando todos los capítulos anteriores con la situación real del mercado y sus tendencias.

## **2 Fundamentos Teóricos.**

### **2.1 Introducción.**

En este capítulo sentaremos las bases teóricas para la correcta comprensión del resto del documento. Abordaremos los conceptos de Voz sobre IP (VoIP), los protocolos H.323 o SIP y otros aspectos y protocolos relacionados directamente con ellos.

En el proyecto se ha desestimado la inclusión de otras tecnologías de transporte de voz como VoFR (Voz sobre Frame Relay) o VoATM (Voz sobre ATM) al querer ofrecer la solución más conocida y ampliamente utilizada, gracias a una mayor implantación de las redes IP en las PYMEs.

### **2.2 Comunicaciones de voz sobre redes IP.**

#### **2.2.1 Historia.**

La historia de las tecnologías de transporte de voz comienza con la invención del teléfono en 1860 por Alexander Graham Bell. A partir de entonces, las redes telefónicas y, posteriormente las redes de datos, evolucionaron separadamente, como se ve en la siguiente cronología:

Evolución de la Red Telefónica:

- 1860 → Invención del Teléfono.
- 1920 → Red Telefónica Analógica.
- 1970 → Red Digital Integrada.
- 1980 → SDH, Fibra Óptica.
- 1990 → RDSI, RDSI BA(ATM)
- 2000 → xDSL

Evolución de las Redes de Datos:

- 1970 → Redes Propietarias.
- 1980 → Aparición de Internet.
- 1990 → LAN de alta velocidad.
- 2000 → Servicios de Voz sobre IP
  - ITU-T : H.323
  - IETF : SIP

#### **2.2.2 Escenarios.**

En cualquiera de las tecnologías de transporte de voz sobre redes IP vamos a encontrar distintos escenarios dependiendo de quién inicia o recibe la llamada, ya sea un ordenador (o teléfono IP) o un teléfono convencional. Así nos encontramos con 5 posibles escenarios:



a) *Escenario 1:*

Un teléfono convencional conectado a la RTC (Red Telefónica Conmutada) inicia la llamada hacia un ordenador con conexión a una red IP.

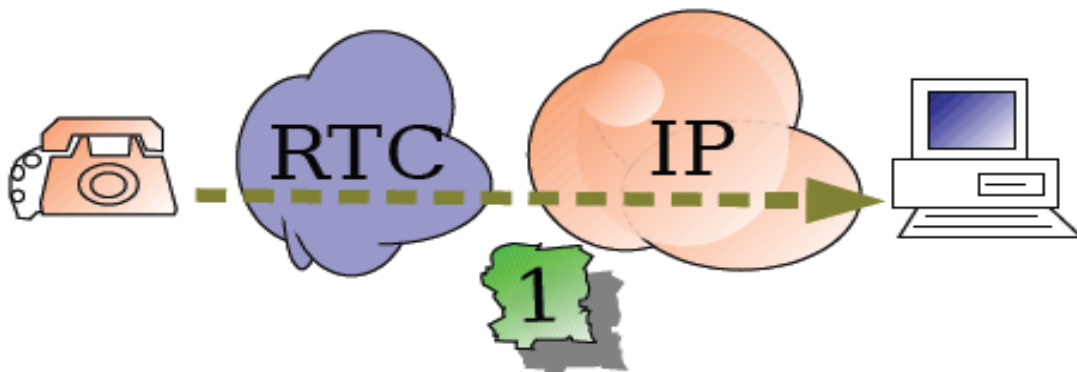


Figura 2.1: Escenario 1

b) *Escenario 2:*

Un ordenador con conexión a una red IP inicia la llamada a un teléfono conectado a la RTC.

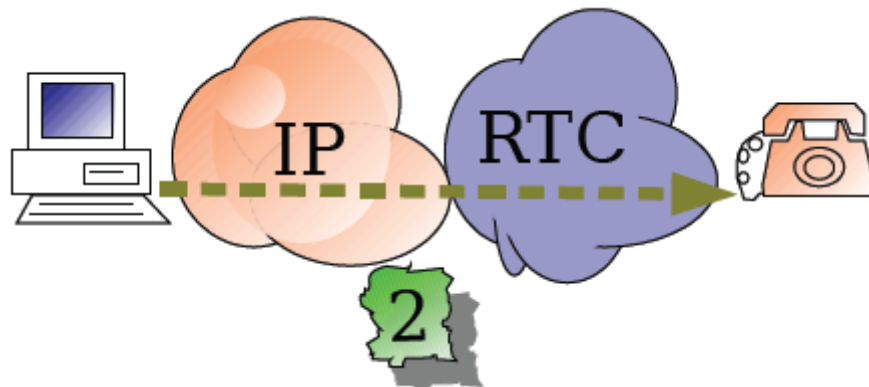


Figura 2.2: Escenario 2

c) *Escenario 3:*

Un teléfono convencional conectado a la RTC inicia la llamada hacia otro teléfono normal también conectado a la RTC a través de una red IP.

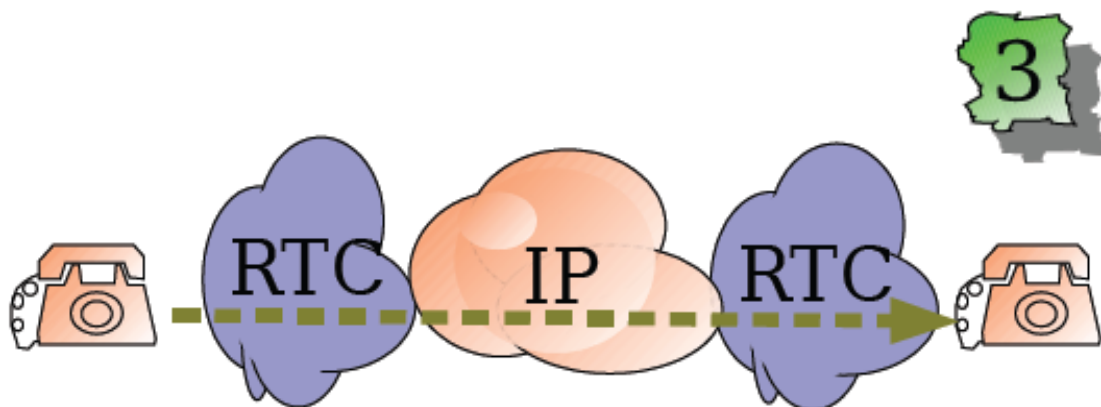


Figura 2.3: Escenario 3

d) *Escenario 4:*

Dos ordenadores conectados a una red IP mantienen una conversación sin necesidad de RTC ni teléfonos convencionales.

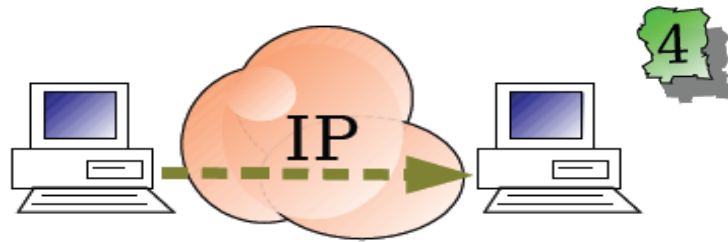


Figura 2.4: Escenario 4

e) *Escenario 5:*

Dos ordenadores conectados a una red IP inician y mantienen una llamada a través de la red RTC.

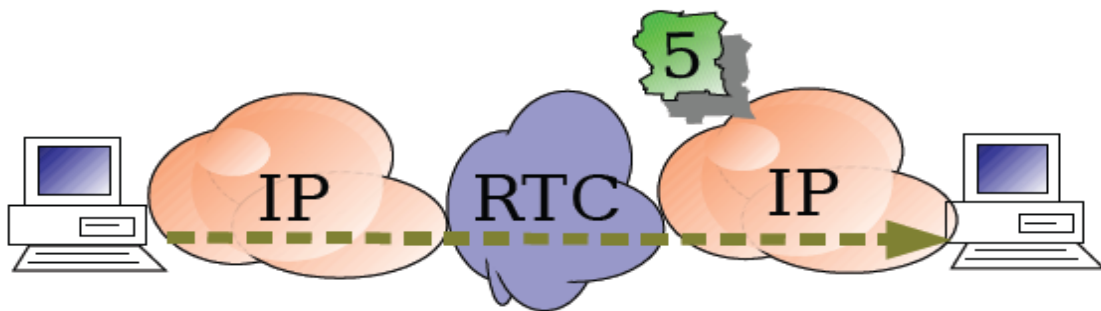


Figura 2.5: Escenario 5

### 2.2.3 *Ventajas e Inconvenientes.*

*Ventajas de VoIP:*

- Al contar con una infraestructura propia y única, se requiere una inversión inicial rápidamente amortizable, que puede ser aún menor si empleamos la multiplexación estadística sobre ATM o *Frame Relay* en líneas dedicadas alquiladas.
- Administración y gestión sencilla de la infraestructura.
- Menor coste de mantenimiento.
- Gracias a la utilización de las redes IP, es fácil añadir servicios nuevos, tiene una gran difusión por el número de usuarios conectados a *Internet* y al aprovechar la red de redes nos beneficiamos del transporte gratuito.

*Inconvenientes de la VoIP:*

En la calidad de la conversación influyen demasiados factores como son el terminal utilizado, la baja tasa de algunos *codecs* y algunos aspectos de las redes IP como el retardo y su variación o las pérdidas de paquetes.

Para solucionar estas desventajas podemos sobredimensionar el ancho de banda e implementar las redes de servicio garantizado QoS (*Quality of Service*).

### 2.2.4 *Transporte de la Voz en redes IP.*

Los efectos del transporte de voz sobre redes IP son la variación del retardo y la pérdida de secuencia en la entrega de paquetes. Para contrarrestarlos podemos usar *buffers* y números de secuencia.

La IETF ofrece dos protocolos de transporte para proporcionar números de secuencia,

marcas de tiempo e identificación del tipo de carga útil, que no influyen en la calidad de la red de transporte IP. Estos dos protocolos trabajan sobre UDP (*User Datagram Protocol*) y son RTP y RTCP.

a) *RTP (Real Time Protocol, RFC 1889)*.

El protocolo RTP se usa tanto en SIP como en H.323 y soporta la transferencia de tráfico de audio y voz en redes de conmutación de paquetes. Como protocolo de transporte permite al receptor detectar cualquier pérdida de paquetes y ofrece una marca de tiempo para que pueda compensar el retardo producido por la transmisión.

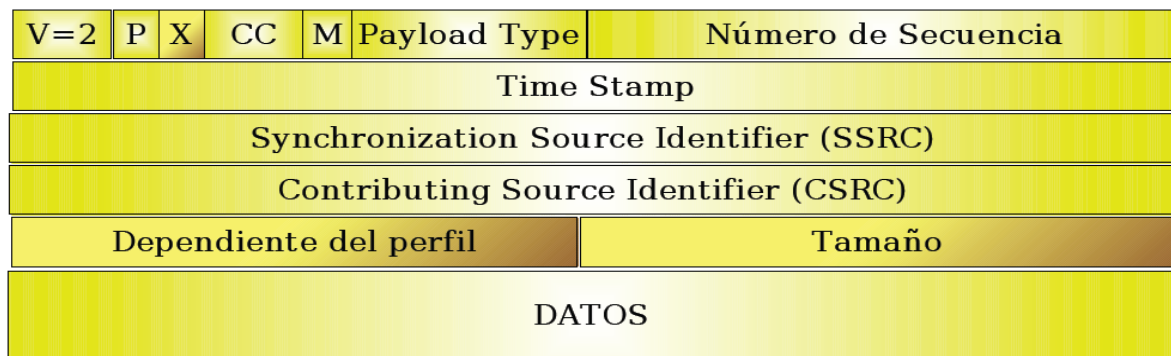
En la cabecera RTP encontramos información útil de cómo cada *codec* fragmenta los paquetes para reconstruir correctamente el flujo de datos (*stream*) enviado.

La seguridad no recae en RTP en forma de autenticación, pero podemos cifrar el contenido mediante algoritmos de bloque.

Las funciones de RTP incluyen:

- *Números de secuencia*: ayudan a la identificación de paquetes perdidos.
- *Identificador de sobrecarga*: informa del tipo de *codec* utilizado. En ocasiones es necesario cambiar el *codec* dinámicamente para ajustar el tráfico al ancho de banda disponible.
- *Indicador de Trama*: separa dos de las tramas en las que se divide el *stream* de audio o vídeo.
- *Identificador de origen*: provee una manera de distinguir diferentes orígenes en sesiones *multicast*.
- *Sincronización entre medios*: se utiliza para compensar los retardos de paquetes de diferentes *streams*. Con marcas de tiempo se ajustan los *buffers* de salida.

El formato de trama de RTP es el siguiente:



0

32

Figura 2.6: Formato de trama RTP

**P**=*Padding*, **X**=*Extensiones tras CSRC*, **CC**=*Cuenta CSRC*, **M**=*Marcador*

**Payload type**=*Tipo de sobrecarga*, **Número de secuencia**=*Comienza con un nº aleatorio*

**Timestamp**=*Cuenta de ticks desde la emisión del primer paquete ( 1 tick = 1/8000 )*

**SSRC**=*Origen del envío, a un mismo origen, mismo tiempo y nº de secuencia*

b) *RTCP (Real Time Control Protocol, RFC 1890)*.

El protocolo RTCP trabaja conjuntamente con el protocolo RTP en una sesión RTP para implementar nuevas funciones como:

- *QoS Feedback*: RTCP envía periódicamente información sobre la calidad de servicio.
- *Control de sesión*: con el uso del paquete BYE se anuncia a los demás participantes que se finaliza la sesión.
- *Identificación*: en los paquetes se incluyen el nombre, dirección de correo y teléfono de todos los participantes de la sesión.
- *Sincronización entre medios*: RTCP provee la información necesaria para que se reproduzcan a la vez los *streams* transmitidos por separado de audio y vídeo.

El tráfico de paquetes RTCP aumentan con el número de usuarios, incrementando el ancho de banda ocupado. Para prevenir esta masificación se acota el ancho de banda para este protocolo (lo normal es acotarlo a un 5%).

Los tipos de paquete RTCP son:

- *SR (Sender Report)*: Información sobre transmisión y recepción.
- *RR (Receiver Report)*: Información de recepción para los receptores.
- *SDES (Source Description)*: Parámetros del origen (CNAME).
- *BYE*: Enviado por un participante cuando abandona la conferencia.
- *APP*: Funciones propietarias específicas de aplicación.

Como ejemplo, vemos el formato de la trama de dos tipos típicos como el SR y el SDES:

a) *Sender Report*:

V=2	P	RC	PT=SR=200	Longitud
Sender (SSRC del autor del report)				
NTP Timestamp (byte más significativo)				
NTP Timestamp (byte menos significativo)				
RTP Timestamp				
Sender's Packet Count				
Sender's Octet Count				
SSRC 1 (SSRC de la primera fuente)				
Datos RR adicionales				
SSRC N				
Datos RR adicionales				

Figura 2.7: Tipo Sender Report

**RC:** *Report Count*, **PT:** Carga útil = 200 para SR, **Longitud del Report**,

**SSRC:** qué lo origina, **NTP timestamp:** Segundos desde el 1/1/1900

**RTP timestamp:** el mismo instante en *ticks* de RTP (equivalencia)

**Paquetes y octetos enviados desde el inicio de la sesión (por SSRC)**

**Conjunto de RR:** uno por cada fuente escuchada

b) *Paquetes SDES:*

Como el anterior pero con las siguientes diferencias:

**SC:** *Source count*

**PT:** Carga útil = 202 para SDES

Cada origen tiene un campo SSCR (o CSRC) y una lista con información:

- CNAME: identificador único del formato **usuario@host** dirección IP.
- NAME: nombre común del origen.
- *E-MAIL, PHONE, LOCATION,...*

### 2.3 Protocolo H.323.

Es un estándar de la ITU-T (*International Telecommunications Union*). Esta recomendación fue diseñada originalmente para el tráfico de contenido multimedia en redes LAN, pero se extendió para cubrir la voz sobre IP. Por este motivo no se tiene en cuenta la calidad de servicio (QoS). La estandarización de H.323 permite que productos de diferentes fabricantes cooperen sin problemas.

H.323 define la señalización necesaria para llamadas y conferencias con los *codecs* más utilizados y utiliza para el transporte los protocolos RTP/RTCP aunque no se utilizan todas las capacidades de RTCP como los paquetes SDES.

*La pila de protocolos y la historia de H.323:*

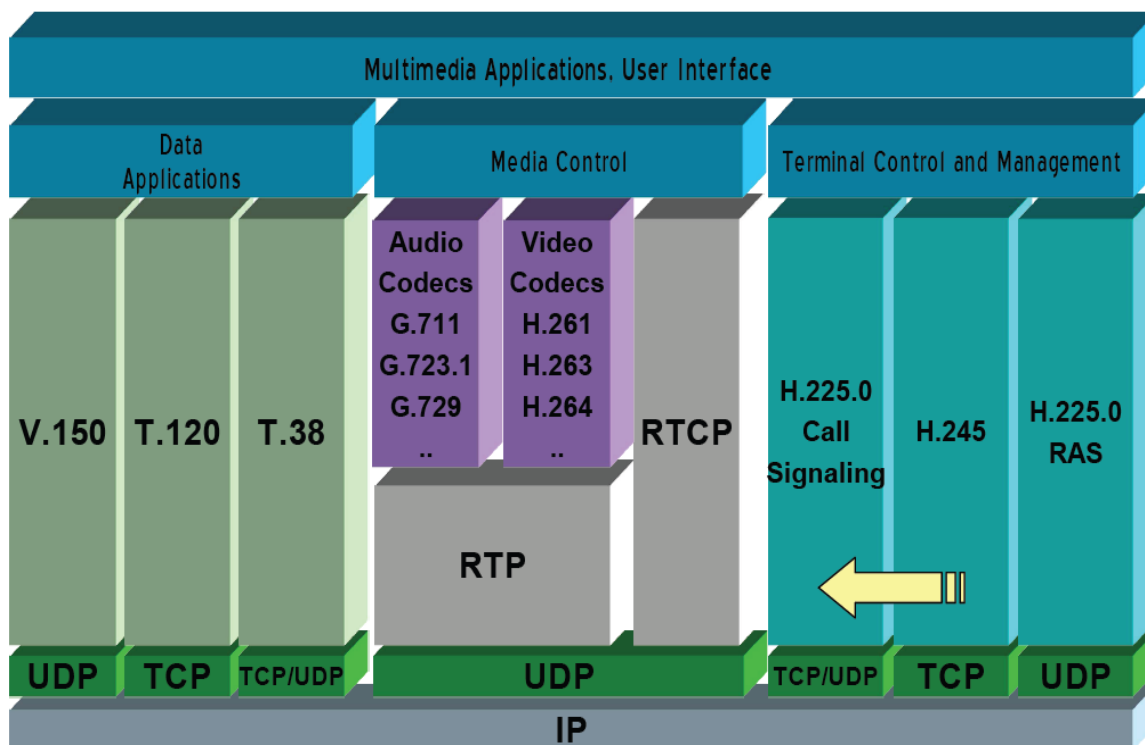


Figura 2.8: Pila de protocolos de H.323

- 1996 → H.323 versión 1.
- Enero 1998 → H.323 versión 2.  
Se le agrega seguridad con H.235, *FastConnect*, *Tunneling* con H.245 y servicios suplementarios con H.450.1.2.3.
- Septiembre 1999 → H.323 versión 3.  
Se añade control remoto con H.282 y los anexos G (comunicación entre dominios), E (multiplexado de señalización de llamada) y F (*Simple Endpoint Type*), así como 5 servicios suplementarios nuevos y MIBs de H.323.
- Noviembre 2000 → H.323 versión 4.  
Descomposición del *gateway* en dos (MGC y MC), multiplexación de transmisión de *streams*, control mediante HTTP (anexo K), una revisión de H.450 (la 8-11), principios de QoS, fax en tiempo real, pago a crédito y la URL H.323 (`h323:user@host`).
- Julio 2003 → -> H.323 versión 5.  
El canal de señalización de H.223 permanece abierto durante la llamada (cerrarlo implica el fin de la transmisión), se añade un segundo procedimiento de finalización de la llamada, se mejora la función de *time to live* en el *gatekeeper*, se aporta la secuencia de confirmación de admisión y se añade a H.323 el anexo R.

### 2.3.1 Los Componentes de la Arquitectura H.323.

En este punto vamos a describir los principales integrantes de una infraestructura H.323 típica como son los *endpoints* ( terminales, *gateways* y *Multipoint Control Units*) y los *gatekeepers*.

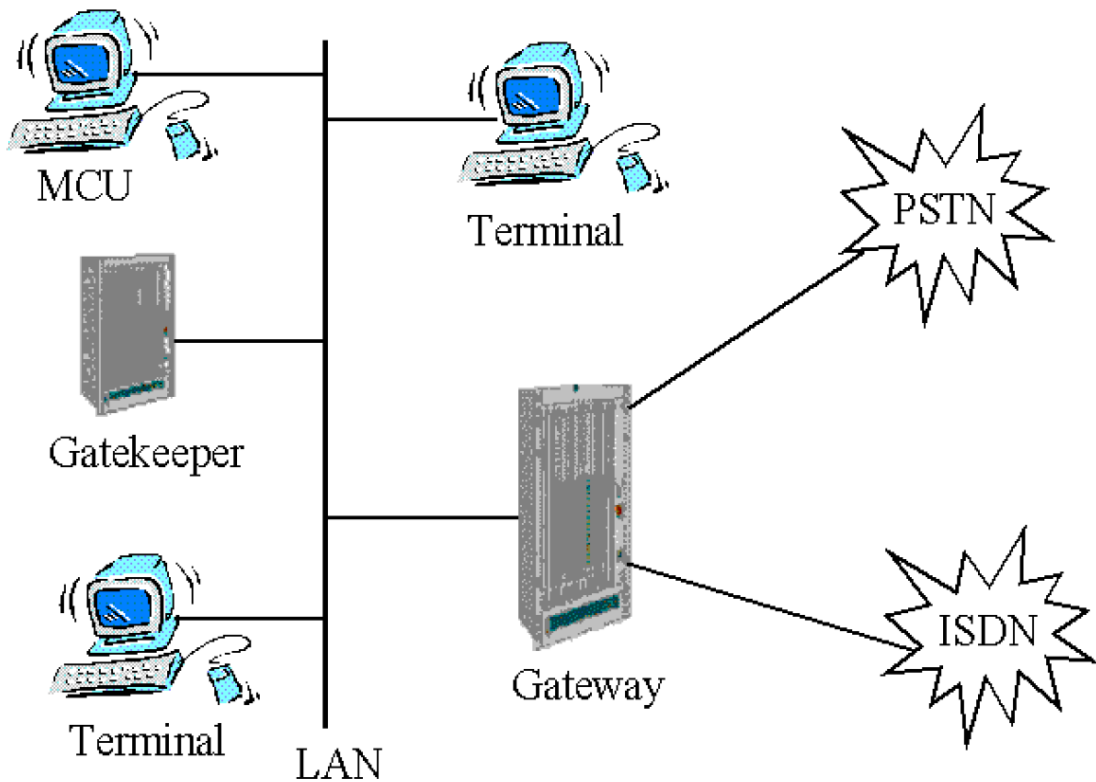


Figura 2.9: Componentes de la arquitectura H.323

#### a) *Terminales:*

Son puntos finales de los clientes de la arquitectura y se encargan de las comunicaciones de dos vías en tiempo real. Todos los terminales H.323 tienen que soportar H.245 (para controlar el acceso a la utilización de los canales), Q.931 (se requiere para la señalización e inicio de las llamadas), *Registration Admission Status* (RAS, interactúa con el *gatekeeper*) y RTP. Como opción, los terminales pueden ser compatibles también con los protocolos de conferencia de datos T.120, *codecs* de vídeo y con MCU.

Los terminales H.323 se comunican con otros terminales H.323, con un *gateway* H.323 o con un MCU.

Como ejemplo de terminales típicos podemos señalar teléfonos *hardware* H.323, videoteléfonos o *softphones*. Utilizaremos teléfonos convencionales conectados a una tarjeta de interfaz de voz con puertos FXS (*Foreign Exchange Station*) de un *router* Cisco 1761 como terminales H.323.

#### b) *Gateways:*

Los *gateways* H.323 se ocupan de la comunicación en tiempo real entre dos terminales de la red IP o entre otros terminales ITU de una red de conmutación de paquetes o con otro *gateway* H.323. Implementan la función de “traductores”, ya que favorecen la transición entre formatos de transmisión distintos y entre *codecs* de audio o vídeo según la capacidad del enlace. Son una parte muy importante de cualquier arquitectura de VoIP porque proveen la interfaz entre la Red Telefónica Básica y la red pública de *Internet*. Así mismo, para las llamadas entre terminales de una misma LAN no son necesarios. Los *gateways* H.323 se usan en entornos en los que se necesita salida a la RTB o para comunicar diferentes redes con otros *gateways* usando los protocolos H.245 y Q.931.

En el proyecto se utilizará un *router* Cisco 1761 como *gateway* H.323 gracias a la tarjeta de interfaz de voz con puertos FXO (*Foreign Exchange Office*).

#### c) *Gatekeepers:*

Los más importantes componentes de un sistema H.323 son los *gatekeepers* porque ejercen de administradores de la arquitectura. Cada *gatekeeper* es el punto central de todas las llamadas de su zona (suma de un *gatekeeper* y todos los terminales que tenga suscritos). Las funciones más comunes de los *gatekeepers* son:

- *Traducción de direcciones:* se traducen direcciones “alias” a direcciones de transporte usando una tabla de traducción actualizada mediante los mensajes de registro.
- *Control de admisión:* pueden ofrecer o denegar acceso basándose en políticas de autorización, en las direcciones origen y destino o cualquier otro criterio.
- *Señalización de la llamada:* el *gatekeeper* elige si compartir la señalización de la llamada con los *endpoints* o procesarla íntegra por su cuenta. También puede dejar conectado el canal de la señalización de la llamada directamente entre los terminales origen y destino.
- *Autorización de llamada:* usando la señalización H.225, el *gatekeeper* es capaz de rechazar la llamada si el terminal origen o destino poseen algún tipo de restricción por período de tiempo o por destino.
- *Administración del ancho de banda:* gracias al uso del protocolo H.225, es posible controlar el número de terminales H.323 que tienen acceso permitido a la red. El *gatekeeper* también puede denegar llamadas por falta de ancho de banda.
- *Administración de llamadas:* el *gatekeeper* mantiene una lista de las llamadas H.323 entrantes y salientes. Esto se usa para conocer el estado de cada terminal y recoger información para la función de administración del ancho de banda.

En el proyecto se desestima la inclusión de un *gatekeeper* por la sencillez de las pruebas realizadas en el laboratorio, pero existen dispositivos *hardware* de múltiples fabricantes o *software* como GNUGk, de licencia libre.

d) *Multipoint Control Unit (MCU)*:

La unidad de control multipunto se encarga de hacer posible la participación de tres o más terminales y *gateways* en una conferencia. Es un componente opcional que consiste en un *Multipoint Controller (MC)* y *Multipoint Processors (MP)* opcionales. El MC o controlador multipunto determina las capacidades comunes de los terminales por la señalización del protocolo H.245 y controla a los MP o procesadores multipunto, que se encargan de la multiplexación y conmutación de los datos, vídeo y audio.

2.3.2 *Control y Señalización de H.323.*

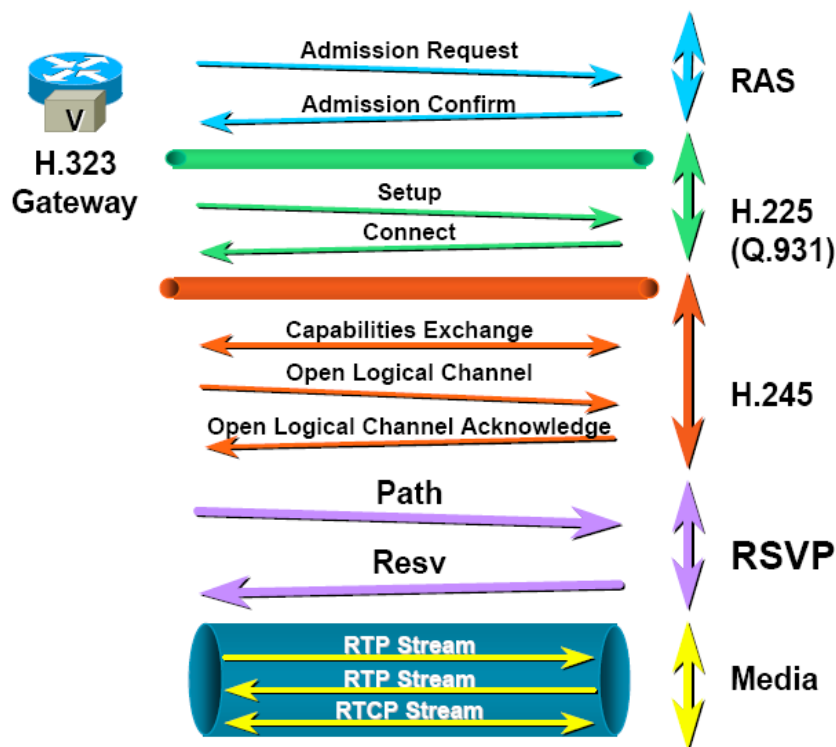


Figura 2.10: Señalización de H.323

En un sistema H.323 se emplean los siguientes protocolos de control:

- Señalización y control de llamada con H.225.0/Q.931.
- Control multimedia con H.225.0 RAS (*Registration , Admission and Signaling*) y H.245.

Para iniciar una llamada se usa el canal de H.225 RAS, después del establecimiento se utiliza H.245 para negociar los *streams* multimedia.

a) *H.225.0:RAS* :

El canal de RAS se usa para la comunicación entre los *endpoints* y el *gatekeeper*. Los mensajes RAS se envían sobre el protocolo de transporte UDP, por lo que se recomiendan los *timeouts* y el reenvío de paquetes.



Los procedimientos definidos por el canal RAS son:

- *Descubrimiento del gatekeeper:*

Con este proceso un *endpoint* encuentra el *gatekeeper* en el cual debería suscribirse. Se envía un mensaje *multicast* (a destinos seleccionados entre varias opciones) llamado *Gatekeeper Request* (GRQ) que pregunta por el *gatekeeper*. Uno o más *gatekeepers* responderán con la confirmación del *gatekeeper* (*Gatekeeper Confirm, GCF*) que indica su disposición a ser el *gatekeeper* del *endpoint* y la dirección de transporte de su canal RAS. Cuando se reciben los GCF, el *endpoint* elige su *gatekeeper* y se registra con él. Si no se recibe ninguno, se retransmite el GRQ tras un intervalo de tiempo determinado.

Un *gatekeeper* puede estar saturado o programado para no aceptar nuevos usuarios, así que enviará un mensaje GRJ (*Gatekeeper Reject*) rechazando la posibilidad de suscripción.

- *Registro del endpoint:*

Con este proceso, un *endpoint* se une a una zona suscribiéndose a un *gatekeeper* descubierto en el proceso anterior informándole de su dirección de transporte y su “alias”. El registro se realiza enviando un mensaje *Registration Request* (RRQ, petición de registro) al *gatekeeper* a través de su dirección de transporte del canal RAS. Seguidamente, el *endpoint* recibirá un mensaje aceptando su registro (*Registration Confirmation, RCF*) o denegándolo (*Registration Reject, RRJ*). Con el registro, el *gatekeeper* asigna un “alias” sólo a una dirección de transporte.

Si un *endpoint* quiere cancelar el registro envía al *gatekeeper* un mensaje *Unregister Request* (URQ) al que el *gatekeeper* responderá con un mensaje de confirmación *Unregister Confirmation* (UCF). Por el contrario, si es el *gatekeeper* el que cancela la suscripción del *endpoint*, éste recibe el mensaje URQ y confirmará su recepción y aceptación de la cancelación del registro al *gatekeeper* con un mensaje UCF.

- *Localización del endpoint:*

Para conseguir la información necesaria para establecer contacto con un *endpoint*, se envía un mensaje *Location Request* (LRQ, petición de localización) con el “alias” del *endpoint* buscado y el *gatekeeper* donde está registrado dicho *endpoint* responderá al que envió el LRQ (ya sea otro *gatekeeper* o un *endpoint*) con un mensaje *Location Confirmation* (LCF, confirmación de localización) con la información necesaria.

Todos los *gatekeepers* que reciban el LRQ por su canal RAS y no tengan registrado el “alias” buscado contestarán con un mensaje *Location Reject* (LRJ, renuncia de localización).

- *Admisiones, cambio de ancho de banda y estado:*

El canal RAS también se usa para la transmisión de mensajes de admisión, cambio de ancho de banda y estado. Estos mensajes se intercambian entre un *endpoint* y un *gatekeeper* y se utilizan para el control de admisiones y administración del ancho de banda.

Antes de comenzar una llamada, el mensaje *Admissions Request* (ARQ) especifica el ancho de banda requerido, que el *gatekeeper* puede reducir si envía el mensaje *Admissions Confirm* (ACF) con su medida de ancho de banda. Durante la llamada, para cambiar el ancho de banda se usa el mensaje *Bandwidth Change Request* (BRD) y puede ser enviado tanto por el *endpoint* como por el *gatekeeper*.

### b) Señalización de llamada con H.225.0

El canal de señalización de llamada se usa para transportar los mensajes de control H.225 sobre TCP, pues proporciona fiabilidad. En redes que no implementan un *gatekeeper*, los mensajes de señalización se pasan directamente entre los dos extremos de la llamada, asumiendo unas conocidas direcciones de transporte de señalización. Por contra, en las redes con *gatekeeper*, el intercambio de mensajes inicial es entre el *endpoint* que llama y el *gatekeeper* donde está suscrito el *endpoint* destino de la llamada.

La ausencia o presencia del *gatekeeper* condiciona no sólo la señalización, sino también el control posterior. Sin *gatekeeper* se establece un canal de control con el protocolo H.245 entre los dos *endpoints*, mientras que si éste está presente, dicho canal viaja a través del *gatekeeper*.c) *Control multimedia y de conferencia con H.245:*

El protocolo H.245 se utiliza para establecer y mantener los canales de transferencia multimedia transportados sobre RTP/RTCP. Las funcionalidades que aporta H.245 son:

- *Elección de maestro y esclavo:* se selecciona un MCU como maestro para la administración de conferencias.
- *Cambio de parámetros:* cuando la llamada está iniciada, H.245 se encarga del cambio y de las negociaciones.
- *Control de canales multimedia:* tras negociar los parámetros adecuadamente se abren o cierran nuevos canales lógicos de transferencia multimedia.
- *Control de conferencias:* en una conferencia, H.245 desarrolla un modelo de envío de datos entre los *endpoints*.

### 2.3.3 Llamadas y Envío de Mensajes H.323.

En este apartado veremos un ejemplo sencillo de llamada entre dos *endpoints* (dos ordenadores con un *softphone* (como *NetMeeting* o *GnomeMeeting*) instalado en una red con *gatekeeper*:

- *Primera fase: Inicio de llamada.*

Se emplea un subconjunto de Q.921 definido en H.225.0 y TCP 1720 como canal de señalización de llamada.

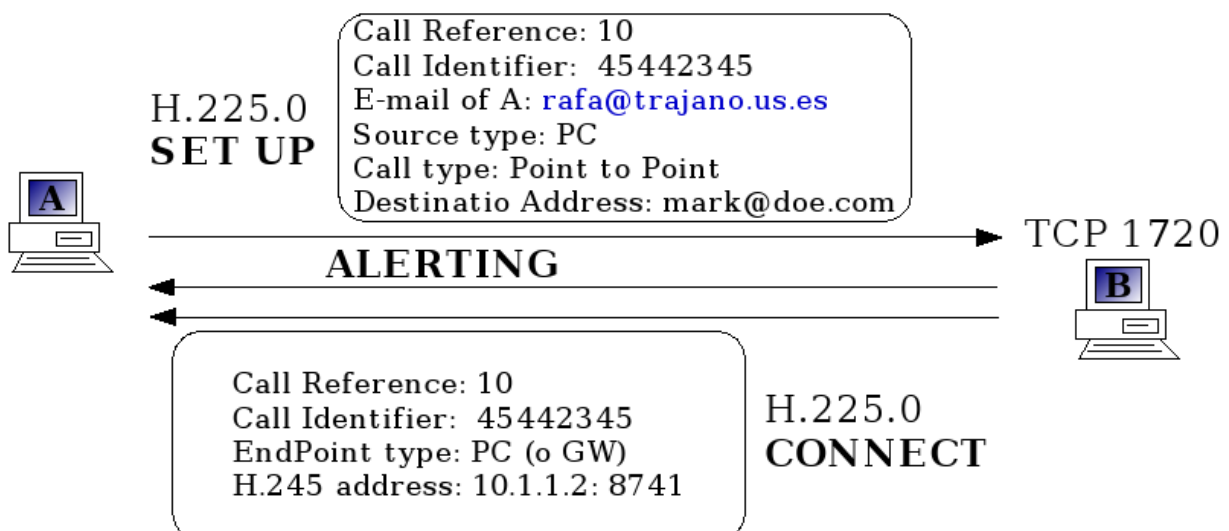


Figura 2.11: Inicio de llamada H.323

- *Segunda fase: Establecimiento del canal de control.*

El canal de control está definido en H.245 con el que se negocian los parámetros de la conexión y se eligen maestros y esclavos.

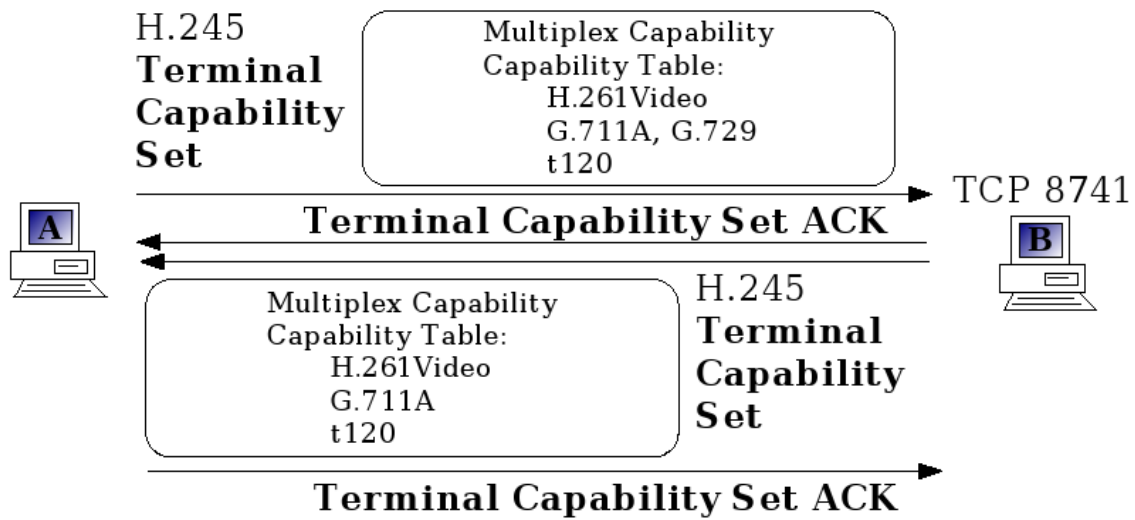


Figura 2.12: Establecimiento del canal de control de H.323

- *Tercera fase: Comienzo de la llamada.*

En esta fase se abren los canales lógicos de transferencia multimedia y se sigue negociando los maestros y esclavos.

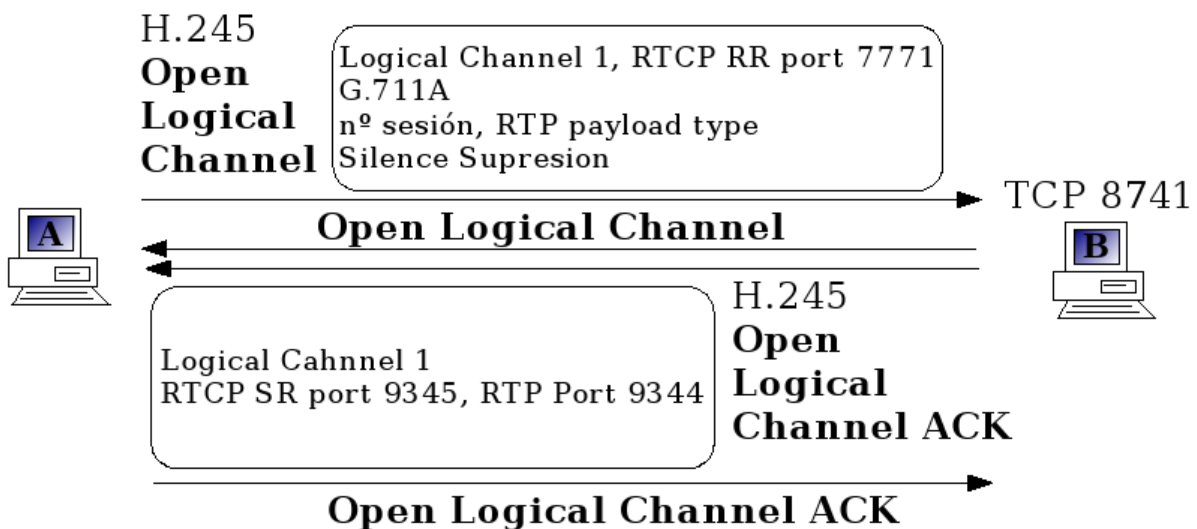


Figura 2.13: Comienzo de la llamada H.323

- *Cuarta fase: Diálogo.*

Ya tenemos varios canales abiertos sincronizados y podemos observar el estado de la llamada con los mensajes SR y RR.

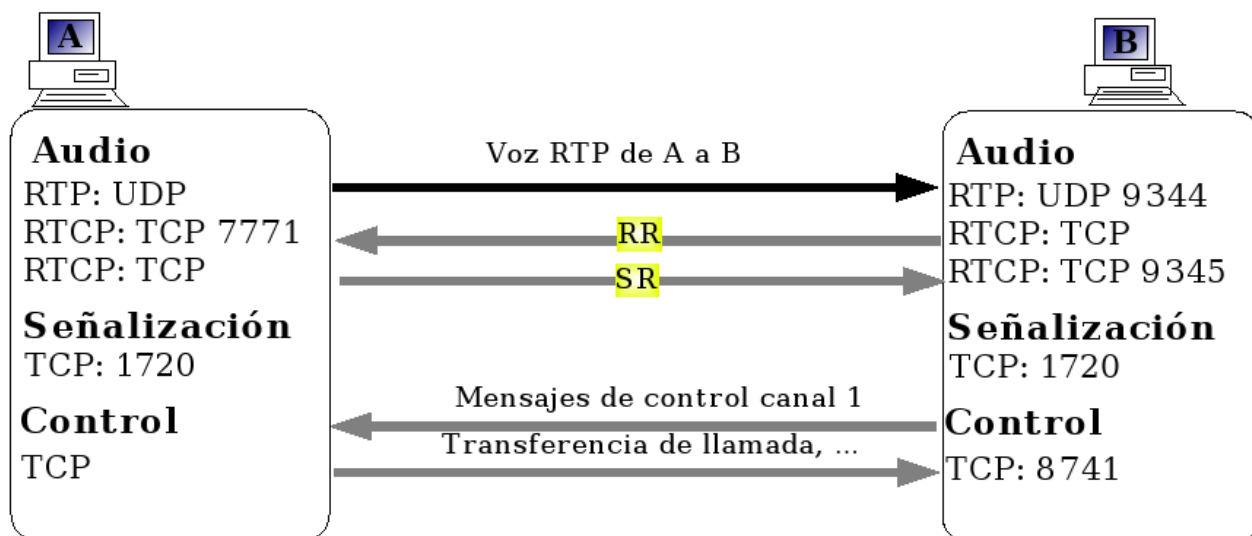


Figura 2.14: Diálogo H.323

- *Quinta fase: Finalización de la llamada.*

La secuencia lógica para terminar una llamada es:

- El terminal que quiere acabar la comunicación envía el mensaje *Close Logical Channel* por cada canal abierto.
  - Espera la recepción de los ACKs por cada canal.
  - Tras recibir todos los ACKs envía el mensaje *End Session Command* para el protocolo H.245.
  - Espera recibir por el canal H.245 un mensaje exacto al que envió para cerrar el canal.
  - Finalmente se envía un mensaje *Release Complete* al protocolo H.225.0.

El problema de la finalización de las llamadas en H.323 es que no todos los terminales cumplen con esta secuencia lógica cerrando directamente los canales de comunicación, lo que se traduce en pérdida de convergencia en la red al tener uno (o varios) equipos esperando el restablecimiento de los canales.

## 2.4 Protocolo SIP.

El *Session Initiation Protocol* (protocolo de inicio de sesión) es el estándar del IETF para la comunicación VoIP, que fue definido en marzo de 1999 por el grupo MMUSIC en la RFC 2543. Posee una arquitectura cliente/servidor parecida al conocido HTTP y es un protocolo de control de la capa de aplicación del modelo OSI que se usa para crear, mantener y finalizar sesiones entre dos o más usuarios. Además provee fiabilidad, no teniendo que recurrir al protocolo TCP y depende del protocolo SDP (*Session Description Protocol*, protocolo descriptor de sesión) que negocia entre los conferenciantes el grupo de *codecs* que se van a utilizar en la llamada. SIP soporta bien el redireccionamiento gracias a los *proxies*, que ubican y destinan el tráfico hacia los terminales.

### 2.4.1 La Pila de Protocolos de SIP.

SIP trabaja con los protocolos RSVP (*Resource Reservation Protocol*, protocolo de reserva de recursos), RTP/RTCP, RTSP (*Real Time Streaming Protocol*, protocolo de envío multimedia en tiempo real), SAP (*Session Announcement Protocol*, protocolo de anuncio de sesión) y SDP (*Session Description Protocol*, protocolo descriptor de sesiones) como se observa en el siguiente diagrama:

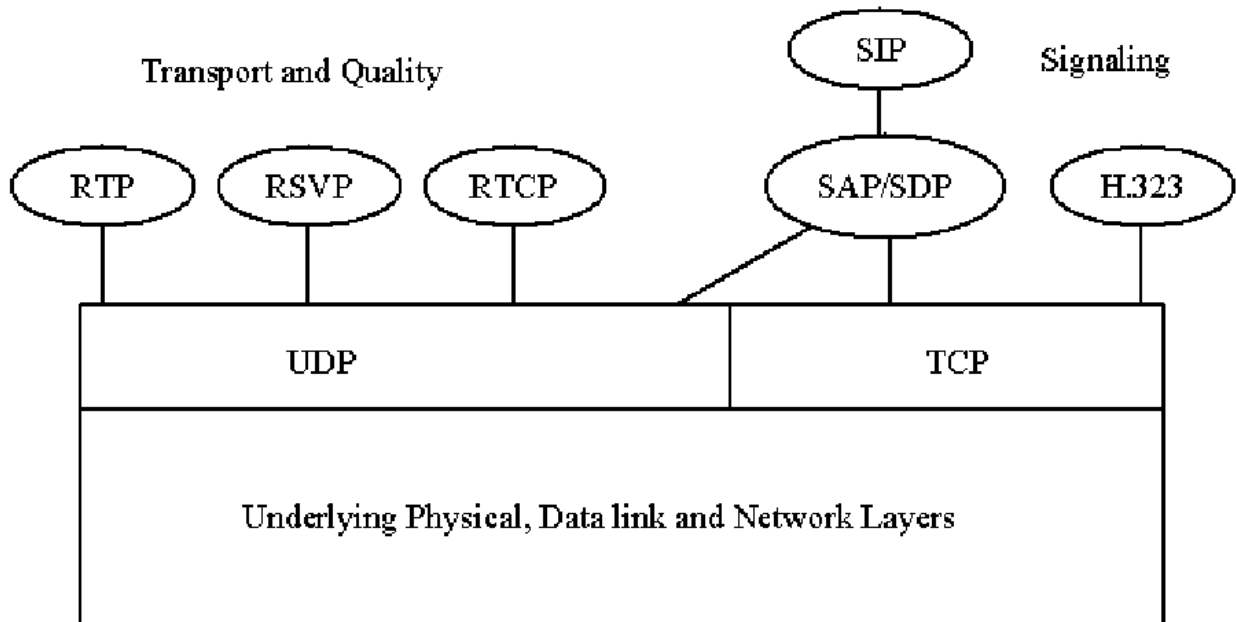


Figura 2.15: La pila de protocolos SIP

### 2.4.2 Elementos de una Arquitectura SIP.

Los elementos principales de un sistema SIP son:

- *Agentes de usuario*: realizan las funciones de manejo de las llamadas entrantes o salientes. También filtrado de llamadas, localización del usuario y reintento de llamadas fallidas. En el proyecto usaremos *Linphone* agente de usuario *software* y las extensiones de la tarjeta FXO del *router* Cisco 1761 como agentes de usuario *hardware*.
- *Servidores*: incluyen varias funciones, como son:
  - *Proxy*: actúa como servidor en un lado y como cliente en el otro y transmite los paquetes con o sin cambios.
  - *Registrar*: permite el registro y cancelación de agentes de usuario.
  - *Redirigir*: devuelve la dirección del siguiente servidor (siguiente salto).

En el laboratorio usaremos SER (*Sip Express Router*).

Para comunicar los elementos de una arquitectura SIP se usan búsquedas DNS (*Domain Name System*) o direcciones IP configuradas y específicamente direcciones *multicast* para el servicio de registro del servidor.

### 2.4.3 Mensajes SIP

Los mensajes SIP se codifican con HTTP v1.1 y caracteres ISO 10646 y ocupan mayor ancho de banda que los mensajes H.323. Hay dos tipos de mensajes: peticiones y respuestas.

#### a) Peticiones:

- *ACK*: enviado por un cliente para confirmar que ha recibido la respuesta.
- *BYE* : enviado para finalizar una llamada.
- *Cancel* : aborta una solicitud enviada previamente.
- *Invite*: se utiliza para iniciar una llamada.
- *Options*: permite a un cliente conocer los métodos de un servidor.
- *Register* : para que los agentes de usuario puedan registrar su localización actual.

#### b) Códigos de respuesta:

- *1xx*: Información (180 *Ringin*g)
- *2xx*: Éxito (200 OK)
- *3xx*: Redirección (301 *Moved permanently*)
- *4xx*: Error del Cliente (400 *Bad Request*, 406 *Not Acceptable*)
- *5xx*: Error del Servidor (502 *Bad Gateway*)
- *6xx*: Fallo General (600 *Busy Everywhere*)

### 2.4.4 Funciones de SIP.

Las funciones de SIP son.

#### a) Resolución de direcciones:

Es una de las principales funciones de SIP y la efectúan tanto los agentes de usuario como los servidores aunque suele ser tarea del *proxy*. La resolución implica la mayoría de las ocasiones una traducción de una dirección en formato URI (*Universal Resource Identifier*, identificador universal de recurso) a una dirección IP. Normalmente se manda un mensaje DNS SRV, seguido de un ENUM *Lookup* (búsqueda ENUM) y de un *Location Server Lookup* (búsqueda de localización de servidor).

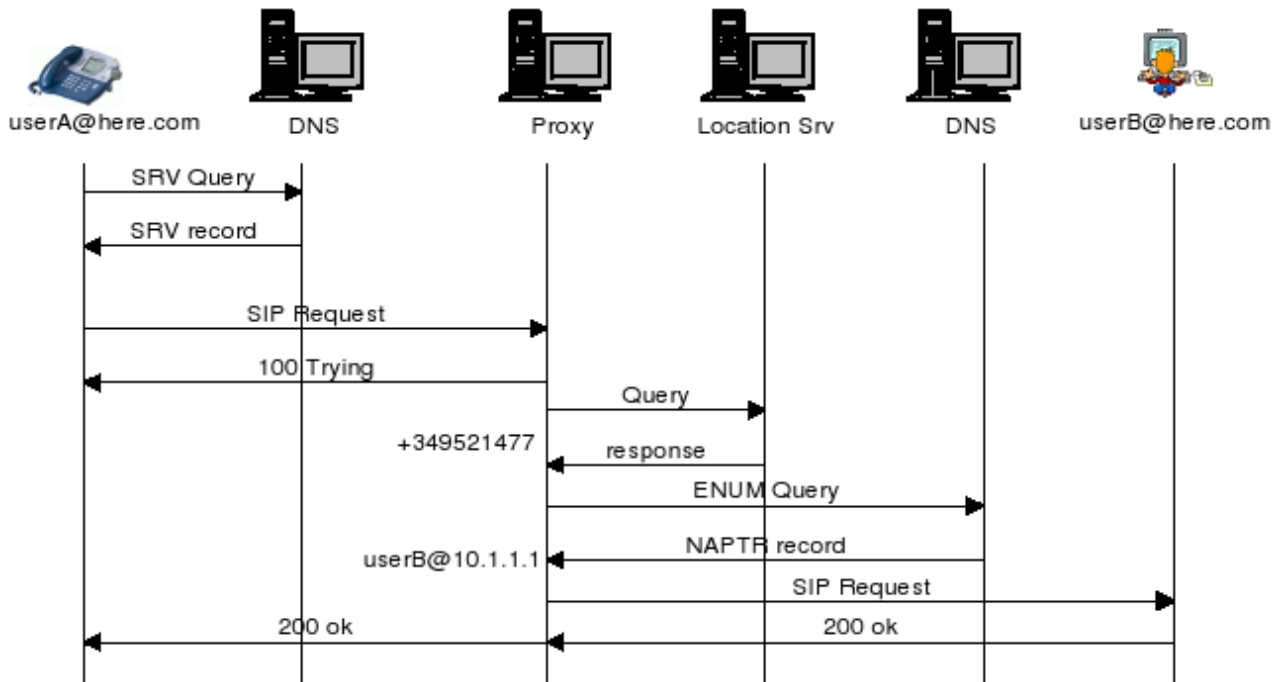


Figura 2.16: Mensajes para la resolución de direcciones con SIP

b) *Establecimiento de la sesión:*

Se realiza a través del *proxy* cuando está presente, si no, entre agentes de usuario directamente.

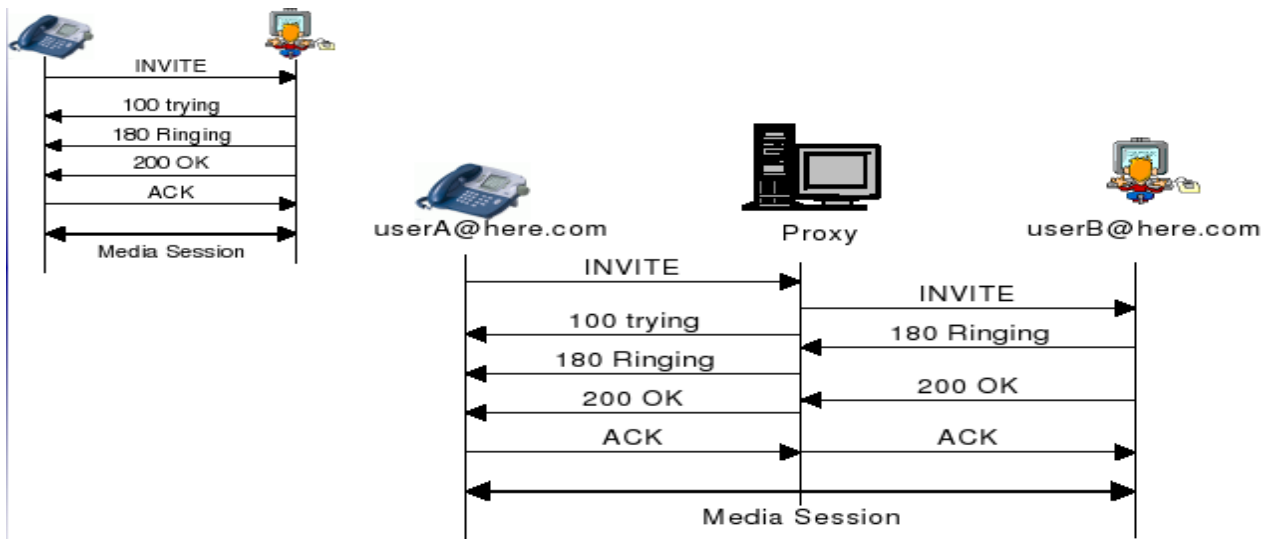


Figura 2.17: Establecimiento de la sesión con y sin gatekeeper

c) *Negociación de contenidos:*

La negociación se realiza con parte de las peticiones *Invite* y *ACK* con el protocolo SDP. Este protocolo también utiliza un sistema cliente/servidor, está diseñado para la arquitectura multimedia de *Internet* y está descrito en la RFC 2327. Con esta negociación se acuerda entre las dos partes los *codecs* que se van a usar, el tipo de contenido transportado, etc.

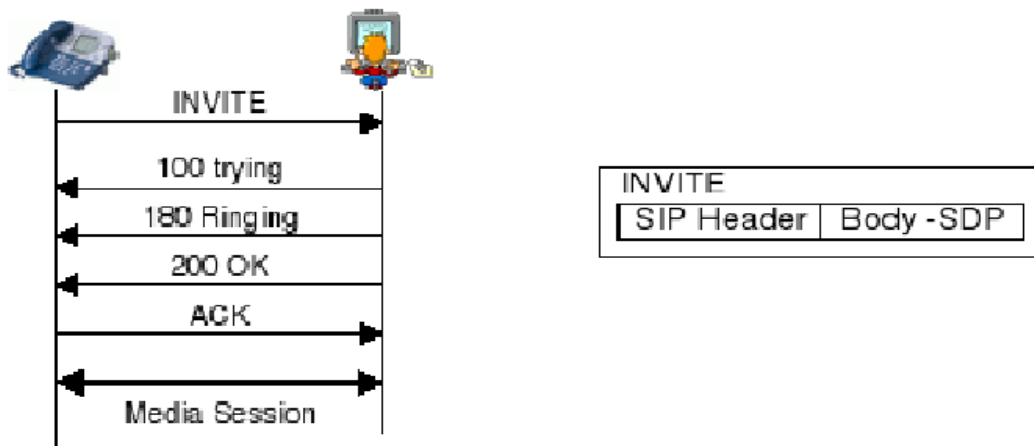


Figura 2.18: Negociación de contenidos



d) *Modificación de la sesión:*

Sólo se puede renegociar la sesión después de un primer establecimiento, no siendo indispensable cortar la transmisión en curso y con posibilidad de cambiar tanto el tipo de sesión como el *codec* usado o la dirección IP y el puerto.

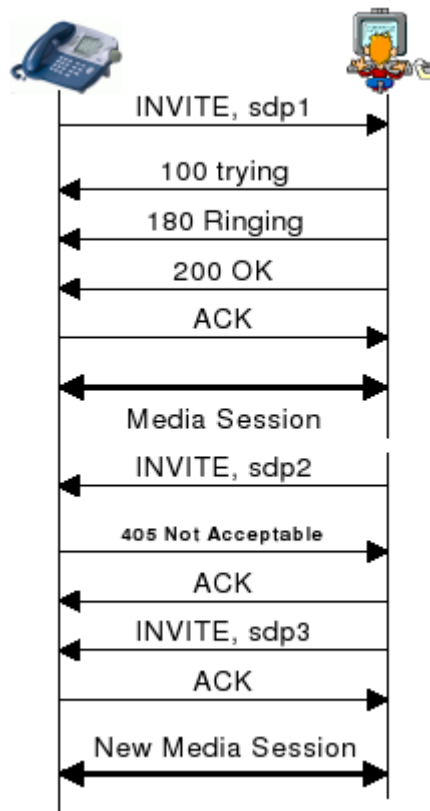


Figura 2.19: Cambio de parámetros de sesión

e) *Terminación y cancelación de la sesión:*

Hay una pequeña diferencia entre la terminación y la cancelación que se basa en la petición utilizada, es decir, si se termina la sesión es que ésta ha llegado a establecerse y si se cancela es que la sesión todavía está en la etapa de establecimiento como se aprecia en los siguientes diagramas:

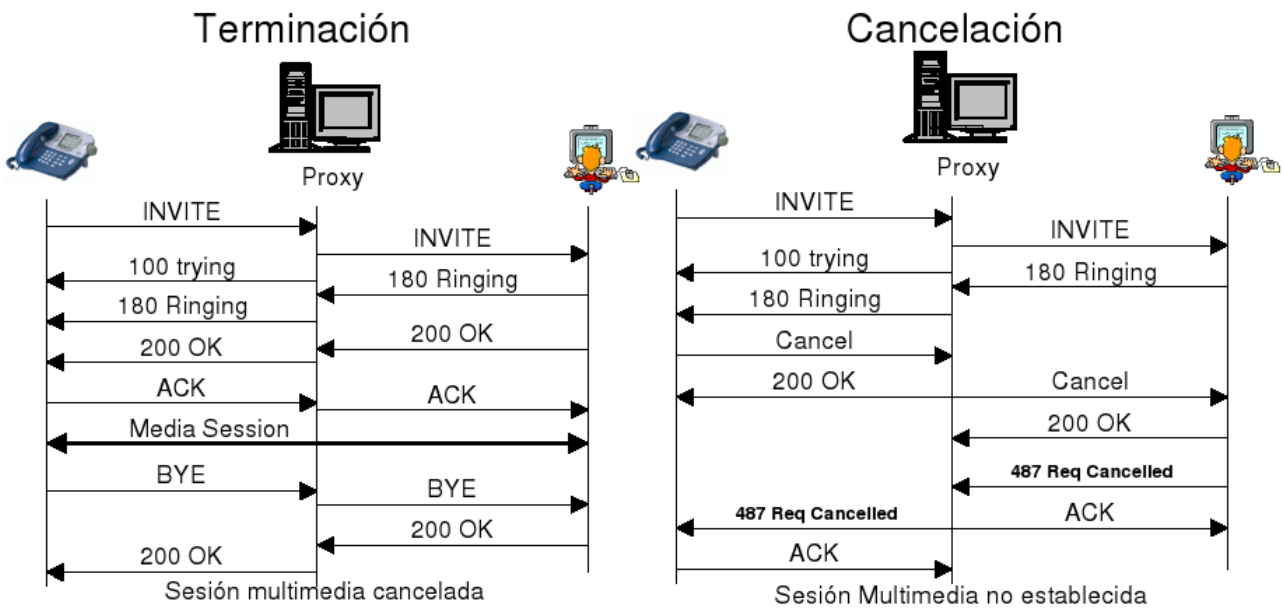


Figura 2.20: Terminación y cancelación de sesión

f) *Control de llamada con REFER:*

Gracias a esta funcionalidad de SIP podemos estar manteniendo una conversación con otro agente de usuario y antes de desconectarnos dejarlo llamando a un tercero con el mensaje REFER.



Figura 2.21: Control de llamada

g) *Sesiones con QoS:*

Es una aproximación a la implementación de la calidad de servicio como funcionalidad aprovechando que SDP transporta información QoS y usando tres extensiones de SIP como son:

- 183 *Session Progress* con SDP
- Confiabilidad a 183
- *PreCondition* MET COMET

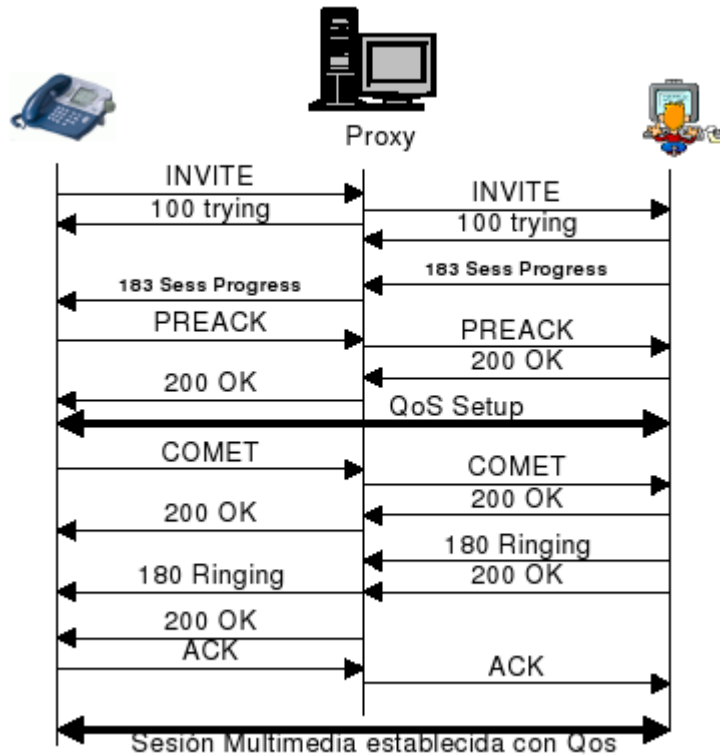


Figura 2.22: QoS con SIP

h) *Creación de servicios con SIP:*

Una de las mayores ventajas de SIP, ya que se pueden implementar de una manera flexible y sencilla servicios más avanzados que la PSTN debido a la ingente cantidad de información que se transporta en el establecimiento de la llamada usando las redes IP. Además, estas nuevas funcionalidades pueden residir en todos los componentes de la arquitectura o sólo en algunos, garantizando la compatibilidad.

Las herramientas que hacen posible esta extensión de servicios son SIP CGI (RFC 3050), SIP *Servlets* o el lenguaje CPL (*Call Processing Language*, lenguaje de procesamiento de llamadas).

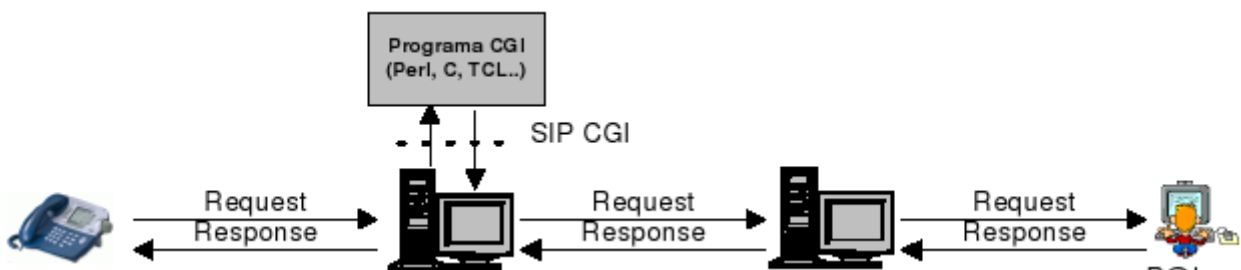


Figura 2.23: Nuevos servicios SIP

### 2.4.5 Ejemplo de Operación SIP.

Vamos a describir un ejemplo sencillo de llamada con SIP en el que un agente de usuario invita a otro a establecer una conferencia.

El agente de usuario que inicia la llamada compone un mensaje INVITE con la dirección en formato URI del agente de usuario destino ([usuario@dominio](#)) y normalmente lo envía al servidor que hace de *proxy*, el cual intenta averiguar la IP a la que pertenece el dominio del URI. Para esto, el *proxy* consulta al servidor de localización que le da la dirección del siguiente salto y manda el mensaje INVITE a dicha dirección. Cuando se alcanza el agente de usuario destino tras sucesivos saltos, éste envía la respuesta aceptando la llamada. Cuando la respuesta afirmativa llega al *proxy*, se la envía a su usuario registrado que inició la llamada y queda a la espera de la confirmación mediante ACK. Tras esto, la comunicación se establece directamente entre los dos usuarios.

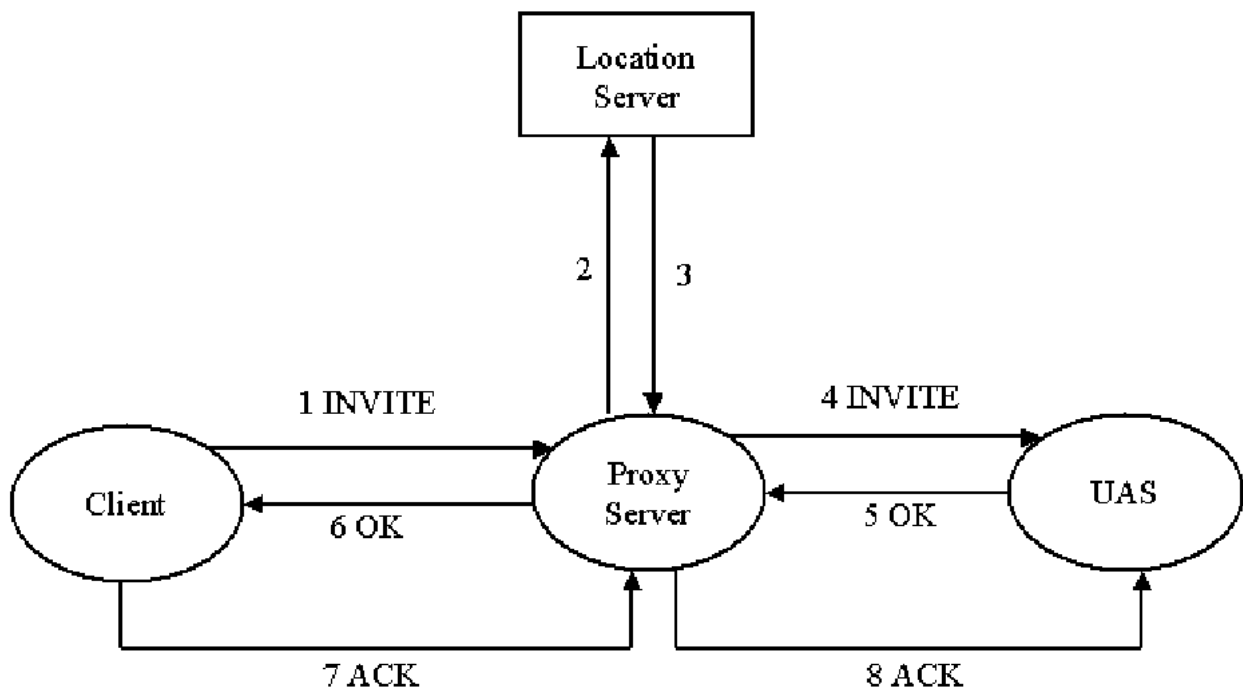


Figura 2.24: Ejemplo de operación SIP

## 2.5 SIP frente a H.323.

En la comparativa entre los dos sistemas, SIP despunta por varios motivos:

- SIP adaptó su señalización para las redes IP y H.323 para ATM y RDSI.
- H.323 es más complejo que SIP, con demasiada sobrecarga y sin medios sencillos para desarrollar nuevas funcionalidades.
- SIP usa los códigos de error y mecanismos de autenticación que HTTP al escribirse por y para *Internet*.
- La especificación de H.323 incluye cientos de elementos mientras que SIP sólo necesita 37 cabeceras distintas.
- SIP codifica los mensajes en texto plano como HTTP y H.323 lo hace en formato binario basado en ASN.1.
- H.323 no es tan escalable como SIP, como se demuestra en búsquedas complejas entre varios dominios.
- SIP es modular y está respaldado por la IETF, pero H.323 tiene demasiadas especificaciones de distintos vendedores.

- SIP usa URI, dota de prioridad a las llamadas e implementa *multicast*, mientras que H.323 sólo abre canales lógicos cuando son necesarios y tiene muy desarrollado el descubrimiento del *gatekeeper*.

## 2.6 Codecs.

La elección del *codec* es una de las elecciones más importantes en la transmisión de voz sobre redes IP, porque afecta directamente al consumo de ancho de banda y a la calidad de la conversación.

En esta tabla se han resumido los dos *codecs* más típicos en la solución VoIP de Cisco, el G.729 y el G.711, comparando el ancho de banda con el protocolo RTP normal con el consumo de la versión propietaria de Cisco (cRTP) y la latencia.

<b>Codec</b>	<b>Ancho de Banda Consumido</b>	<b>Ancho de Banda Consumido con cRTP (2 bytes de cabecera)</b>	<b>Latencia</b>
G.729 con un muestreo de 10 ms por trama	40 kbps	9.6 kbps	10 ms
G.729 con cuatro muestras de 10 ms por trama	16 kbps	8.4 kbps	40 ms
G.729 con dos muestras de 10 ms por trama	24 kbps	11.2 kbps	20 ms
G.711 con una muestra de 10 ms por trama	112 kbps	81.6 kbps	10 ms
G.711 con dos muestras de 10 ms por trama	96 kbps	80.8 kbps	20 ms

*Tabla 1: Comparativa de los codecs más comunes*

### 3 Descripción de las Herramientas.

Este capítulo contiene una breve reseña de las aplicaciones *software* y equipos *hardware* que vamos a utilizar en el proyecto. El puesto del laboratorio está provisto de un ordenador personal, un *router* Cisco 1761, un *Gateway* GSM BossLAN y dos teléfonos sencillos que actúan como extensiones.

#### 3.1 GNU/Linux Kubuntu 5.10 Brezzy Badger.

Utilizaremos un PC (*Personal Computer*) Compaq con procesador Intel Pentium III a 733 Mhz con 128 MB de RAM sobre el que se ha instalado la distribución de GNU/Linux Kubuntu 5.10, basada en Debian con el kernel 2.6.12-9-386 como sistema operativo. Este PC lo utilizaremos para configurar el *router* con el programa C-Kermit, como teléfono *software* usando Linphone y como *proxy* SIP con SER.

*¿Qué quiere decir kernel 2.6.12-9-386?*

Se refiere a la versión del núcleo de Linux que utiliza la distribución, un núcleo de la serie 2.6, versión 12, revisión 9 para la arquitectura de procesador i386. Los núcleos con número de versión par (2.6, 2.4,...) se consideran estables y los de versión impar (2.5, 2.3,...) inestables.

El primer kernel o núcleo de Linux (v.1.0) fue publicado en un canal de noticias en 1991 por Linus Torvalds, un estudiante de informática de Helsinki (Finlandia), que desarrolló un clon de Unix para su PC de casa basándose en Minix, un pequeño sistema operativo libre dedicado a la enseñanza.

La progresión de Linux se inició al unirse programadores de todo el mundo a través de Internet que ayudan a Linus a programar el núcleo y otras herramientas como el entorno de ventanas o programas de FTP, llegando al estado actual en el que es una clara alternativa a otros sistemas operativos como Microsoft Windows, MacOS, ZetaOS ...

Una de las bazas con las que Linux ofrece una clara ventaja desde el punto de vista del usuario, aparte de la seguridad o la estabilidad, es en el apartado de las licencias. En el año 1985 la fundación Free Software sentó las bases legales para la licencia GPL (*General Public License*), lo que inició el movimiento del *Software* Libre. Al liberar Linus Torvalds el núcleo de Linux bajo GPL, hizo posible lo siguiente:

- "Libertad 0", ejecutar el programa con cualquier propósito (privado, educativo, público, comercial, etc.).
- "Libertad 1", estudiar y modificar el programa (para lo cuál es necesario poder acceder al código fuente).
- "Libertad 2", copiar el programa de manera que se pueda ayudar a cualquiera.
- "Libertad 3", mejorar el programa, y hacer públicas las mejoras, de forma que se beneficie toda la comunidad.

De esta primera licencia libre surgieron otras también utilizadas en aplicaciones GNU/Linux como Open Source, LGPL, Creative Commons, etc que son más restrictivas pero conservan la actitud abierta.

*¿Qué significa basada en Debian?*

Debian es una de las más populares distribuciones de GNU/Linux, al incluir "de serie" estrictamente *software* bajo licencia GPL, por su vasto y eficaz equipo de desarrolladores y por su característica gestión de *software* mediante la herramienta apt-get. De ahí que otras comunidades adopten como base de sus distribuciones una sencilla, segura y potente como Debian.

Kubuntu, es una nueva distribución de Linux, que surge de la incorporación del sistema de escritorio KDE en la distribución Ubuntu (que incluye GNOME como escritorio). La distribución

Ubuntu está patrocinada por la empresa Canonical Ltd. propiedad del millonario Mark Shuttleworth.

### 3.2 C-Kermit.

C-Kermit es un paquete de *software* licenciado bajo GPL, que permite el uso combinado de comunicaciones serie y de red, ofrece una plataforma cruzada, un medio independiente y consistente para el establecimiento de conexiones, sesiones de terminal, transferencia y administración de archivos, traducción de conjuntos de caracteres y automatización de tareas de comunicación.

En el proyecto lo utilizaremos para comunicarnos con el *router*, para configurarlo o comprobar su configuración.

### 3.3 SER (SIP Express Router).

Es un servidor VoIP gratuito desarrollado bajo licencia GPL por iptel.org basado en el protocolo de Inicio de Sesión (SIP, RFC3261), diseñado para implementar infraestructuras telefónicas de gran escala y extensible mediante componentes de otros vendedores. SER guarda información de los usuarios, inicia sesiones VoIP, transmite mensajes instantáneos y garantiza la integración con otras aplicaciones, eliminando el riesgo de una “trampa de un sólo vendedor”. Cuando usemos SIP en el proyecto, este *software* ofrece los servicios de *proxy*, entre los que destacan:

- Recibe solicitudes de alta de usuarios.
- Transforma direcciones SIP según las necesidades de los agentes de usuario.
- Pone en contacto a los agentes de usuario dentro y fuera del mismo dominio.

El Router Expres SIP se basa en un modelo flexible de *plug-ins* para acoger nuevas aplicaciones. Terceras partes pueden enlazar fácilmente sus *plug-ins* con el código del servidor y dotarlo de servicios avanzados y personalizables como cuentas RADIUS, pasarela SMS o peticiones ENUM.

Su robustez y diseño permiten dar servicio a una gran ciudad con millones de usuarios con un moderno PC dual-CPU durante las horas de pico. Incluso se ha probado sobre un IPAQ PDA llegando a servir 150 cps (*calls per second* – llamadas por segundo).

La configuración de SER se realiza a través de la modificación de su *script ser.cfg* y sus políticas de admisión y enrutamiento permiten que adopte varios papeles en la infraestructura telefónica como barrera de seguridad de la red, servidor de aplicaciones o pasarela PSTN, por ejemplo.

### 3.4 Linphone.

Linphone es un paquete *software* licenciado bajo licencia GPL que ofrece un simple teléfono web que permite efectuar llamadas entre dos extremos a través de una red IP como Internet o una LAN. Linphone usa el Protocolo de Inicio de Sesión (SIP) para establecer la conexión con el *host* remoto. En este protocolo cada usuario está identificado por una dirección de la forma “sip:nombre@host”, está desarrollado para ser simple y no es compatible con H.323. Los requisitos para arrancar Linphone son:

- Linux. Cualquier distribución es adecuada.
- Gnome 1.12 o superior. Sólo se necesitan sus librerías, por lo que no es necesario que esté en ejecución.
- Una tarjeta de sonido correctamente configurada.
- Altavoces o auriculares.
- Micrófono.
- Una conexión a una red IP usando un módem, una tarjeta de red o cualquier otro medio.

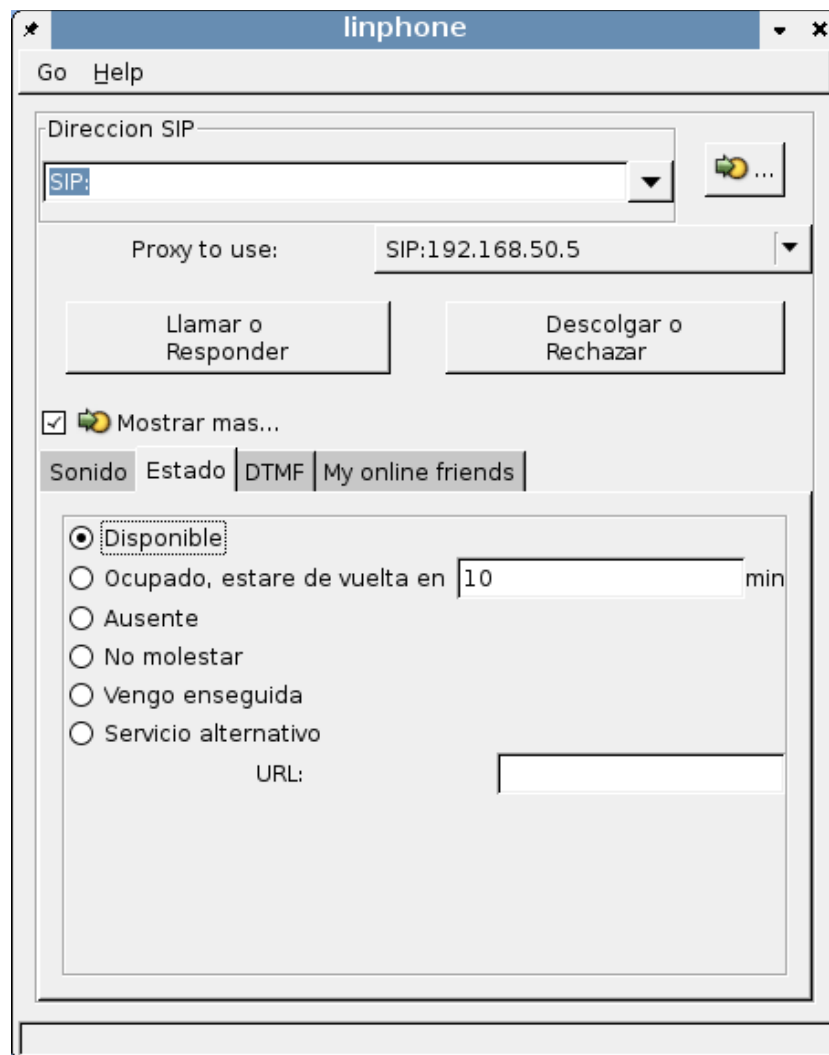


Figura 3.1: Una captura de Linphone en funcionamiento.



### 3.5 Router Cisco 1761 con VIC FXS de 2 Puertos y VIC FXO de 2 puertos.

Para simular las distintas infraestructuras del proyecto usaremos un *router* Cisco de la serie 1700 al que se le han instalado dos tarjetas VIC (*Voice Interface Card*) para dotarlo de 2 puertos FXS y otros 2 FXO.

#### 3.5.1 Router Cisco 1761.

Cualquier *router* proporciona estas funcionalidades:

- Conectividad física.
- Conectividad lógica.
- Cálculo y mantenimiento de una ruta.
- Seguridad.

El comportamiento del *router* depende del núcleo de sistema operativo que se esté ejecutando, así como de la configuración activa en cada momento. El sistema operativo de los *routers* Cisco es propietario, y se conoce con el nombre de Cisco IOS (*Internet Operating System*). La configuración activa, viene marcada por los archivos de configuración del sistema, y contienen las instrucciones y parámetros que controlan el flujo de tráfico desde y hacia los *routers*.

#### *La serie 1700.*

Los *routers* modulares de la serie 1700 están diseñados para proporcionar una plataforma integrada de *e-business* a un bajo coste para PYMEs y oficinas. Surge como solución a la demanda de flexibilidad y facilidad de uso en el entorno de la integración datos/voz/vídeo/fax, conexiones de banda ancha a Internet y políticas de seguridad.

Para aplicaciones de voz, los *routers* de la serie 1700 soportan comunicaciones analógicas y digitales permitiéndoles trabajar con la infraestructura telefónica existente de una manera sencilla y potente.

El diseño modular de la serie 1700 contempla la adicción de nuevas tarjetas de interfaz WAN (WICs) y tarjetas de interfaz de voz (VICs), permitiendo ampliar la capacidad de la red y facilitando los cambios en las tecnologías WAN y de voz. Las tecnologías WAN soportadas incluyen DSL, ISDN, enlaces T1/E1 y Frame Relay y para voz se implementa VoIP y VoFR (*Voice over Frame Relay*).



Figura 3.2: Dos modelos de la serie 1700 de Cisco.

#### *CLI (Command Line Interface).*

Mediante un programa de telnet cualquiera, como C-Kermit, hemos de conectarnos a la interfaz de consola del *router* (físicamente mediante un conector RJ45-DB25) o a una dirección IP (virtualmente con una de las 5 consolas virtuales de que dispone el *router*) para configurarlo adecuadamente.

Al conectarnos, el *router* espera nuestras órdenes de configuración o chequeo a través de la Interfaz de Línea de Comandos (CLI) del sistema operativo Cisco IOS. Desde aquí, se configuran todos los aspectos del *router*.

### 3.5.2 VIC FXS y VIC FXO.

Para dotar a un *router* de la serie 1700 de funcionalidades VoIP o VoFR se insertan en las ranuras del mismo nombre sendas tarjetas de Interfaz de Voz con distinta función:

#### **FXS.**

La tarjeta con dos puertos FXS (*Foreign Exchange Station*) nos permite conectar al *router* dos teléfonos o terminales convencionales como extensiones para recibir o realizar llamadas. Este tipo de puertos están preparados para ofrecer la señalización necesaria para el correcto funcionamiento del terminal, como es el tono de espera de marcado, el aviso de llamada entrante, la señal que se produce al descolgar *-on-hook*, o al colgar el teléfono *-off-hook*.

#### **FXO.**

Esta tarjeta VIC nos proporciona dos puertos FXO (*Foreign Exchange Office*) que nos servirán para conectar el *router* a una roseta telefónica de pared y alcanzar la red PSTN (*Plain Switched Telephone Network*) o RTB (Red Telefónica Básica); para conectar una centralita analógica (lo que aumentará el número de extensiones disponibles) o una digital (que permite también encaminamiento de tráfico de llamadas entrante) y como haremos posteriormente, para conectarle un *gateway* GSM (*Global System Mobile*) para dotar al conjunto de acceso desde y hacia dicha red GSM.

### 3.6 Gateway GSM BossLan.

Es un conversor de telefonía GSM o móvil a convencional o fija. También llamado Link GSM por su carácter “ligero”, referido a la escasa configuración de que requiere. Para hacerlo funcionar sólo es necesario introducir, en la ranura dispuesta a tal efecto, una tarjeta SIM convencional cuyo único requisito es la ausencia de un número de identificación personal (PIN, *Personal Identification Number*) y conectarlo a la red eléctrica.



Figura 3.3: Gateway GSM BossLan.

Tras unos segundos, una vez que la tarjeta SIM está en funcionamiento, sabremos que la pasarela está preparada para cursar llamadas observando los *LEDs* nombrados *NET* y *POWER*. El *LED POWER* es de color rojo y ha de estar encendido desde que arrancamos el *Gateway* GSM y el *LED NET* está fijo mientras se inicia la tarjeta SIM y parpadeante cuando la pasarela está preparada.

## 4 Desarrollo del proyecto.

### 4.1 Descripción del problema. Soluciones propuestas.

La empresa **Sun Conexión International Trade Company** se dedica al negocio inmobiliario, aprovechando la cada vez más ingente demanda de particulares de otros países europeos por residencias en el levante español, ya sea como vivienda de vacaciones o de retiro tras la jubilación. La empresa mantiene como objetivo la apertura y gestión de 3 oficinas en Europa, una en España, en el municipio de Lorca (Murcia), y las otras dos como resultado de la demanda inmobiliaria. Todos los estudios de mercado realizados apuntan a la instalación de una sucursal de la compañía en Alemania y en el Reino Unido.

Tras la apertura de estas tres oficinas y una vez alcanzado el volumen de negocio deseado, cabe esperar que el tráfico telefónico entre estas delegaciones sea considerable, con el coste subsiguiente, que corresponderá a una parte importante del total de gastos de la empresa. También se desea que los agentes comerciales puedan, desde su teléfono móvil, realizar llamadas a cualquier oficina, lo cual incrementa aun más el coste, pues las tarifas de los operadores móviles en llamadas internacionales son mayores que las de los operadores de telefonía fija.

En este marco, se plantea este Proyecto Fin de Carrera para intentar suprimir costes en la medida de lo posible y para adquirir experiencia para dotar a las urbanizaciones de extranjeros que venda la empresa de servicios como llamadas asequibles a los países de origen.

Los objetivos a conseguir tras la implantación de VoIP en la empresa son:

- Coste nulo para las llamadas entre oficinas de la empresa.

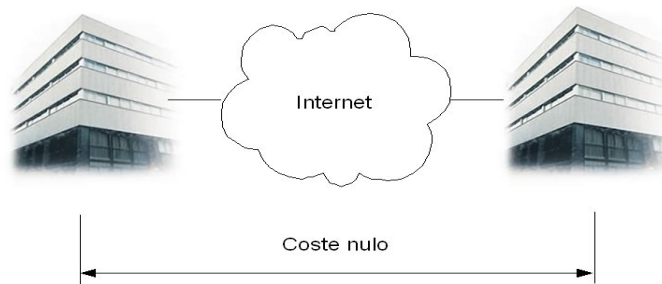


Figura 4.1: Coste nulo entre delegaciones

- Llamadas entre los agentes comerciales móviles y las oficinas a coste de llamada local o tarifa plana de empresa.

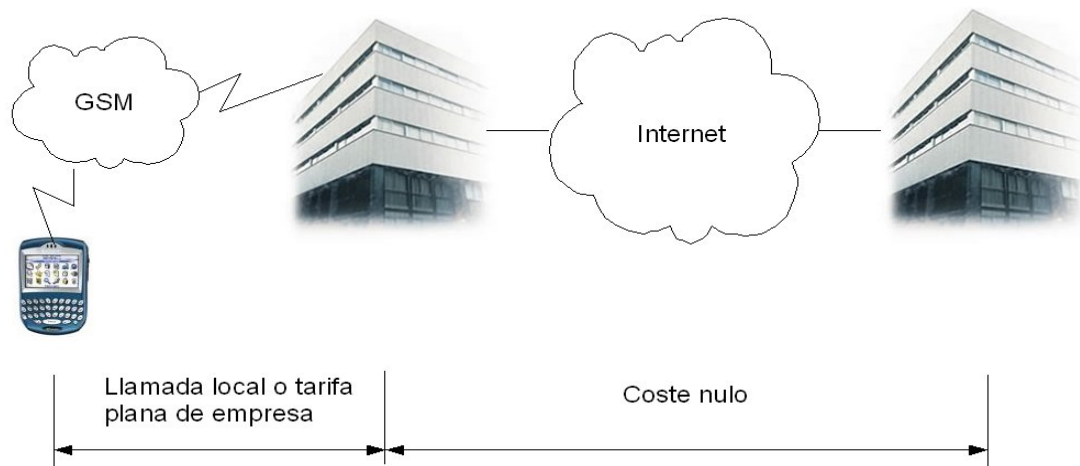


Figura 4.2: Llamadas entre comerciales y oficinas

- Llamadas entre los agentes comerciales móviles y los clientes de otros países a coste de llamada local o tarifa plana de empresa más el coste de la llamada local entre la oficina y el cliente.

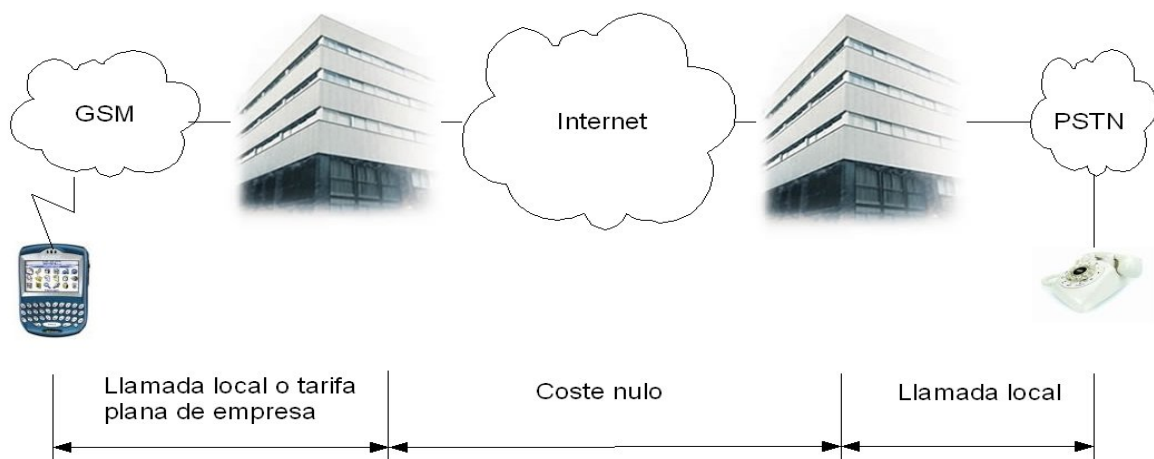


Figura 4.3: Llamadas entre comerciales y clientes de otros países

### ***Soluciones Propuestas.***

Se presentan dos soluciones distintas en base a las tecnologías utilizadas y la funcionalidad proporcionada, que nos permitirán obtener el nivel de experiencia necesaria para la migración a VoIP de la empresa y para enfrentarnos a posibles cambios de topología:

- *Pasarela VoIP H.323 desde y hacia la RTC vía GSM con teléfonos convencionales.*

Con esta infraestructura se pueden realizar y recibir llamadas desde teléfonos con la ayuda de un *gateway* GSM con una tarjeta SIM instalada y con el *router* Cisco 1761 configurado con el protocolo H.323. Este protocolo es el estándar de facto de la industria de la telefonía sobre redes IP y la opción por defecto en la inmensa mayoría de los equipos *hardware*.

En este ejemplo simularemos la arquitectura completa con las tres oficinas, dos con teléfonos analógicos y una tercera que actuará de pasarela hacia las redes telefónicas (ya sean móviles o fijas) públicas del país en cuestión.

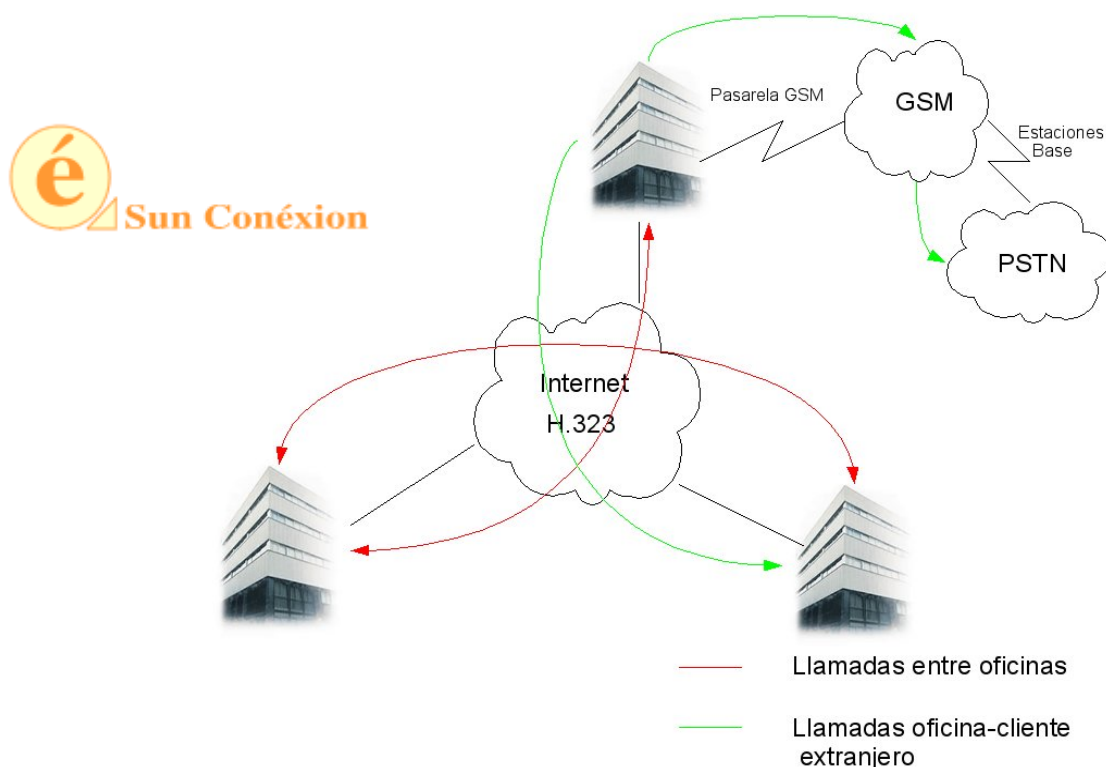


Figura 4.4: Pasarela VoIP H.323 desde y hacia la RTC vía GSM con teléfonos convencionales

- *Pasarela VoIP SIP desde y hacia la RTC vía GSM con teléfonos convencionales y ordenadores personales.*

En este modelo se cambia el protocolo de VoIP a favor de SIP (posiblemente el futuro no sólo de la VoIP sino de las comunicaciones por *Internet*) con el *router* SER y el Cisco 1761 configurado como agente de usuario y se permite la inclusión de *softphones* (Linnphone) instalados en ordenadores personales de una LAN cualquiera. Requiere más configuración pero brinda acceso a nuevas funcionalidades y es fácilmente extensible.

En este ejemplo, nos centraremos en sólo una oficina con toda la funcionalidad. Así, el *router* queda como un agente de usuario más y como pasarela entre la red puramente de datos y la de telefonía junto con el *gateway* GSM.

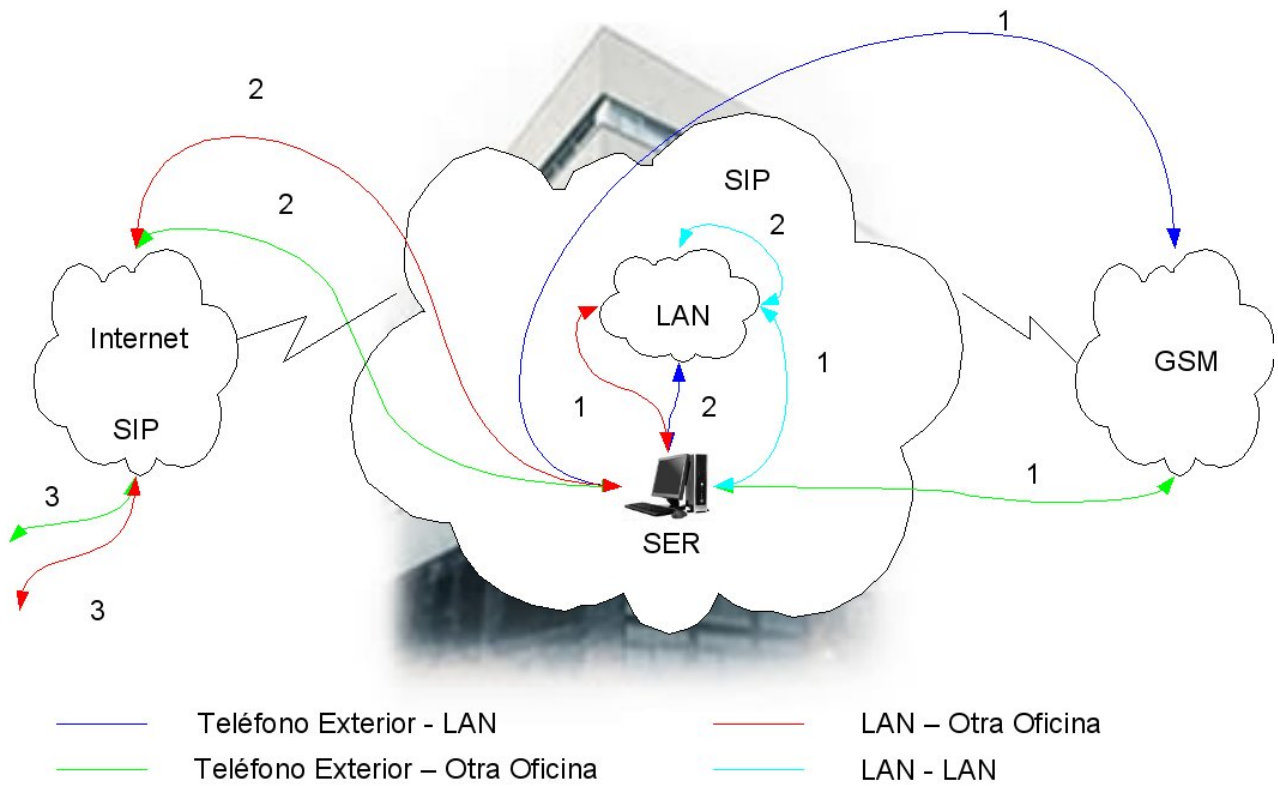


Figura 4.5: Pasarela VoIP SIP desde y hacia la RTC via GSM con teléfonos convencionales y ordenadores personales

En el resto del capítulo veremos la configuración paso a paso de las infraestructuras anteriores y un primer acercamiento a la solución que presta Cisco a la VoIP.

## 4.2 Configuración del Puesto del Trabajo. Llamada entre dos Puertos FXS del *Router* Cisco.

### 4.2.1 Preparación del Puesto de Trabajo en el Laboratorio.

En el laboratorio encontramos todo el material necesario para la simulación de las distintas soluciones arriba propuestas y que conectaremos según el siguiente diagrama:

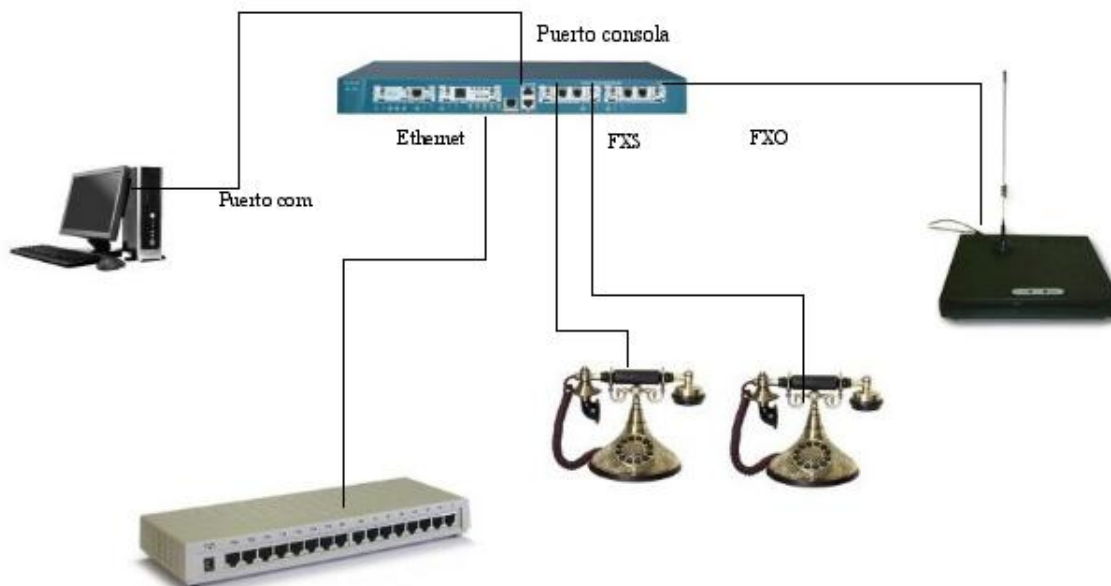


Figura 4.6: Conexión del puesto de trabajo

No importa el orden de conexión pues lo conectaremos todo con los equipos apagados y desconectados de la red eléctrica para evitar alguna (aunque pequeña) molesta descarga. Procederemos así:

- Partimos de que el ordenador personal está instalado y configurado con la distribución de GNU/Linux Kubuntu Brezzy Badger 5.04 y dispone de conexión a *Internet* (esto no es imprescindible pero mejora la instalación de programas).
- Colocamos las tarjetas de interfaz de voz en dos *slots* del *router*. Concretamente, pondremos la tarjeta de dos puertos FXS en el *slot* 2 y la de dos puertos FXO en el *slot* 3.
- Al PC le conectaremos por su puerto COM1 (interfaz DB25), el cable de consola del *router* Cisco, y el otro extremo al puerto *Console/Aux* del *router*. Esta conexión nos servirá para configurar el router desde el ordenador mediante un programa de *telnet* como C-Kermit.
- Conectaremos el puerto *Ethernet* router al *hub* para simular la red de *Internet* con un cable cruzado UTP de categoría 5e y conectores RJ-45.
- Usaremos unos latiguillos con conectores RJ-11 para unir los teléfonos analógicos a los dos puertos FXS de la tarjeta de interfaz de voz del *router*.
- Con otro latiguillo con conectores RJ-11 conectaremos el *gateway* GSM a un puerto FXO de la tarjeta de interfaz de voz del *slot* 3.
- Por último, conectaremos todos los equipos a la red eléctrica y los encenderemos todos menos el *router*.

#### 4.2.2 *Instalación de C-Kermit. Mensajes de Inicio y Configuración Básica del Router.*

Tras el montaje del puesto de trabajo en el laboratorio y su puesta en marcha instalaremos el programa de *telnet* C-Kermit en el ordenador. Para lo cual abriremos una terminal de comandos haciendo clic en el menú Aplicaciones, Herramientas del Sistema y Terminal. Se mostrará una ventana con el intérprete de comandos en el que escribiremos:

```
$sudo aptitude install ckermit
```

para instalar el paquete. Con esta orden, se busca el paquete en un índice de repositorios (servidores en *Internet* con paquetes precompilados), se descarga la última versión y se instala en el sistema satisfaciendo todas sus dependencias (si depende de otros paquetes).

Una vez instalado C-Kermit, lo ejecutaremos con:

```
$kermit -l /dev/ttyS0 -s 9600
```

que hace referencia al puerto al que hemos conectado el router (COM1 equivale en Linux a */dev/ttyS0*) y a la velocidad empleada en la conexión. Y nos aparece:

```
C-Kermit 8.0.211, 10 Apr 2004, for Linux  
Copyright (C) 1985, 2004,  
Trustees of Columbia University in the City of New York.  
Type ? or HELP for help.  
(/home/psanchez/) C-Kermit>
```

Este es la línea de comandos de C-Kermit, para conectarnos al *router* hemos de ajustar la espera de retorno de carro, pues si intentamos conectar sin hacerlo:

```
(/home/psanchez/) C-Kermit>connect  
Connecting to /dev/ttyS0, speed 9600  
Escape character: Ctrl-\ (ASCII 28, FS): enabled  
Type the escape character followed by C to get back,  
or followed by ? to see other options.  
?Carrier required but not detected.  
*****  
Hint: To CONNECT to a serial device that  
is not presenting the Carrier Detect signal,  
first tell C-Kermit to:
```

```
SET CARRIER-WATCH OFF
```

```
*****  
(/home/psanchez/) C-Kermit>
```

Nos informa de la orden que tenemos que ejecutar para conectarnos a un dispositivo como el *router*, por lo que:

```
(/home/psanchez/) C-Kermit>set carrier-watch off  
(/home/psanchez/) C-Kermit>connect
```

Acto seguido, encenderemos el *router* y observaremos en pantalla el proceso de arranque prestando especial atención a los mensajes de inicio.



Tras una secuencia inicial en pantalla en la que se nos informa de la carga del Cisco IOS en la RAM (memoria de acceso aleatorio) del *router* aparecen los mensajes a los que es imprescindible prestarles atención para solucionar posibles fallos de configuración así como la licencia del sistema operativo.

El *router* realiza un diagnóstico del *hardware*:

```
Smart Init is enabled
smart init is sizing iomem
  ID             MEMORY_REQ      TYPE
MainBoard       0X00027A80 1760
                0X000F3BB0 public buffer pools
                0X00211000 public particle pools
                0X0003B100 DSP Buffers
0X0013          0X0001B800 Card in slot 0
0X000E          0X00000000 Card in slot 2
0X0016          0X00000000 Card in slot 3
TOTAL:          0X00382F30
```

**Imprime la licencia del sistema operativo:**

```
If any of the above Memory Requirements are
"UNKNOWN", you may be using an unsupported
configuration or there is a software problem and
system operation may be compromised.
```

```
Allocating additional 6623046 bytes to IO Memory.
```

```
PMem allocated: 90358154 bytes; IOMem allocated: 10305142 bytes
```

#### Restricted Rights Legend

```
Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.
```

```
cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706
```

Informa del tipo concreto de *router* que es y de las interfaces que posee, acabando con un mensaje invitando a pulsar *enter* para iniciar la configuración:

```
Cisco IOS Software, C1700 Software (C1700-IPVOICE-M), Version 12.3(7)XR3,
RELEASE SOFTWARE (fc2)
```

```
Synched to technology version 12.3(7.11)T1
```

```
Technical Support: http://www.cisco.com/techsupport
```

```
Copyright (c) 1986-2004 by Cisco Systems, Inc.
```

```
Compiled Sat 25-Sep-04 15:29 by ealyon
```

```
Image text-base: 0x80008154, data-base: 0x8192AB18
```

```
Cisco 1760 (MPC860P) processor (revision 0x500) with 88240K/10064K bytes
of memory.
```

```
Processor board ID FOC090729EX (3566990647), with hardware revision 0000
```

```
MPC860P processor: part number 5, mask 2
```

```
1 FastEthernet interface
```

```
2 Low-speed serial(sync/async) interfaces
```

```
2 Voice FXO interfaces
```

```
2 Voice FXS interfaces
```

```
32K bytes of NVRAM.
```

```
32768K bytes of processor board System flash (Read/Write)
```

```
Press RETURN to get started!
```

Si no pulsamos *enter* inmediatamente, aparecerán en pantalla los mensajes del inicio de las interfaces del *router*, informando si hemos reiniciado el *router*, complejidad de los *codecs* utilizados, etc:

```
*Mar 1 00:00:03.962: %LINEPROTO-5-UPDOWN: Line protocol on Interface
VoIP-Null0, changed state to up
```

```
*Mar 1 00:00:04.972: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed
state to up
```

```
*Mar 1 00:00:05.981: %LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/0, changed state to up
```

```
*Mar 1 00:00:12.945: %SYS-6-LOGGERSTART: Logger process started
```

```
*Mar 1 00:00:13.033: %LINK-3-UPDOWN: Interface Serial0/0, changed state
to down
```

```
*Mar 1 00:00:13.033: %LINK-3-UPDOWN: Interface Serial0/1, changed state
to down
```

```
*Mar 1 00:00:13.538: %IPM_DSPRM-5-CODEC_CMPLX_CLI: Codec Complexity for
VIC card in slot 2 is changed.
```

```
*Mar 1 00:00:13.602: %IPM_DSPRM-5-CODEC_CMPLX_CLI: Codec Complexity for
VIC card in slot 3 is changed.
```

```
*Mar 1 00:00:14.035: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/0, changed state to down
```

```
*Mar 1 00:00:14.035: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/1, changed state to down
```

```
*Mar 1 00:00:14.035: %LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/0, changed state to up
```

```
*Mar 1 00:00:14.603: %SYS-5-CONFIG_I: Configured from memory by console
*Mar 1 00:00:15.902: %SYS-5-RESTART: System restarted --
Cisco IOS Software, C1700 Software (C1700-IPVOICE-M), Version 12.3(7)XR3,
RELEASE SOFTWARE (fc2)
Synched to technology version 12.3(7.11)T1
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2004 by Cisco Systems, Inc.
Compiled Sat 25-Sep-04 15:29 by ealyon
*Mar 1 00:00:16.014: %SNMP-5-COLDSTART: SNMP agent on host RP2 is
undergoing a cold start
*Mar 1 00:00:16.062: %LINK-5-CHANGED: Interface Serial0/0, changed state
to administratively down
*Mar 1 00:00:16.062: %LINK-5-CHANGED: Interface Serial0/1, changed state
to administratively down
*Mar 1 00:00:16.903: %LINK-3-UPDOWN: Interface Foreign Exchange Station
2/0, changed state to up
*Mar 1 00:00:16.903: %LINK-3-UPDOWN: Interface Foreign Exchange Station
2/1, changed state to up
*Mar 1 00:00:16.907: %LINK-3-UPDOWN: Interface Foreign Exchange Office EU
3/0, changed state to up
*Mar 1 00:00:16.907: %LINK-3-UPDOWN: Interface Foreign Exchange Office EU
3/1, changed state to up
```

La primera vez que iniciamos el *router* hemos de configurar sus parámetros más básicos como el nombre, contraseñas, sus interfaces para adecuarlas a nuestro ejemplo, etc.

Lo configuraremos así:

*! La exclamación significa una línea de comentario en el archivo de configuración del router.*

*! Configuramos el nombre y acceso al router.*

```
Router>enable
```

```
Router#configure terminal
```

*!Le cambiamos el nombre al router.*

```
Router(config)#hostname RP2      !RP2 o cualquiera.
```

*!Activamos la contraseña del modo privilegiado, ponemos 1234 como password.*

*!La opción secret es para que la cifre con MD5.*

```
RP2(config)#enable secret 1234
```

*!Contraseñas de consola y acceso telnet a las cinco consolas virtuales (de la 0 a la 4).*

```
RP2(config)#line console 0
```

```
RP2(config-line)#password 1234
```

```
RP2(config-line)#login
```

```
RP2(config-line)#exit
RP2(config)#line vty 0 4
RP2(config-line)#password 1234
RP2(config-line)#login
RP2(config-line)#exit
```

Como el *router* sólo dispone de un puerto *Ethernet* y queremos simular tres oficinas distintas, partiremos el rango de direcciones privadas disponible en el laboratorio para obtener dos VLANs (redes privadas virtuales). Para partirlo, cogemos el rango de direcciones (192.168.50.0/24) y lo hacemos cuatro grupos modificando la máscara de subred, quedando:

- VLAN 1: @192.168.50.0 con máscara 255.255.255.192
- VLAN 2: @192.168.50.64 con máscara 255.255.255.192
- VLAN 3: @192.168.50.128 con máscara 255.255.255.192
- VLAN 4: @192.168.50.192 con máscara 255.255.255.192

De estas cuatro VLANs usaremos sólo las tres últimas dándole estos comandos al *router*, con los que le asignamos sendas direcciones IP aleatorias dentro de cada VLAN:

```
RP2(config)# interface fastEthernet 0/0.1
RP2(config-subif)# ip address 192.168.50.60 255.255.255.192
RP2(config-subif)# encapsulation dot1Q 2 !asignamos el número 2 a la VLAN
RP2(config-subif)# no shutdown
RP2(config-subif)# exit
RP2(config)# interface fastEthernet 0/0.2
RP2(config-subif)# ip address 192.168.50.120 255.255.255.192
RP2(config-subif)# encapsulation dot1Q 3 !asignamos el número 3 a la VLAN
RP2(config-subif)# no shutdown
RP2(config-subif)# exit
RP2(config)# interface fastEthernet 0/0.3
RP2(config-subif)# ip address 192.168.50.190 255.255.255.192
RP2(config-subif)# encapsulation dot1Q 4 !asignamos el número 4 a la VLAN
RP2(config-subif)# no shutdown
RP2(config-subif)# exit
```

### 4.3 Llamada entre Puertos FXS de la misma Tarjeta. Primer Acercamiento a los *Dial-Peers*.

En este primer apartado expondremos el caso más sencillo de llamada VoIP con Cisco, usando una misma tarjeta FXS, simulando una oficina distinta en cada puerto de la misma. Con el tráfico por el *hub* haciendo las veces de *Internet* tenemos todo lo necesario para emular una llamada mediante VoIP entre dos oficinas (2 VLANs) separadas por miles de kilómetros.



Figura 4.7: Llamada entre oficinas simulada con un sólo router

Así, podremos ahorrar para la empresa todo el coste telefónico entre delegaciones, al estar utilizando una infraestructura pública como *Internet*.

#### ***Introducción a los dial-peers. VoIP con Cisco.***

Una llamada implica un origen y un destino, ahora bien, desde el punto de vista de cada *router* hay que implementar 2 *dial-peers* (compañeros de marcado) para cada flujo entrante o saliente, un *dial-peer pots* y otro *dial-peer voip*. El primero de los *dial-peers* se asigna al puerto analógico de la tarjeta VoIP (ya sea FXS o FXO) dándole un número de teléfono o patrón válido para la E.164 (recomendación de la UIT que asigna a cada país un código numérico para las llamadas internacionales) y el segundo atribuye una dirección a un número o patrón de destino.

La configuración del *router* para la llamada entre dos puertos de la misma tarjeta FXS situada en el *slot 2* quedaría:

*!Se numerarán los dial-peers pots desde el 1 como unidades (1,2,3,...) y los dial-peers voip desde el 10 como decenas (10,20,30,...)*

*!Configuramos el dial-peer 1 : relacionamos un número de teléfono y un puerto de la !tarjeta.*

```
RP2(config)#dial-peer voice 1 pots
```

```
RP2(config-dial-peer)#destination-pattern 968406179
```

```
RP2(config-dial-peer)#port 2/0
```

```
RP2(config-dial-peer)#no shutdown
```

*!Configuramos el dial-peer 10 : relacionamos un número destino y una @IP destino*

```
RP2(config)#dial-peer voice 10 voip
```

```
RP2(config-dial-peer)#destination-pattern 968467864
```

```
RP2(config-dial-peer)#session target ipv4:192.168.50.120
```

```
RP2(config-dial-peer)#no shutdown
```

*!Configuramos el dial-peer 2 : relacionamos un número de teléfono y un puerto de la tarjeta.*

```
RP2(config)#dial-peer voice 2 pots
```

```
RP2(config-dial-peer)#destination-pattern 968467864
```

```
RP2(config-dial-peer)#port 2/1
```

```
RP2(config-dial-peer)#no shutdown
```

*!Configuramos el dial-peer 20 : relacionamos un número destino y una @IP destino*

```
RP2(config)#dial-peer voice 20 voip
```

```
RP2(config-dial-peer)#destination-pattern 968406179
```

```
RP2(config-dial-peer)#session target ipv4:192.168.50.60
```

```
RP2(config-dial-peer)#no shutdown
```

Hemos de notar que la orden *destination-pattern* en un *dial-peer pots* asigna el número de teléfono propio de un puerto FXS y en un *dial-peer voip* asigna el número de destino a la dirección IP destino.

Si ahora descolgamos uno de los terminales conectados a la tarjeta FXS del *router* escucharemos el tono de marcado, acto seguido, marcaremos el número definido en el correspondiente *dial-peer* del otro teléfono y éste sonará. Al descolgarlo podremos mantener una conversación totalmente gratuita usando el *router* Cisco y la infraestructura de red del laboratorio.

#### 4.4 Pasarela desde y hacia la RTC mediante GSM y H.323.

En este modelo incluimos el *gateway* GSM que nos permitirá enlazar con la Red Telefónica Conmutada gracias a la tarjeta SIM instalada en su interior, como a cualquier teléfono móvil a través de las estaciones base. Simularemos en este caso la infraestructura completa, con 3 oficinas (3 VLANs), de las que dos cuentan con teléfonos analógicos y la restante posee la pasarela GSM mediante la que alcanzaremos la red telefónica pública. Podríamos decir, a modo de ejemplo, que las dos oficinas con teléfonos convencionales se encuentran en Alemania y el Reino Unido, y que la que hace las veces de pasarela se encuentra en España. De este modo, los agentes comerciales podrán realizar llamadas a cualquier oficina (ya sea de Alemania o Reino Unido) a precio de llamada local o tarifa plana entre móviles de empresa. Así mismo, los clientes potenciales pueden localizar a los comerciales españoles con el coste de una llamada local.

En este modelo, no es necesario un *gatekeeper* ya que no cuenta con un directorio de usuarios cambiante, son pocas extensiones dado que las oficinas son pequeñas inmobiliarias y aumentaría la complejidad del apartado en exceso sin incurrir en aumento de funcionalidades.

La infraestructura a simular quedaría:

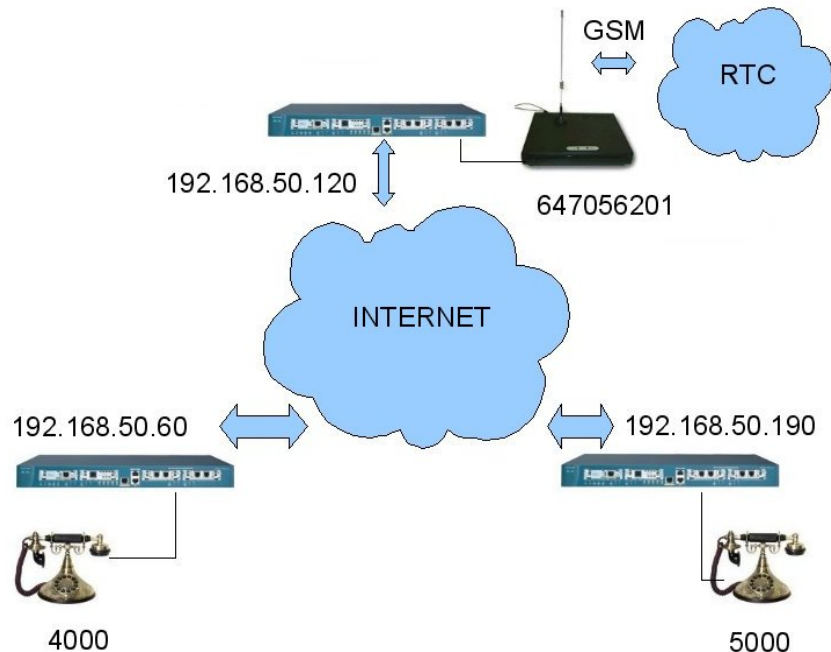


Figura 4.8: Pasarela PSTN con H.323

Como vemos en el dibujo, usamos los números de extensión 4000 y 5000 para las oficinas sin acceso a la red telefónica y el número 647056201 (número de la SIM dentro del *gateway*) para salir a la RTC.

Esto lo simularemos en el *router* de esta manera:

- La extensión 4000 corresponde al puerto 2/0 de la tarjeta FXS y su VLAN es la 2.
- La extensión 5000 corresponde al puerto 2/1 de la tarjeta FXS y su VLAN es la 4.

- El *gateway* GSM se conecta al puerto 3/0 de la tarjeta FXO y pertenece a la VLAN 3.

Al intentar esta arquitectura con los parámetros por defecto del *router* no podremos usar los puertos de la tarjeta FXO situada en el *slot* 3. Para poder activar los 4 puertos de voz ( 2 FXS y 2 FXO ) hemos de cambiar la complejidad del conjunto de *codecs* de las tarjetas a “*medium*”, para que el DSP (procesador digital de señal) del *router* pueda gestionar los 4 puertos. Tomamos la precaución de desactivar los puertos de las tarjetas para poder cambiar la complejidad del *codec*. Con desactivar uno anula los dos puertos de la tarjeta. En nuestro ejemplo, hemos colocado la FXS en el *slot* 2 y la FXO en el *slot* 3 por lo que, al iniciar el *router*, sólo habrá activado los dos de la FXS (por su número menor de *slot*) ante la falta de recursos del DSP. Los comandos a introducir en el *router* serían:

```
RP2>enable !Para entrar al modo privilegiado, tras lo que nos pide una contraseña
```

```
RP2#configure terminal
```

```
RP2 (config) #voice-port 2/0 !Para entrar a la configuración de puerto de voz
```

```
RP2 (config-voiceport) #shutdown !Lo desactivamos
```

```
RP2 (config-voice-port) #exit
```

*!En el menú de configuración de cada tarjeta cambiamos la complejidad de los codecs.*

```
RP2 (config) #voice-card 2
```

```
RP2 (config-voicecard) #codec complexity medium
```

```
RP2 (config-voice-card) #exit
```

```
RP2 (config) #voice-card 3
```

```
RP2 (config-voicecard) #codec complexity medium
```

```
RP2 (config-voicecard) #exit
```

Ahora borraremos antes de crear unos nuevos, los *dial-peers* del ejemplo del punto anterior:

```
RP2 (config) #no dial-peer voice 1 pots
```

```
RP2 (config) #no dial-peer voice 10 voip
```

```
RP2 (config) #no dial-peer voice 2 pots
```

```
RP2 (config) #no dial-peer voice 20 voip
```

Tras esto, ya podemos crear los *dial-peers* de este modelo. Hay que resaltar que al ser H.323 la opción por defecto del *router* no hay que tocar parámetros de protocolos voz ni de *codecs*.

*!Se numerarán los dial-peers pots desde el 1 como unidades (1,2,3,...) y los dial-peers voip desde el 10 como decenas (10,20,30,...)*

*!Configuramos el dial-peer 1 : relacionamos un número de teléfono y un puerto de la tarjeta*

```
RP2 (config) #dial-peer voice 1 pots
```



```
RP2 (config-dial-peer)#destination-pattern 4000
```

```
RP2 (config-dial-peer)#port 2/0
```

```
RP2 (config-dial-peer)#no shutdown
```

*!La orden exit no es necesaria para pasar entre interfaces que cuelgan del mismo menú. Configuramos el dial-peer 10 : relacionamos cualquier número destino y una @IP destino*

```
RP2 (config)#dial-peer voice 10 voip
```

*!Con este destination-pattern nos referimos a cualquier número de 9 cifras (9 puntos)*

```
RP2 (config-dial-peer)#destination-pattern .....
```

```
RP2 (config-dial-peer)#session target ipv4:192.168.50.120
```

```
RP2 (config-dial-peer)#no shutdown
```

*!Configuramos el dial-peer 2 : relacionamos un número de teléfono y un puerto de la tarjeta.*

```
RP2 (config)#dial-peer voice 2 pots
```

```
RP2 (config-dial-peer)#destination-pattern 5000
```

```
RP2 (config-dial-peer)#port 2/1
```

```
RP2 (config-dial-peer)#no shutdown
```

*!Configuramos el dial-peer 20 : relacionamos un número destino y una @IP destino*

```
RP2 (config)#dial-peer voice 20 voip
```

```
RP2 (config-dial-peer)#destination-pattern 4000
```

```
RP2 (config-dial-peer)#session target ipv4:192.168.50.60
```

```
RP2 (config-dial-peer)#no shutdown
```

*!Configuramos el dial-peer 30 : relacionamos un número destino y una @IP destino*

```
RP2 (config)#dial-peer voice 30 voip
```

```
RP2 (config-dial-peer)#destination-pattern 5000
```

```
RP2 (config-dial-peer)#session target ipv4:192.168.50.190
```

```
RP2 (config-dial-peer)#no shutdown
```

*!Configuramos el dial-peer 3 : relacionamos el número de teléfono de la SIM y un puerto de la tarjeta FXO.*

```
RP2 (config)#dial-peer voice 3 pots
```

```
RP2 (config-dial-peer)#destination-pattern 647056201
```

```
RP2 (config-dial-peer)#port 3/0
```

```
RP2 (config-dial-peer)#no shutdown
```

*!Configuramos el dial-peer 4 : relacionamos el número de teléfono de la SIM y un número de extensión (4 cifras = 4 puntos) y un puerto de la tarjeta.*

*!De esta manera, cuando alguien llame al número de la SIM, recibirá un tono de marcado invitándole a introducir el número de cuatro cifras de la extensión.*

```
RP2 (config)#dial-peer voice 4 pots
```

```
RP2 (config-dial-peer) #destination-pattern 647056201....  
RP2 (config-dial-peer) #port 3/0  
RP2 (config-dial-peer) #no shutdown  
RP2 (config-dial-peer) #exit
```

Con esta sencilla configuración, hemos dotado a dos oficinas remotas de acceso a la Red Telefónica Conmutada de un país sin coste (exceptuando el coste de llamadas dentro de ese dicho país) y también ahorramos dinero para el cliente, que ahora puede llamar a cualquier oficina de cualquier país de la empresa al coste de una llamada nacional.

#### **4.5 Pasarela desde y hacia la RTC mediante GSM y SIP. Inclusión de LAN.**

En este apartado vamos a centrarnos en una sólo oficina central del conjunto. En esta oficina colocaremos el servidor SIP (único) de la empresa que permite que una gran cantidad de usuarios se suscriban para obtener acceso a la infraestructura VoIP de la compañía. Así, aumentamos el número de extensiones disponibles, pues contamos con cada ordenador de una posible LAN y las ya disponibles del *router*, que actuará como agente de usuario SIP y como traductor de números de extensión y entrantes desde el *gateway* GSM. Si a esto le sumamos que cada usuario posee un identificador inequívoco, el URI SIP (*Universal Resource Identifier SIP*, identificador universal SIP de recurso), permitimos a cualquier usuario de SIP llamarlo a través de de *Internet* sin coste usando, por ejemplo, botones *clic to dial* (pulsar para marcar) en la página web de la empresa.

La infraestructura que pretendemos emular queda:

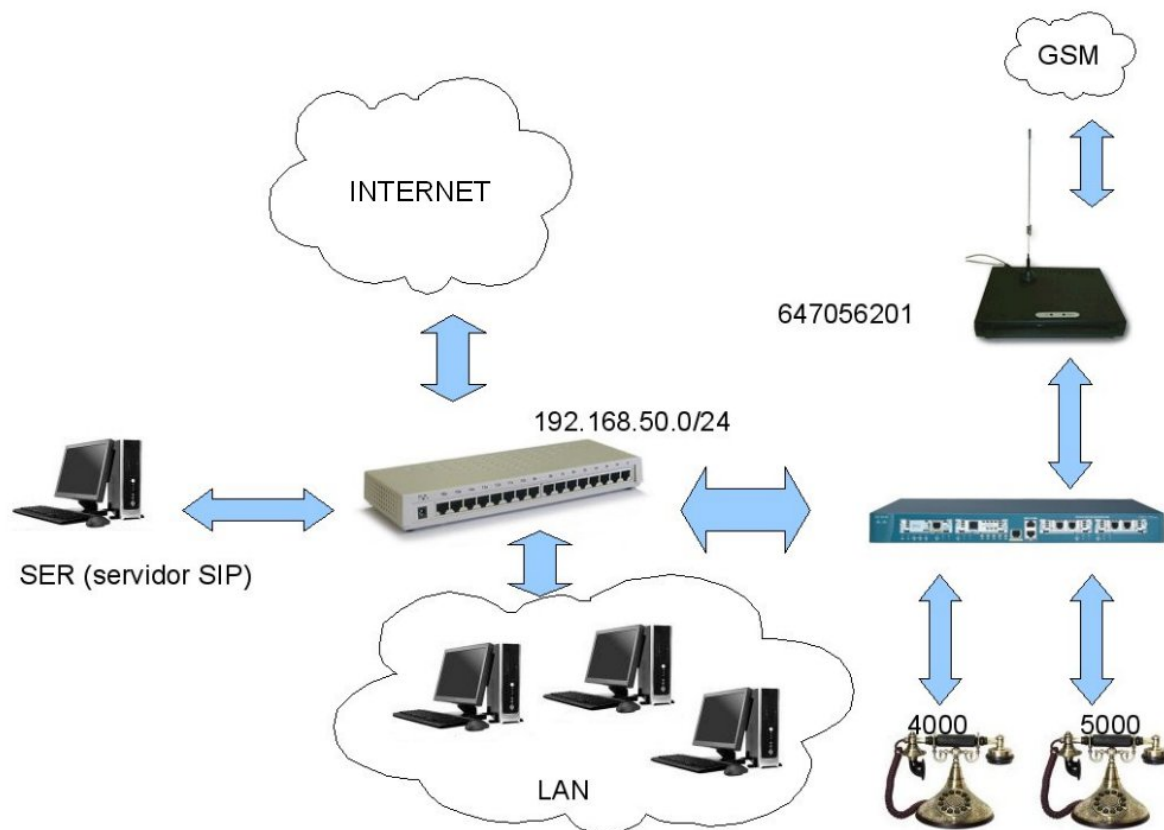


Figura 4.9: Simulación de una LAN y teléfonos de una oficina

Para simularlo en el laboratorio hemos de hacer algunos cambios en la configuración de los equipos, como:

- Para simular la LAN de la empresa, conectaremos el PC al *hub* con un cable UTP de categoría 5e con conectores RJ-45.
- En el PC instalaremos el servidor SIP (SER) y un agente de usuario (Linphone).
- Le añadimos al ordenador un micrófono y unos altavoces o auriculares de buena calidad pues repercutirán directamente en la correcta transmisión y recepción de la llamada.

El conexionado del laboratorio sería:

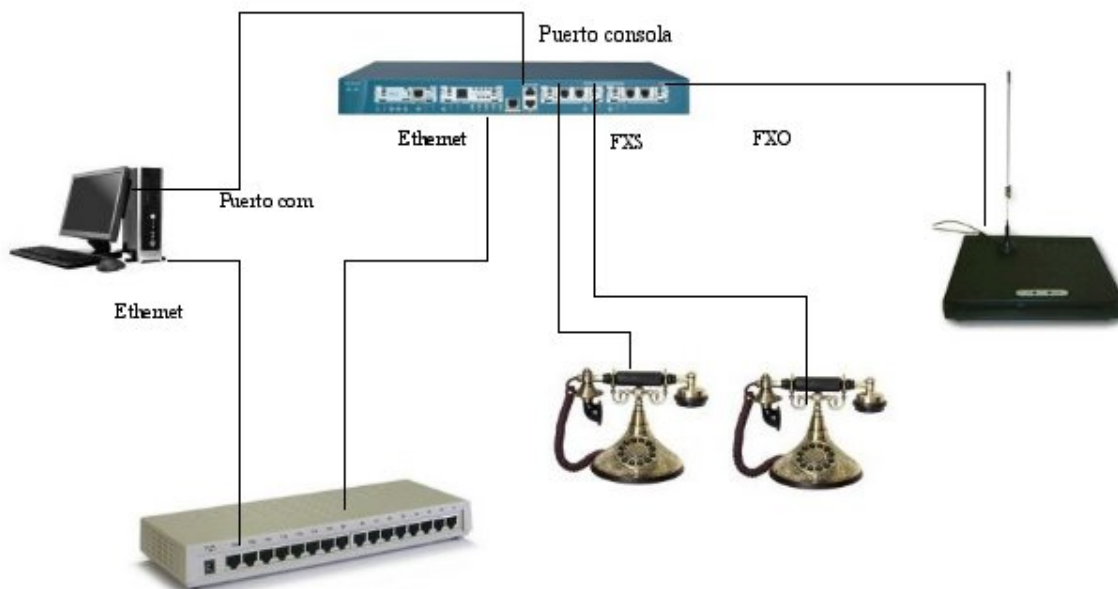


Figura 4.10: Conexionado para la simulación con SIP

En el resto del apartado veremos la instalación y configuración del SIP Express Router (SER), del agente de usuario del PC (Linphone) y del *router* (como agente de usuario y cambios en los *dial-peers*).

#### 4.5.1 SIP Express Router (SER). Instalación y Configuración.

La solución de [iptel.org](http://iptel.org) como servidor SIP ofrece todo lo que necesitamos para dotar de las últimas funcionalidades del protocolo SIP a nuestra infraestructura de VoIP. SER actúa de *registrar*, de *proxy* y de *router*.

La compañía [iptel.org](http://iptel.org) ha licenciado el código fuente del servidor sólo para GNU/Linux y bajo la licencia GPL, lo que permite que programadores de todo el mundo puedan mejorarlo e incorporarle nuevas funciones.

Para obtener el código del servidor o un paquete precompilado para nuestra distribución de GNU/Linux (en nuestro caso elegiremos Debian) podemos descargarlo de la página [web http://www.iptel.org](http://www.iptel.org) en la sección de descargas (*downloads*). Tras guardarlo en el disco duro del PC, abrimos una terminal de comandos (Aplicaciones, Herramientas del Sistema, Terminal) y nos situamos en el directorio donde hayamos descargado el paquete y ejecutamos:

```
$sudo dpkg -i ser_0.9.4-0.1_i386.deb
```

Lo que resulta:

```
Seleccionando el paquete ser previamente no seleccionado.  
(Leyendo la base de datos ...  
136707 ficheros y directorios instalados actualmente.)  
Desempaquetando ser (de ser_0.9.4-0.1_i386.deb) ...  
Configurando ser (0.9.4-0.1) ...
```

```
There are errors in the configuration file. Please fix them first
```

```
0(25517) WARNING: fix_socket_list: could not rev. resolve 192.168.50.5
0(25517) WARNING: fix_socket_list: could not rev. resolve 192.168.50.5
config file ok, exiting...
```

El instalador encuentra un error porque SER no está diseñado para usarse en direcciones privadas, aunque el paquete arregla el archivo de configuración. Con lo que el servidor queda instalado y funcionando. Tras poner en funcionamiento el SER, podemos añadirle algunos *scripts* descargables desde el mismo sitio que el núcleo del servidor que le proporcionan competencias con RADIUS, LDAP, configuración vía *web*, etc.

### ***El script de configuración general.***

La configuración general de SER se realiza modificando un *script* se encuentra tras la instalación en la ruta */etc/ser.cfg*. Este *script* define el comportamiento del servidor y proporciona las reglas para la admisión de usuarios y la emisión de mensajes.

Tras la rutina de inicio, el *script* se mantiene a la espera de la llegada de mensajes para cualquiera de los dominios en los que opera SER (en nuestro caso, un único dominio que corresponde al rango de direcciones privado del laboratorio, 192.168.50.0/24). Cuando llega un mensaje destinado a nuestro dominio, se observa si éste es del tipo *REGISTER*. Si lo es, el servidor actúa como un *registrar* SIP y actualiza la base de datos de contactos de los usuarios. Opcionalmente, verifica la identidad del usuario para impedir la manipulación de los contactos. Cuando el mensaje no es del tipo *REGISTER*, se procesa usando la base de datos de localización de los usuarios. Si el usuario que emite el mensaje está en la base de datos, el *script* le reenvía el mensaje, por el contrario, si no lo encuentra devolverá al emisor una respuesta 404.

Este *script* tiene algunas limitaciones con las opciones por defecto, como:

- Para permitir compatibilidad con cualquier demonio de *MySQL*, la autenticación está desactivada.
- Incluso activando la autenticación, no hay relación entre el nombre de usuario dado en el proceso de autenticación y la dirección de registro, por lo que un usuario puede conseguir entrar al servidor con un nombre y después registrarse con otro. Se arregla con el uso de la función *check\_to* (comprobar) del módulo *auth* (autenticación), pero requiere la modificación del *script*.
- Con la autenticación activada descomentando las órdenes del *script* referentes al control de acceso, las contraseñas viajarán por la red en forma de texto plano.
- No hay implementados planes de marcado, lo que causa que cualquier usuario puede llamar a cualquier destino. Esto no es interesante si queremos otorgar privilegios de llamada como acceso a la red telefónica pública.
- El *script* asume que todos los usuarios usan la dirección o el nombre del servidor SER como la parte de su URI correspondiente a la dirección. Pueden especificarse otras direcciones con la función *alias*.

Vamos a detallar la estructura del *script* de configuración paso a paso:

- Observamos una primera parte de presentación con el número de versión y una breve descripción de la función del *script* en forma de “*script* de configuración simple de inicio rápido”. Tras lo que encontramos la definición de los parámetros globales de configuración

que definirán el comportamiento del servidor. Se ha traducido la parte de los comentarios para un mejor entendimiento:

```
# $Id: ser.cfg,v 1.25.2.1 2005/02/18 14:30:44 andrei Exp $
#
# script de configuración simple de inicio rápido
#
# ----- parámetros globales de configuración -----

#debug=3          # nivel de depurado (en la línea de comandos:-dddddddddd)

#fork=yes

#log_stderr=no   # (en la línea de comandos: -E)

/* Descomenta estas líneas para entrar en el modo de depurado

fork=no

log_stderr=yes

*/

check_via=no     # (en la línea de comandos: -v)

dns=no          # (en la línea de comandos: -r)

rev_dns=no      # (en la línea de comandos: -R)

#port=5060

#children=4

fifo="/tmp/ser_fifo"
```

En esta parte se ha definido el comportamiento del servidor, imponiéndole el uso de multihilo, el nivel de depurado, el uso o no de DNS, si va a mostrar los errores por pantalla o la cola FIFO (*First In, First Out*, el primero que entra es el primero que sale) implementada por SER.

La siguiente parte del *script* corresponde a la carga de los módulos. Además de los módulos que trae SER en el paquete instalado aquí hemos de indicarle la ruta para que cargue módulos de terceras partes como el de RADIUS o el de LDAP:

```
# ----- carga de módulos -----
# Descomenta esta línea si quieres usar una base de datos MySQL
```

```

#loadmodule "/usr/lib/ser/modules/mysql.so"

loadmodule "/usr/lib/ser/modules/sl.so"
loadmodule "/usr/lib/ser/modules/tm.so"
loadmodule "/usr/lib/ser/modules/rr.so"
loadmodule "/usr/lib/ser/modules/maxfwd.so"
loadmodule "/usr/lib/ser/modules/usrloc.so"
loadmodule "/usr/lib/ser/modules/regarstrar.so"
loadmodule "/usr/lib/ser/modules/textops.so"

# Descomenta esta línea para conseguir autenticación
# ;asegúrate de tener cargado mysql.so!
#loadmodule "/usr/lib/ser/modules/auth.so"
#loadmodule "/usr/lib/ser/modules/auth_db.so"

```

El uso de esta parte es sencillo, sólo hemos de indicar la ruta donde se encuentra el módulo y cargarlo con la orden *loadmodule*.

Ahora viene la configuración de los módulos cargados. Se usa para darles instrucciones específicas de funcionamiento.

```

# ----- configuración de parámetros específicos de módulo -----

# -- parámetros del módulo usrloc --

modparam("usrloc", "db_mode", 0)

# Descomenta esta línea para usar una base de datos MySQL
# para almacenamiento persistente y comenta la línea anterior
#modparam("usrloc", "db_mode", 2)

# -- parámetros del módulo auth --
# Descomenta si se ha cargado el módulo auth
#
#modparam("auth_db", "calculate_ha1", yes)
#
# Si has puesto el parámetro caluculate_ha1 como "yes", descomenta también
# el siguiente parámetro
#
#modparam("auth_db", "password_column", "password")

# -- parámetros del módulo rr --
# Si tienes problemas con el agente de usuario, añádele valor al parámetro

```

```
# ;lr para arreglarlo
modparam("rr", "enable_full_lr", 1)
```

Así ajustamos el comportamiento de los módulos a nuestro entorno particular, permitiendo solventar problemas con agentes de usuario o bases de datos mal configuradas.

La última parte del *script* se encarga de las decisiones de reenvío y enrutado de los mensajes SIP. Es aquí donde podemos hacer uso de las funciones que SER pone a nuestra disposición para reescribir URIs completas o partes de ellas según nos sea necesario. Sólo tenemos que editar el *script* y colocar nuestro código en la rutina principal de enrutado.

```
# ----- lógica de enrutado -----

# lógica principal de enrutado

route{

    # comprobación inicial de integridad -- mensajes con
    # max_forwards==0, o mensajes excesivamente largos
    if (!mf_process_maxfwd_header("10")) {
        sl_send_reply("483", "Too Many Hops");
        break;
    };
    if (msg:len >= 2048 ) {
        sl_send_reply("513", "Message too big");
        break;
    };

    # grabamos la ruta de todos los mensajes - así nos aseguramos de
    # que los siguientes mensajes vayan a través de nuestro proxy.
    # esto es recomendable si los agentes de usuario de envío y
    # recepción usan distinto protocolo
    if (!method=="REGISTER") record_route();

    # los mensajes siguientes sin diálogo deberían tomar el
    # camino determinado por record_routing
    if (loose_route()) {
        # marcamos la lógica de enrutado en consultas al SER
        append_hf("P-hint: rr-enforced\r\n");
        route(1);
        break;
    };
};
```



```

if (!uri==myself) {
    # marcamos la lógica de enrutado en consultas al SER
    append_hf("P-hint: outbound\r\n");
    route(1);
    break;
};

# si las consultas son de otros dominios usa usrloc
# (en el caso de que no funcione, usa el siguiente comando
# con nombres propios y direcciones)
if (uri==myself) {

    if (method=="REGISTER") {

# Descomenta para conseguir autenticación
#         if (!www_authorize("iptel.org", "subscriber")) {
#             www_challenge("iptel.org", "0");
#             break;
#         };

        save("location");
        break;
    };

    lookup("aliases");
    if (!uri==myself) {
        append_hf("P-hint: outbound alias\r\n");
        route(1);
        break;
    };
    # los destinos nativos SIP se manejan con nuestra base de
    # datos usrloc
    if (!lookup("location")) {
        sl_send_reply("404", "Not Found");
        break;
    };
};
append_hf("P-hint: usrloc applied\r\n");
route(1);
}

route[1]

```

```

{
    # usamos reenvío basado en estado
    # incluso para UDP2TCP
    if (!t_relay()) {
        sl_reply_error();
    };
}

```

Como hemos visto, en esta última parte terminamos de configurar la autenticación si hacemos uso de ella y podemos añadir nuestro propio código.

Para ver lo sencillo que es adaptar SER a nuestro entorno de trabajo vamos a incluir una porción de código en el *script* para redirigir las peticiones entrantes del usuario [6000@192.168.50.5](mailto:6000@192.168.50.5).

En el laboratorio sólo disponemos de un ordenador, que hará las veces tanto servidor como de agente de usuario. Esto se presta a problemas, pues el puerto por defecto de SIP tanto para el SER como para el agente de usuario es el 5060. Cuando el *router* Cisco llama al agente de usuario del PC, ya sea desde una extensión directamente conectada o desde una llamada entrante de la red pública, manda su mensaje INVITE a la dirección SIP [6000@192.168.50.5:5060](mailto:6000@192.168.50.5:5060), que entrará en conflicto con el servidor instalado en la misma dirección IP. Lo solucionamos añadiendo el siguiente código en la rutina principal de enrutado:

```

## Conversion de puerto para sip:6000@192.168.50.5:5060 marcado
## automaticamente por CISCO
## a sip:psanchez2@192.168.50.5:5080 para evitar conflictos de
## puerto con SER
if (uri=~"6000@192.168.50.5") {
    rewriteuser("psanchez2");
    rewriteport("5080");
};

```

También le cambiamos el nombre de usuario a “psanchez2” para permitir una configuración más sencilla del agente de usuario Linphone.

En el código vemos una primera sentencia que “busca” mensajes destinados a la URI problemática y que al encontrarla se vale de las órdenes *rewriteuser* (cambia la parte de usuario de la URI) y *rewriteport* (cambia el puerto de la URI) para transformar de [6000@192.168.50.5:5060](mailto:6000@192.168.50.5:5060) a [psanchez2@192.168.50.5:5080](mailto:psanchez2@192.168.50.5:5080). Así el servidor SIP y el agente de usuario pueden funcionar en una misma máquina con una sólo tarjeta de red sin problemas.

Tras nuestra modificación del *script* de */etc/ser.cfg*, hemos de reiniciar el demonio *serctl*. Este demonio es el que se encarga de recibir peticiones, crear nuevos hilos de proceso que las atiendan y terminar dichos hilos. Para reorganizarlo hemos de introducir lo siguiente en un terminal de comandos:

```
$sudo serctl restart
```

Con esta orden debería ser suficiente según los programadores del servidor pero no es así. Sólo tras un reinicio total del ordenador que alberga el servidor surtieron efecto los cambios realizados en el *script*.

#### 4.5.2 *Linphone. Instalación y Configuración.*

Linphone es el agente de usuario de licencia libre más completo de la actualidad. Está

orientado al uso con SIP desde su primera versión, por lo que cuenta con experiencia en este campo. Existían otros candidatos a su uso en este proyecto como Kphone o X-Lite, pero el primero se desechó por su extrema sencillez y aunque el segundo se acercaba a Linphone en funcionalidad, no tiene licencia libre y está escrito en el lenguaje de programación Java, por lo que resulta bastante pesado (Linphone está escrito en C++).

Para instalarlo usaremos de nuevo los repositorios de Kubuntu con la orden siguiente desde un terminal de comandos:

```
$sudo aptitude install linphone
```

Con lo que lo tendremos instalado y listo para configurar. Vamos a ver detalladamente la configuración de Linphone:

- Para iniciarlo podemos simplemente teclear “linphone” en un terminal o haciendo “clic” en su icono del menú de aplicaciones.

- Tras arrancarlo, veremos una pantalla similar a ésta:

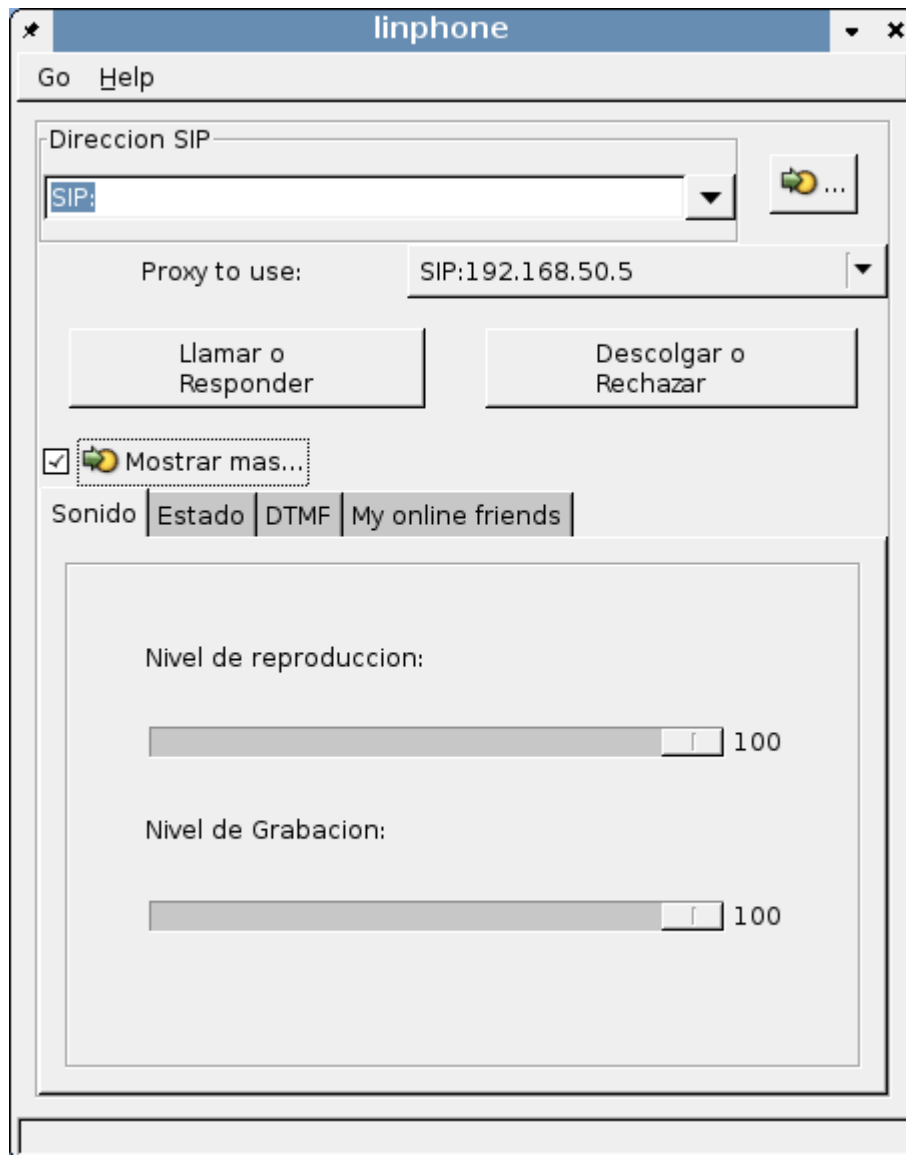


Figura 4.11: Pantalla principal de Linphone

- Nos dirigimos a la opción *Go* y luego a *Preferences*, con lo que entraremos en el menú de configuración:

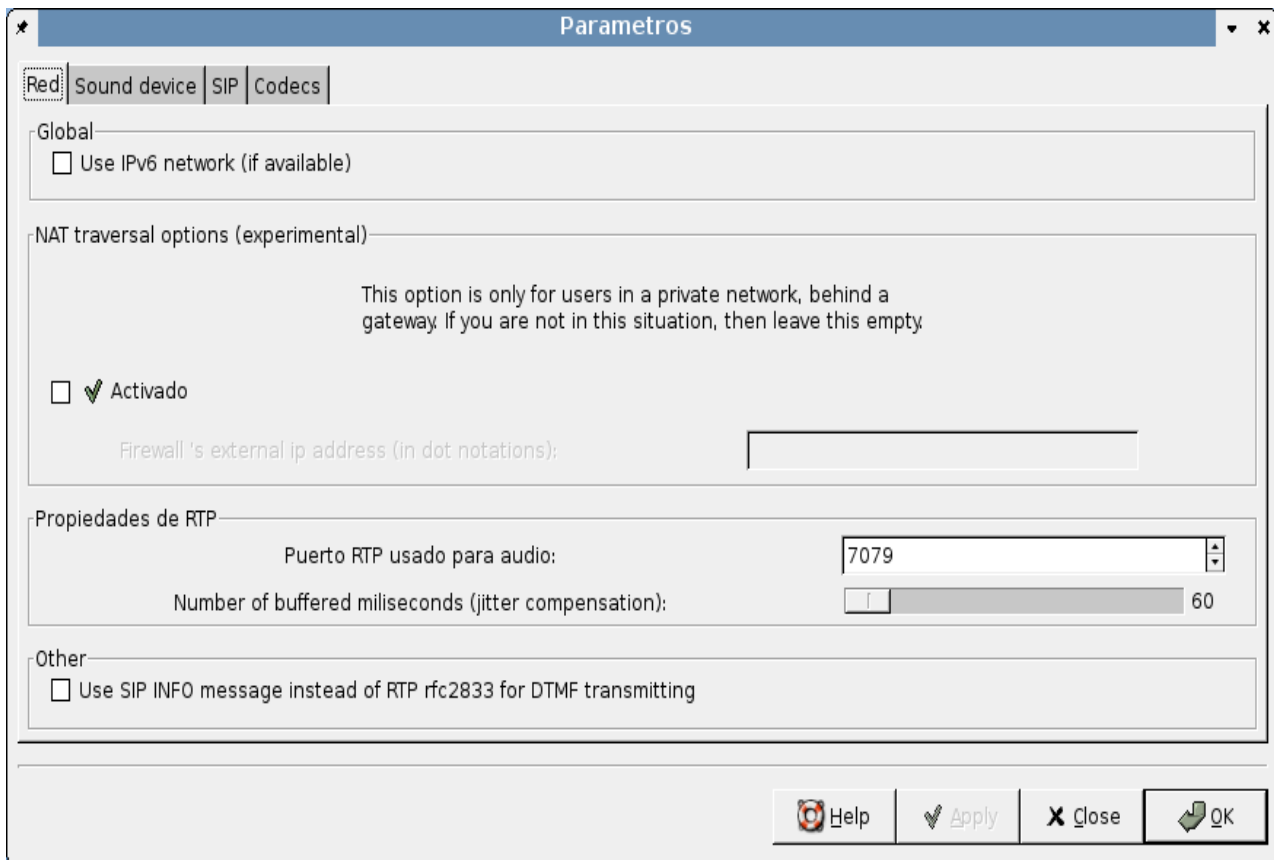


Figura 4.12: Configuración de red en Linphone

- En esta pestaña de configuración (Red) dejaremos todas las opciones tal y como vienen por defecto.
- Activaremos la opción *Use IP v6 network (if available)* sólo si dispusiéramos de un entorno completo basado en IP v6, protocolo que sustituirá a IP v4 pero que no ha sido implantado totalmente. En el laboratorio hay una red *Ethernet* IP v4 y dejaremos esta opción desactivada.
- Usaremos la opción *Activado* del cuadro *NAT transversal options* si estuviéramos detrás de un *firewall* o *router* que haga traducción de direcciones. Como el servidor y el agente están en el mismo rango privado de direcciones, la dejaremos sin marcar.
- La siguiente opción especifica el puerto usado para el audio por el protocolo RTP y el tamaño del *buffer* para la compensación del retardo. Dejamos el puerto y el *buffer* como está.
- La última opción de la pestaña Red incluye la posibilidad de usar el mensaje INFO de SIP para transmitir tonos DMTF en lugar del RFC 2833 de RTP. No la marcamos porque no enviaremos tonos DTMF en las pruebas.

- En la pestaña *Sound Device* especificaremos el dispositivo de sonido utilizado (la tarjeta de sonido), el receptor de sonido (el micrófono) y el tono de llamada. En nuestro caso usaremos la tarjeta de sonido que está integrada en la placa con el demonio de sonido de Linux ALSA (*Advanced Linux Sound Architecture*) y como tono de llamada el que viene por defecto.

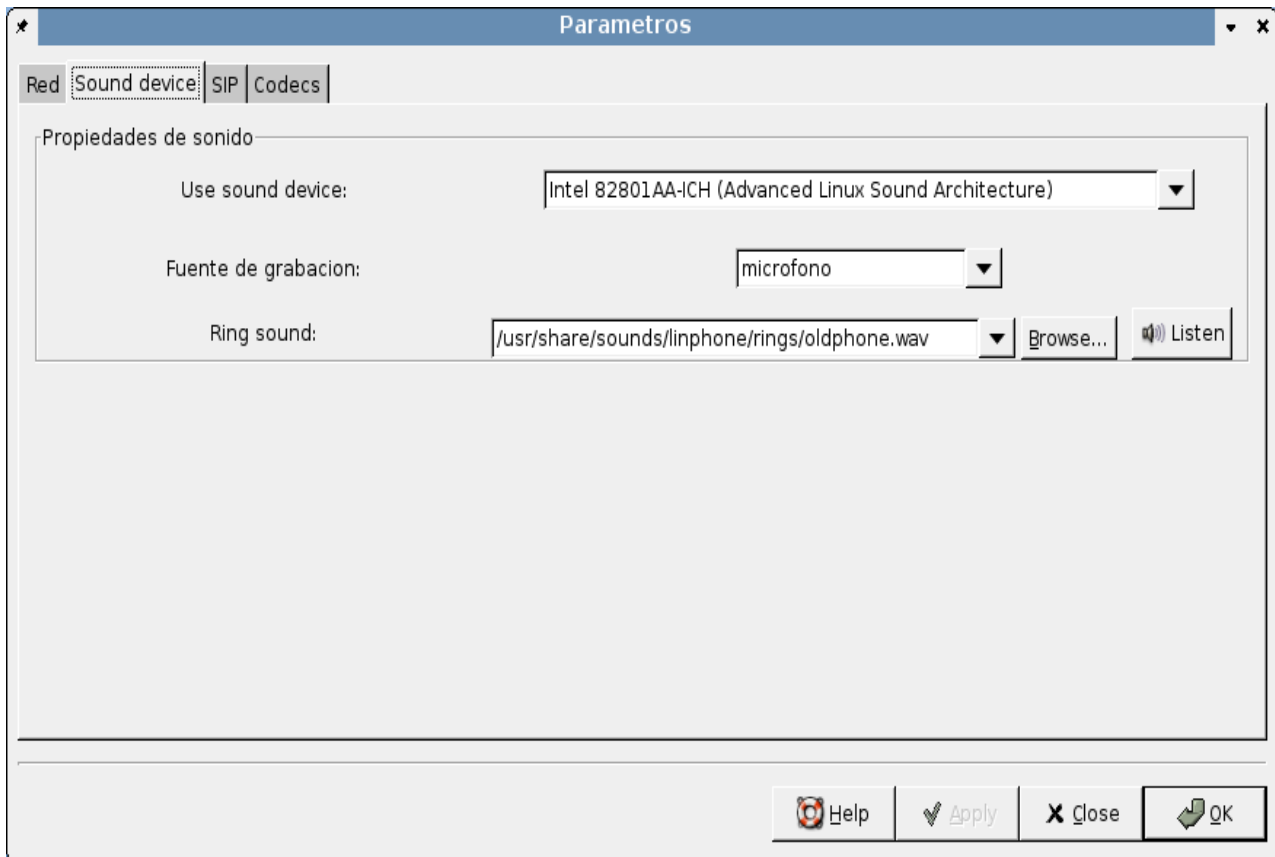


Figura 4.13: Configuración del hardware de sonido en Linphone

- La pestaña SIP es la más importante ya que contiene los parámetros relacionados al protocolo SIP, como son:
  - Puerto SIP: ponemos el 5080 como hemos especificado en la configuración de SER.
  - Identidad: aquí ponemos nuestro nombre de usuario.
  - Servicios remotos: en esta casilla introducimos con *Add proxy/regar* la dirección del servidor SIP donde se va a registrar Linphone. En nuestro ejemplo, la misma dirección IP del ordenador.
  - *Authentication Information*: con la opción bajo este cuadro borramos toda la información SIP relativa a usuarios y claves.

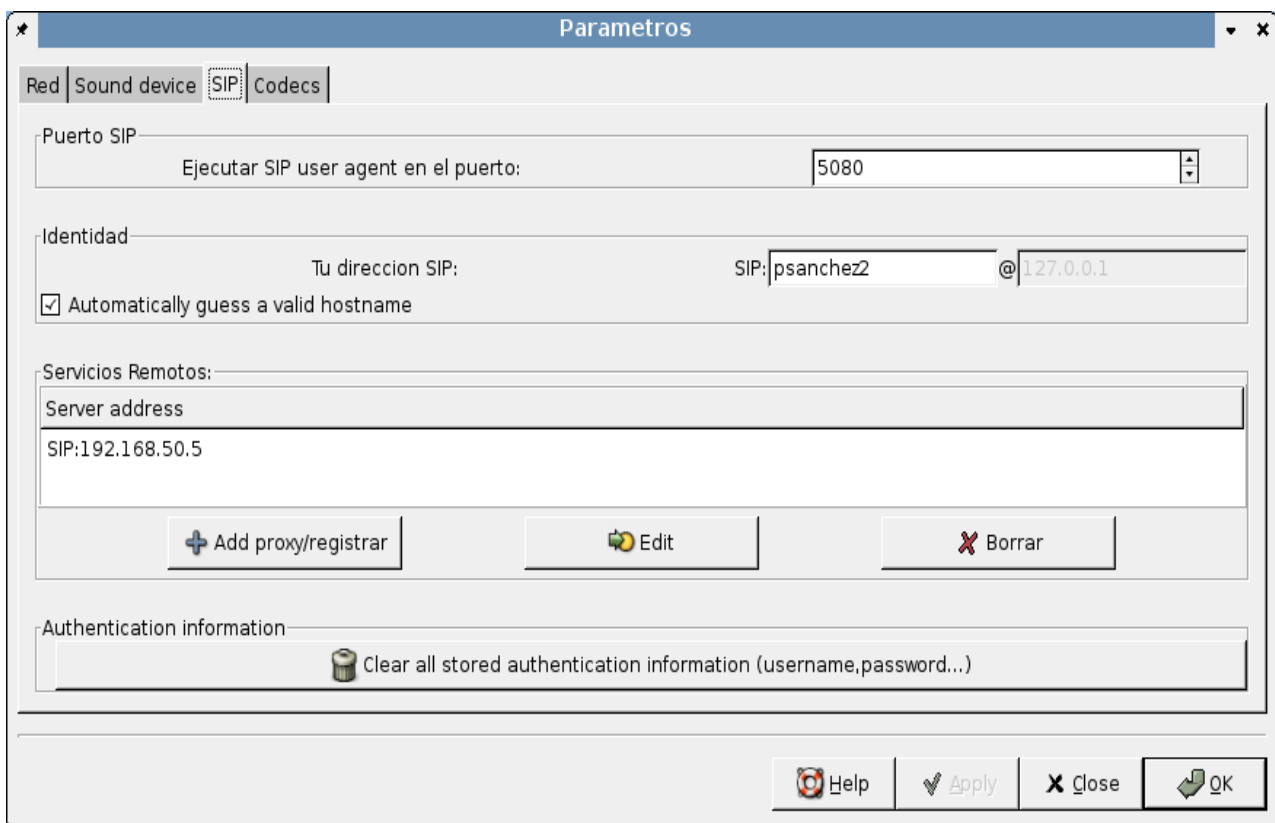


Figura 4.14: Configuración del protocolo SIP en Linphone

- La última parte de la configuración se ocupa de los *codecs*. Los ordenaremos para que se muestran tal y como se observa en la imagen superior para acomodarlos con la posterior configuración del *router*. Si se nos muestra algún *codec* en rojo no debemos utilizarlo porque requiere demasiado ancho de banda y no se aseguraría la calidad de transmisión.

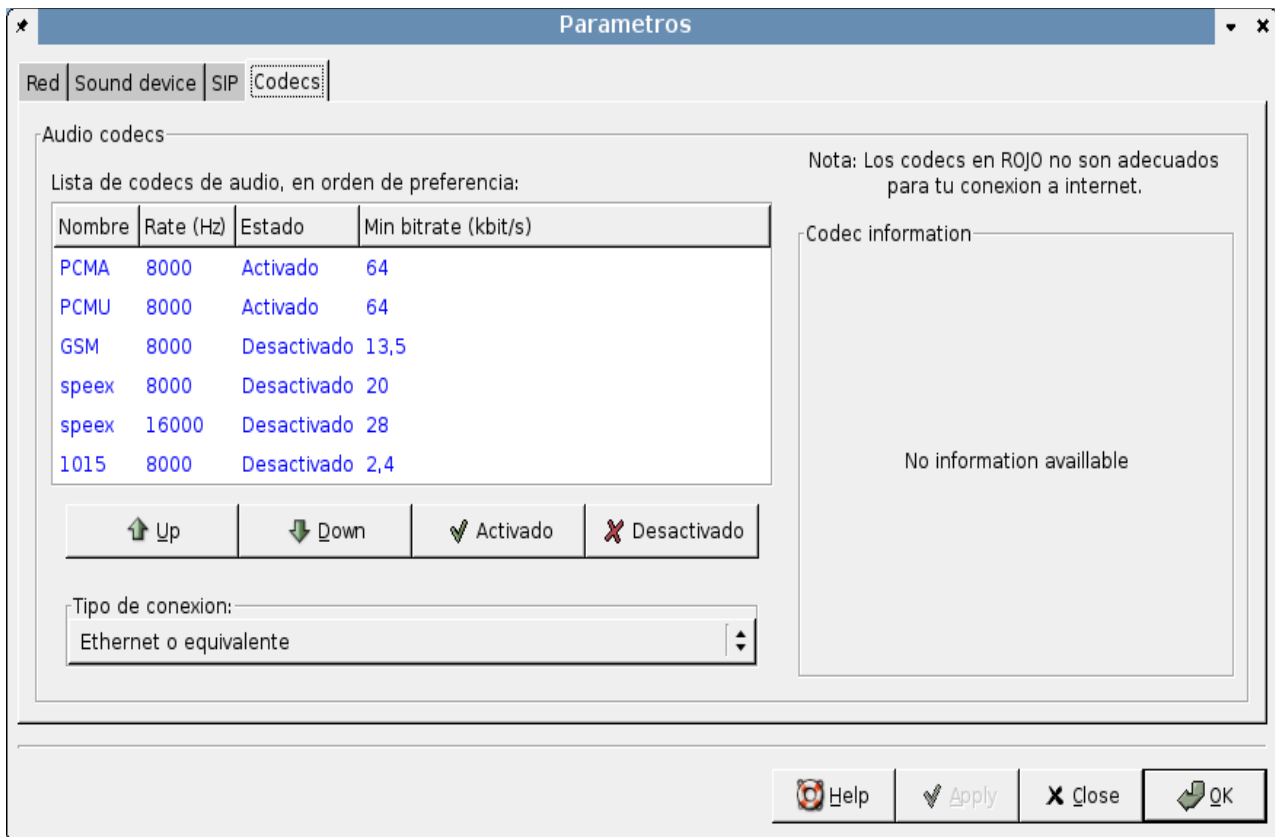


Figura 4.15: Configuración de los codecs utilizados y su orden en Linphone



Una vez que hemos configurado adecuadamente Linphone pulsamos el botón *OK* y nos devuelve a la pantalla principal, desde la que podemos efectuar y recibir llamadas usando la barra de dirección. Debajo de la opción *Mostrar más* nos encontramos cuatro pestañas. La primera de ellas, *Sonido*, nos permite ajustar el volumen de reproducción y grabación, mientras que en la segunda, *Estado*, podemos especificar nuestra disponibilidad, así como proponer servicios alternativos (una página *web*, por ejemplo).

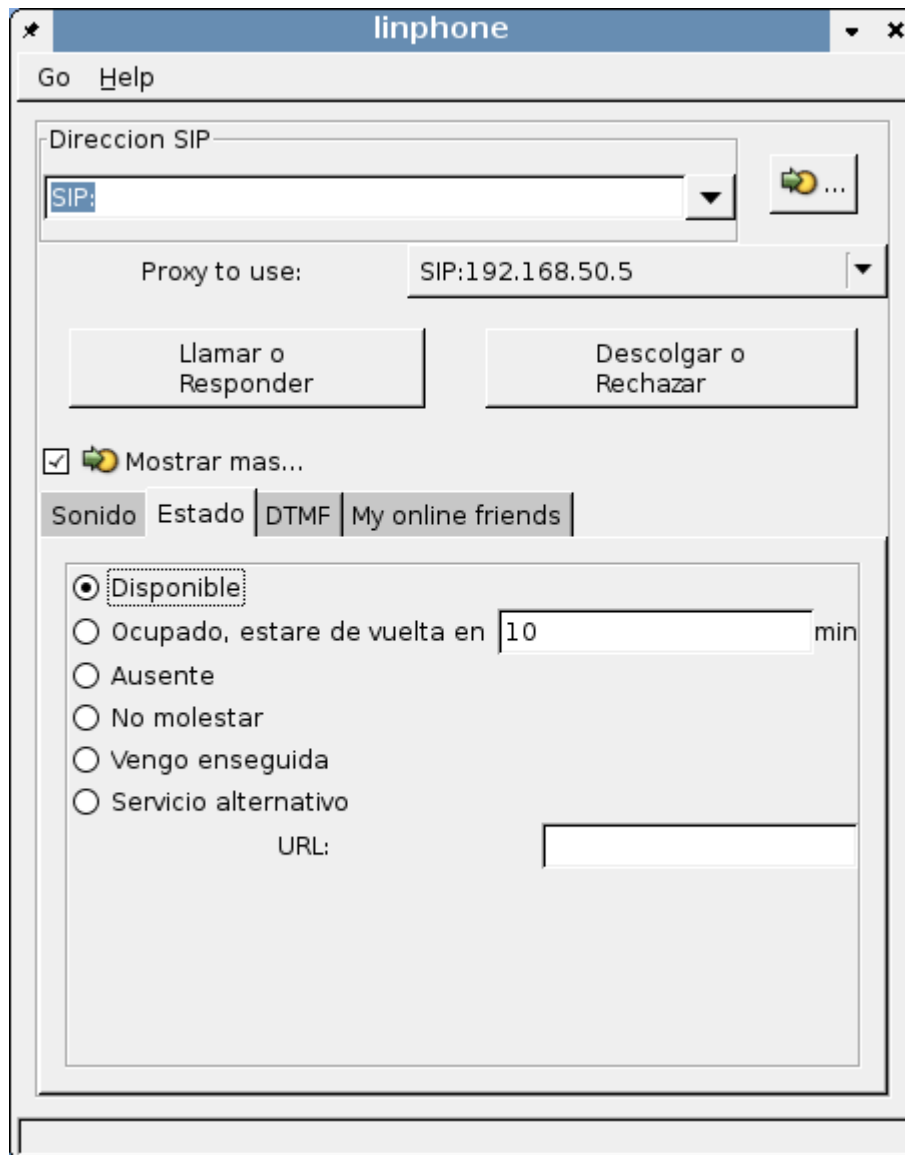


Figura 4.16: Pestaña de estado de Linphone

La siguiente pestaña de opciones avanzadas de la ventana principal de Linphone proporciona un teclado alfanumérico como el de un teléfono convencional para poder enviar tonos DTMF, usados para la configuración de determinados servicios telefónicos o como simples consultas a un servidor.

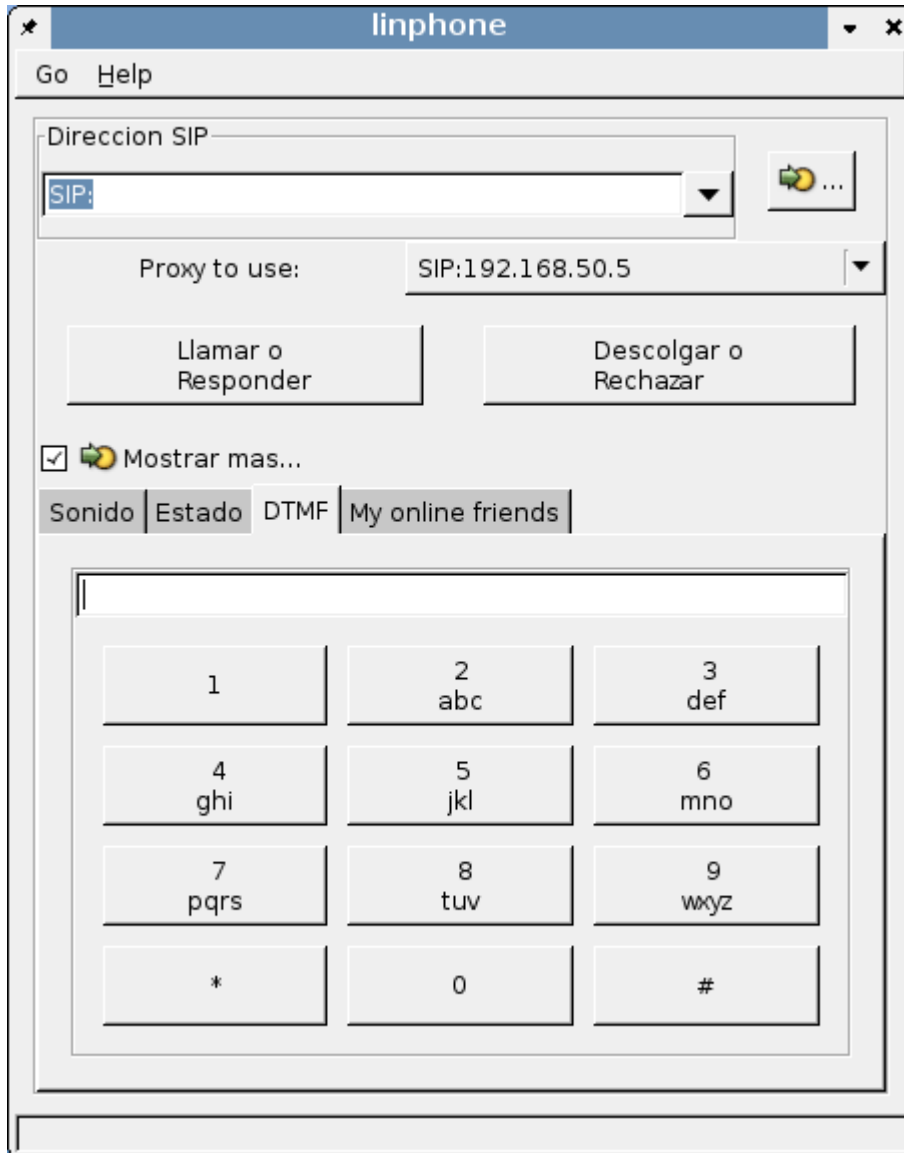


Figura 4.17: Pestaña para enviar tonos DTMF con Linphone

La última pestaña (*My Online Friends*) nos muestra la agenda de nuestro agente de usuario. Esta agenda la podemos modificar en *Go*, *Address Book*, añadiendo o eliminando usuarios y conocer su estado. Hemos de darnos cuenta de que los contactos que aparecen en la imagen son de llamadas salientes realizadas con la infraestructura montada y comprobada, y es por eso por lo que aparecen los números de teléfono como nombre de usuario en la URI y como *host* aparece la dirección IP del *router*, ya que éste es el encargado de encaminar la llamada hacia la red GSM.

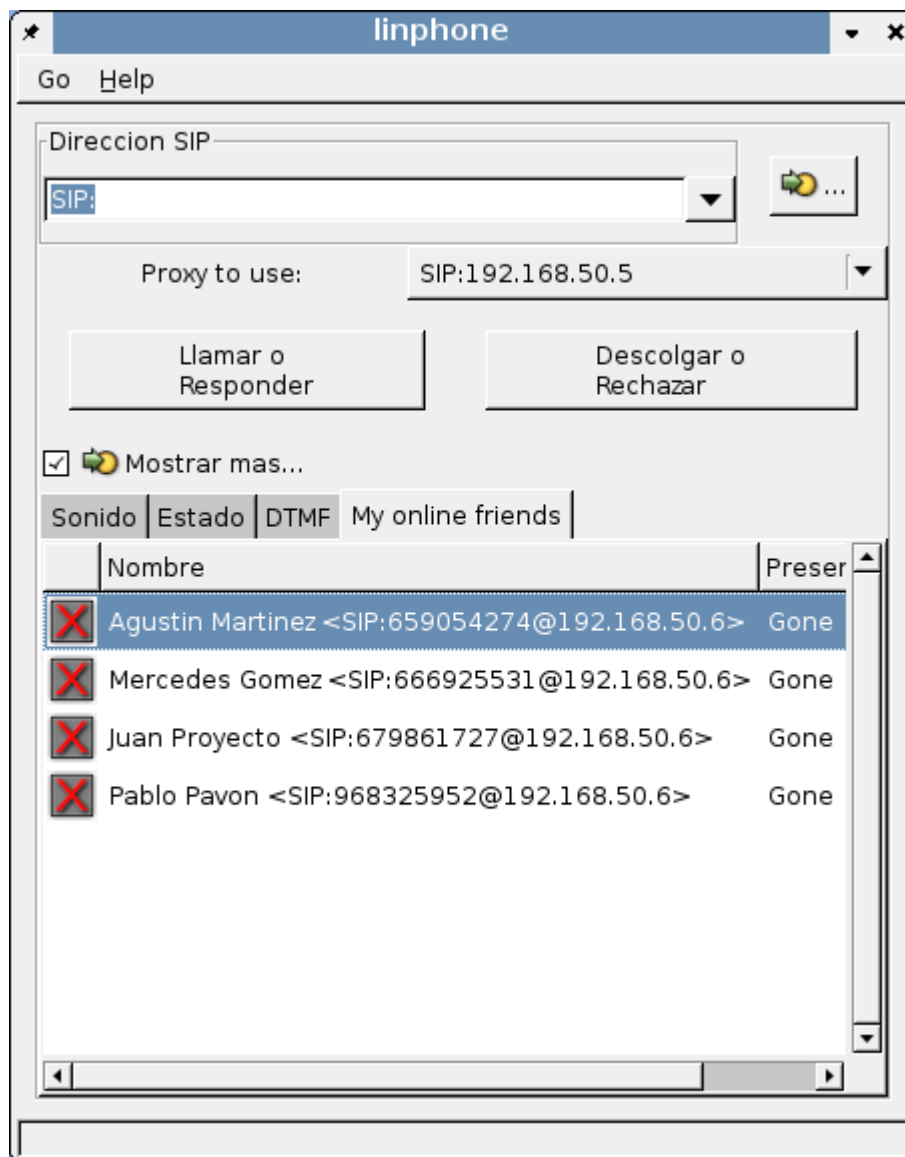


Figura 4.18: Pestaña de contactos de Linphone

Así acaba la configuración y uso de Linphone. Tras las pruebas realizadas con él hemos notado que es un programa fácil de configurar y que cumple notablemente la tarea encomendada con una gran estabilidad.

#### 4.5.3 *El Router Cisco 1761 como agente de usuario. Cambios en los dial-peers SIP.*

Llegado a este punto, vamos a añadir al rol del *router* de pasarela hacia la PSTN, el de agente de usuario SIP. Gracias a esta introducción del *router* en nuestra arquitectura SIP, podremos alcanzar a través de éste a cualquier usuario de la red telefónica pública desde el agente de usuario Linphone del ordenador. Para hacer esto posible, hemos de transformar la configuración existente del *router*, que corresponde al apartado anterior.

### ***Eliminación de las VLANs.***

En este caso, contaremos con todo el rango de direcciones privado del laboratorio, por lo que hemos de quitar la configuración de las VLANs del *router*. Para ello haremos:

```
RP2>enable                !Para entrar al modo privilegiado

RP2#configure terminal    !Para entrar al modo de configuración global

!Para eliminar las VLANs quitamos las subinterfaces
RP2(config)# no interface fastEthernet 0/0.1
RP2(config)# no interface fastEthernet 0/0.2
RP2(config)# no interface fastEthernet 0/0.3
```

### ***Eliminación de los dial-peers voip anteriores.***

Los *dial-peers pots* del ejemplo anterior continúan siendo válidos pues relacionan a los puertos FXS de las dos extensiones y al puerto FXO de la pasarela GSM con sus respectivos números. En cambio, los *dial-peers voip* han de ser cambiados porque ya no se entenderán los binomios “número de extensión-dirección IP”, sino que se incluirá al *router* dentro de la infraestructura como una pasarela con varias extensiones. Así, quitamos de la configuración del *router* los *dial-peers voip* con:

```
RP2(config)#no dial-peer voice 10 voip
RP2(config)#no dial-peer voice 20 voip
RP2(config)#no dial-peer voice 30 voip
```

### ***Establecimiento del router como agente de usuario.***

```
RP2(config)#sip-ua        !Entramos a la configuración de agente de usuario
RP2(config-sip-ua)#no redirection !Sin redirección (no hay NAT)
RP2(config-sip-ua)#retry invite 3 !Número de reintentos de consulta
RP2(config-sip-ua)#retry response 3 !Número de reintentos de
!respuesta
RP2(config-sip-ua)#retry bye 3 !Número de reintentos de
!despedida
RP2(config-sip-ua)#retry cancel 3 !Número de reintentos de
!cancelación de llamada
RP2(config-sip-ua)#retry register 10 !Número de intentos de registro
RP2(config-sip-ua)#timers trying 1000 !Longitud de contadores (ms)

!Dirección del router a registrar en el proxy
RP2(config-sip-ua)#registrar ipv4:192.168.50.6 expires 3600

!Dirección del proxy SIP
RP2(config-sip-ua)#sip-server ipv4:192.168.50.5
```

### ***Creación del conjunto de codecs y su orden de preferencia.***

```
RP2(config)#voice class codec 2 !El 1 está por defecto
RP2(config-voice-codec)#codec preference 1 g711alaw !Primer codec
!preferido
RP2(config-voice-codec)# codec preference 2 g711ulaw !Segundo codec
!preferido
```

### ***Creación de los nuevos dial-peers voip.***

```
RP2(config)#dial-peer voice 10 voip
RP2(config-dial)#destination-pattern 4000
RP2(config-dial)#voice-class codec 2      !Le asignamos este conjunto de
                                           !codecs
RP2(config-dial)#session protocol sipv2   !Usamos SIP v2
RP2(config-dial)#session transport udp    !UDP como protocolo de transporte
```

```
!Dirección del próximo salto SIP (el servidor SIP del PC)
RP2(config-dial)#session target ipv4:192.168.50.6
```

```
!El protocolo para escuchar y enviar tonos DTMF
RP2(config-dial)#dtmf-relay h245-alphanumeric
```

```
!Preferencia para transmitir los paquetes de VoIP
RP2(config-dial)#ip precedence 5
```

!Y así los demás:

```
RP2(config)#dial-peer voice 20 voip
RP2(config-dial)#destination-pattern 5000
RP2(config-dial)#voice-class codec 2
RP2(config-dial)#session protocol sipv2
RP2(config-dial)#session transport udp
RP2(config-dial)#session target ipv4:192.168.50.6
RP2(config-dial)#dtmf-relay h245-alphanumeric
RP2(config-dial)#ip precedence 5
```

```
RP2(config)#dial-peer voice 30 voip
RP2(config-dial)#destination-pattern 6000
RP2(config-dial)#voice-class codec 2
RP2(config-dial)#session protocol sipv2
RP2(config-dial)#session transport udp
RP2(config-dial)#session target ipv4:192.168.50.6
RP2(config-dial)#dtmf-relay h245-alphanumeric
RP2(config-dial)#ip precedence 5
```

Ya tenemos configurada la última parte de la infraestructura SIP, en el siguiente apartado veremos detalladamente el proceso de una llamada y así se terminarán de afianzar los conceptos y configuraciones expuestas a lo largo de este capítulo.

#### 4.5.4 *El Proceso de una Llamada SIP.*

En nuestro ejemplo de SIP, una llamada implica mayor procesamiento que con H.323 pues influyen más equipos. La inclusión de una LAN implica un mayor retardo, pero dada la velocidad de proceso del servidor SER, del *router* Cisco y del conjunto de protocolos SIP, no hay deterioro apreciable en la comunicación.

Vamos a detallar el envío de mensajes entre los diferentes equipos con dos ejemplos de llamada.

El primero es el más sencillo, una extensión de la oficina llama a un ordenador de la LAN, esté o no dentro de la misma sede. Lo veremos usando una herramienta de captura de mensajes en puertos determinados de interfaces de red del PC en el que hemos instalado el servidor SER.

El segundo, por contra, es un caso más complejo, tratándose de una llamada entrante desde la red telefónica pública (ya sea desde la PSTN o desde GSM) hacia un ordenador de la LAN. En este caso usaremos líneas de tiempo para apreciar el orden de las acciones.

##### ***Llamada entre extensión 5000 y PC con número de extensión 6000.***

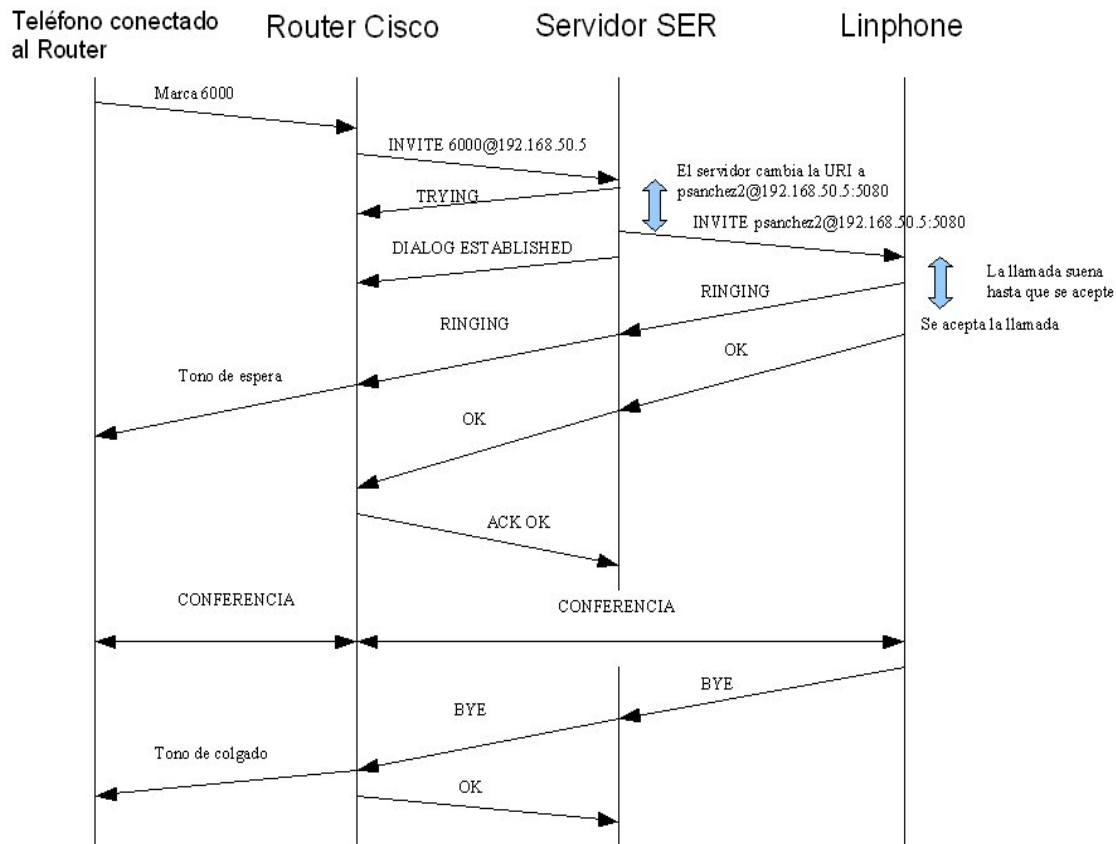
Para capturar los mensajes que se intercambian en la llamada hacemos uso del paquete *ngrep*. Para instalarlo, usaremos *aptitude* de nuevo:

```
$sudo aptitude install ngrep
```

Y lo usaremos en el puerto 5060, que es el puerto por defecto de SIP y en el que opera el servidor SER con:

```
$ sudo ngrep port 5060
```

Esta orden nos mostrará todos los paquetes que entren o salgan por el puerto 5060. Para comprenderlo mejor veremos un diagrama con todos los mensajes y explicaremos éstos detalladamente:



Como vemos, la llamada empieza con el marcado desde una extensión conectada al *router* hacia el *softphone* del PC, que tiene por número de extensión el 6000. Al recibir el número marcado, el *router* compone una petición SIP del tipo INVITE hacia la URI [6000@192.168.50.5](mailto:6000@192.168.50.5), como observamos en la salida de *ngrep*:

```
#
U 192.168.50.6:52938 -> 192.168.50.5:5060
  INVITE sip:6000@192.168.50.5:5060 SIP/2.0..Via: SIP/2.0/UDP 192.168.50.6:5
  060;branch=z9hG4bK3C8..
    From: <sip:5000@192.168.50.6>;tag=35D901-1306..
    To: <sip:6000@192.168.50.5>..
    Date: Fri, 01 Mar 2002 00:58:48 GMT..
    Call-ID:530972AE-2BE611D6-805C9433-F8CABDA8@192.168.50.6..
    Supported: 100rel,timer..
    Min-SE: 1800..
    Cisco-Guid: 1286113092-736498134-2153354291-4174036392..
    User-Agent:Cisco-SIPGateway/IOS-12.x..
    Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, REFER, SUBSCRIBE,
    NOTIFY, INFO, UPDATE, REGISTER..CSeq: 101 INVITE..
```

```
Max-Forwards: 70..
Remote-Party-ID:<sip:5000@192.168.50.6>
    ;party=calling;screen=no;privacy=off..
Timestamp: 1014944328..
Contact: <sip:5000@192.168.50.6:5060>..
Expires: 180..
Allow-Events: telephone-event..
Content-Type: application/sdp..
Content-Length: 227....v=0..
o=CiscoSystemsSIP-GW-UserAgent 7084 4513 IN IP4 192.168.50.6..
s=SIP Call..
c=IN IP4 192.168.50.6..
t=0 0..
m=audio 16878 RTP/AVP 8 0 19..
c=IN IP4 192.168.50.6..
a=rtpmap:8 PCMA/8000..
a=rtp map:0 PCMU/8000..
a=rtpmap:19 CN/8000..
```

En este primer mensaje observamos que el *router* emplea un puerto efímero para enviar el mensaje de tipo INVITE a la URI [6000@192.168.50.5:5060](mailto:6000@192.168.50.5:5060). También vemos en el campo *From:* (desde) la URI que corresponde a la extensión ([5000@192.168.50.6](mailto:5000@192.168.50.6)), en el campo *To:* (para) la URI a la que dirige el mensaje, así como los identificadores de llamada, mensajes permitidos y la configuración de la conexión.

A este mensaje, el servidor SIP le responde con el siguiente:

```
#
U 192.168.50.5:5060 -> 192.168.50.6:5060
SIP/2.0 100 trying -- your call is important to us..
Via: SIP/2.0/UDP 192.168.50.6:5060;branch=z9hG4bK3C8..
From: <sip:5000@192.168.50.6>;tag=35D901-1306..
To: <sip:6000@192.168.50.5>..
Call-ID: 530972AE-2BE611D6-805C9433-F8CABDA8@192.168.50.6..
CSeq: 101 INVITE..
Server: Sip EXpress router (0.9.4 (i386/linux))..
Content-Length: 0..
Warning: 392 192.168.50.5:5060 "Noisy feedback tells:
    pid=7687 req_src_ip=192.168.50.6 req_src_port=52938
    in_uri=sip:6000@192.168.50.5:5060
    out_uri=sip:psanchez2@192.168.50.5:5080
    via_cnt==1"....
```

En el que le informa que está intentando establecer la comunicación con el usuario llamado. Vemos aquí que el servidor distingue entre la URI a la que han llamado o entrante (*in\_uri*) y a la



que el SER transmitirá los mensajes o saliente (*out\_uri*) por si ha habido modificaciones en el *script* de configuración, como es nuestro caso. Así la URI [6000@192.168.50.5:5060](mailto:6000@192.168.50.5:5060) se transforma en la URI [psanchez2@192.168.50.5:5080](mailto:psanchez2@192.168.50.5:5080).

El *proxy* SIP sigue intentando contactar con el agente de usuario llamado, y cuando lo consigue envía al *router* este mensaje:

```
#
U 192.168.50.5:5060 -> 192.168.50.6:5060
SIP/2.0 101 Dialog Establishment..
  Via: SIP/2.0/UDP 192.168.50.6:5060;branch=z9hG4bK3C8..
  Record-Route: <sip:192.168.50.5;ftag=35D901-1306;lr=on>..
  From: <sip:5000@192.168.50.6>;tag=35D901-1306..
  To: <sip:6000@192.168.50.5>;tag=700064580..
  Call-ID: 530972AE-2BE611D6-805C9433-F8CABDA8@192.168.50.6..
  CSeq:101 INVITE..
  Contact: <sip:6000@192.168.50.5:5080>..
  Allow: INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY, MESSAGE,
        INFO..
  Content-Length: 0....
```

En el que le informa de que se ha establecido la conexión (está disponible), tras lo que el servidor SIP envía un mensaje al *router* diciéndole que está enviando la señal de llamada al Linphone:

```
U 192.168.50.5:5060 -> 192.168.50.6:5060
SIP/2.0 180 Ringing..
  Via: SIP/2.0/UDP 192.168.50.6:5060;branch=z9hG4bK3C8..
  Record-Route: <sip:192.168.50.5;ftag=35D901-1306;lr=on>..
  From: <sip:5000@192.168.50.6>;tag=35D901-1306..
  To: <sip:6000@192.168.50.5>;tag=700064580..
  Call-ID: 530972AE-2BE611D6-805C9433-F8CABDA8@192.168.50.6..
  CSeq: 101 INVITE..
  Contact: <sip:6000@192.168.50.5:5080>..
  Allow: INVITE, ACK, OPTIONS, CANCEL,BYE, SUBSCRIBE, NOTIFY, MESSAGE,
        INFO..
  Content-Length: 0....
```

Con lo que el agente de usuario del PC emite el tono de llamada hasta que el usuario acepte la llamada o pase el temporizador. Cuando el usuario acepta, el servidor SIP informa al *router* con el mensaje:

```
#
U 192.168.50.5:5060 -> 192.168.50.6:5060
SIP/2.0 200 OK..
  Via: SIP/2.0/UDP 192.168.50.6:5060;branch=z9hG4bK3C8..
  Record-Route: <sip:192.168.50.5;ftag=35D901-1306;lr=on>..
  From: <sip:5000@192.168.50.6>;tag=35D901-1306..
  To: <sip:6000@192.168.50.5>;tag=700064580..
  Call-ID: 530972AE-2BE611D6-805C9433-F8CABDA8@192.168.50.6..
  CSeq: 101 INVITE..
  Contact: <sip:6000@192.168.50.5:5080>..
  Allow: INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY, MESSAGE,
        INFO..
  Content-Type: application/sdp..
  Content-Length: 170....
  v=0..
  o=psanchez2 123456 654321 IN IP4 192.168.50.5..
  s=A conversation..
  c=IN IP4 192.168.50.5..
  t=0 0..
  m=audio 7079 RTP/AVP 8 0..
  a=rtpmap: 8 PCMA/8000..
  a=rtpmap:0 PCMU/8000..
```

A lo que el *router* responde con la confirmación de establecimiento con un mensaje como éste:

```
#
U 192.168.50.6:52938 -> 192.168.50.5:5060
ACK sip:6000@192.168.50.5:5080 SIP/2.0..
  Via: SIP/2.0/UDP 192.168.50.6:5060;branch=z9hG4bK1704..
  From: <sip:5000@192.168.50.6>;tag=35D901-1306..
  To: <sip:6000@192.168.50.5>;tag=700064580..
  Date: Fri, 01 Mar 2002 00:58:48 GMT..
  Call-ID: 530972AE-2BE611D6-805C9433-F8CABDA8@192.168.50.6..
  Route: <sip:192.168.50.5;ftag=35D901-1306;lr=on>..
  Max-Forwards: 70..
  CSeq: 101 ACK..
  Content-Length: 0....
```

Después de recibir este mensaje de confirmación, el servidor se desentiende y deja conectados a los dos agentes de usuario para la transmisión de los paquetes de voz. De esta manera, no se incrementa la latencia al no tener que pasar todos los paquetes por el servidor SER.

Cuando la llamada termina, se envía un mensaje SIP BYE a través del servidor. En nuestro ejemplo, la llamada la termina el ordenador, por lo que el servidor SIP envía este mensaje al *router*:

```
#
U 192.168.50.5:5060 -> 192.168.50.6:5060
BYE sip:5000@192.168.50.6:5060 SIP/2.0..
Record-Route: <sip:192.168.50.5;ftag=700064580;lr=on>..
Via: SIP/2.0/UDP 192.168.50.5;branch=z9hG4bK3168.9f2349d4.0..
Via: SIP/2.0/UDP 192.168.50.5:5080;rport=5080;branch=z9hG4bK23492271..
From: <sip:6000@192.168.50.5>;tag=700064580..
To: <sip:5000@192.168.50.6>;tag=35D901-1306..
Call-ID: 530972AE-2BE611D6-805C9433-F8CABDA8@192.168.50.6..
CSeq: 102 BYE..
Contact: <sip:6000@192.168.50.5:5080>..
Max-Forwards: 4..
User-Agent:Linphone-1.0.1/eXosip..
Content-Length: 0..
P-hint: rr-enforced....
```

A lo que el *router* responde al servidor con la confirmación de finalización de llamada con un mensaje tipo OK:

```
#
U 192.168.50.6:52938 -> 192.168.50.5:5060
SIP/2.0 200 OK..
Via: SIP/2.0/UDP 192.168.50.5;branch=z9hG4bK3168.9f2349d4.0,
SIP/2.0/UDP 192.168.50.5:5080;rport=5080;branch=z9hG4bK23492271..
From: <sip:6000@192.168.50.5>;tag=700064580..
To: <sip:5000@192.168.50.6>;tag=35D901-1306..
Date: Fri, 01 Mar 2002 00:59:00 GMT..
Call-ID: 530972AE-2BE611D6-805C9433-F8CABDA8@192.168.50.6..
Server: Cisco-SIPGateway/IOS-12.x..
Content-Lengt: 0..
CSeq: 102 BYE....
```

Con la recepción por parte del servidor de este mensaje acaba todo el proceso que implica una llamada con SIP de una extensión telefónica a un agente de usuario *software*.

***Llamada desde un usuario de la red telefónica pública a un ordenador de la LAN con número de extensión 6000.***

La llamada sigue los siguientes pasos:

1. Un usuario de la PSTN llama al número de teléfono de la tarjeta SIM (647056201)
2. La señal de llamada llega al *gateway* GSM, que descuelga automáticamente.
3. La llamada pasa al *router*, que observa en el *dial-peer pots 4* que le faltan 4 dígitos por marcar (el número de extensión), por lo que emite un tono de marcado.
4. El tono de marcado llega al llamante y éste pulsa los 4 dígitos requeridos (marca 6000 para llamar al ordenador).
5. El *router* recibe las 4 cifras y envía una petición INVITE al servidor SER con la URI de formato *número\_extensión@dirección\_puesta\_en\_dial-peer* es decir :
  - 4000 -> [4000@192.168.50.6](mailto:4000@192.168.50.6)
  - 5000 -> [5000@192.168.50.6](mailto:5000@192.168.50.6)
  - 6000 -> [6000@192.168.50.5](mailto:6000@192.168.50.5)
6. El SER transmite la invitación siguiendo la configuración del *script*, luego envía la invitación del *router* a [psanchez2@192.168.50.6:5080](mailto:psanchez2@192.168.50.6:5080). Si el número de la extensión llamada fuera de una conectada directamente al *router*, se reenvía a éste el mensaje INVITE.
7. El agente de usuario descuelga o acepta la llamada, con lo que queda establecido el flujo de comunicación entre la extensión llamante y el PC.
8. Cuando uno de los dos interlocutores cuelga, se informa al *proxy* SIP y éste se lo hace saber al otro agente de usuario, dando así por terminada la conferencia.

Y, por último, lo veremos todo resumido en el siguiente diagrama:

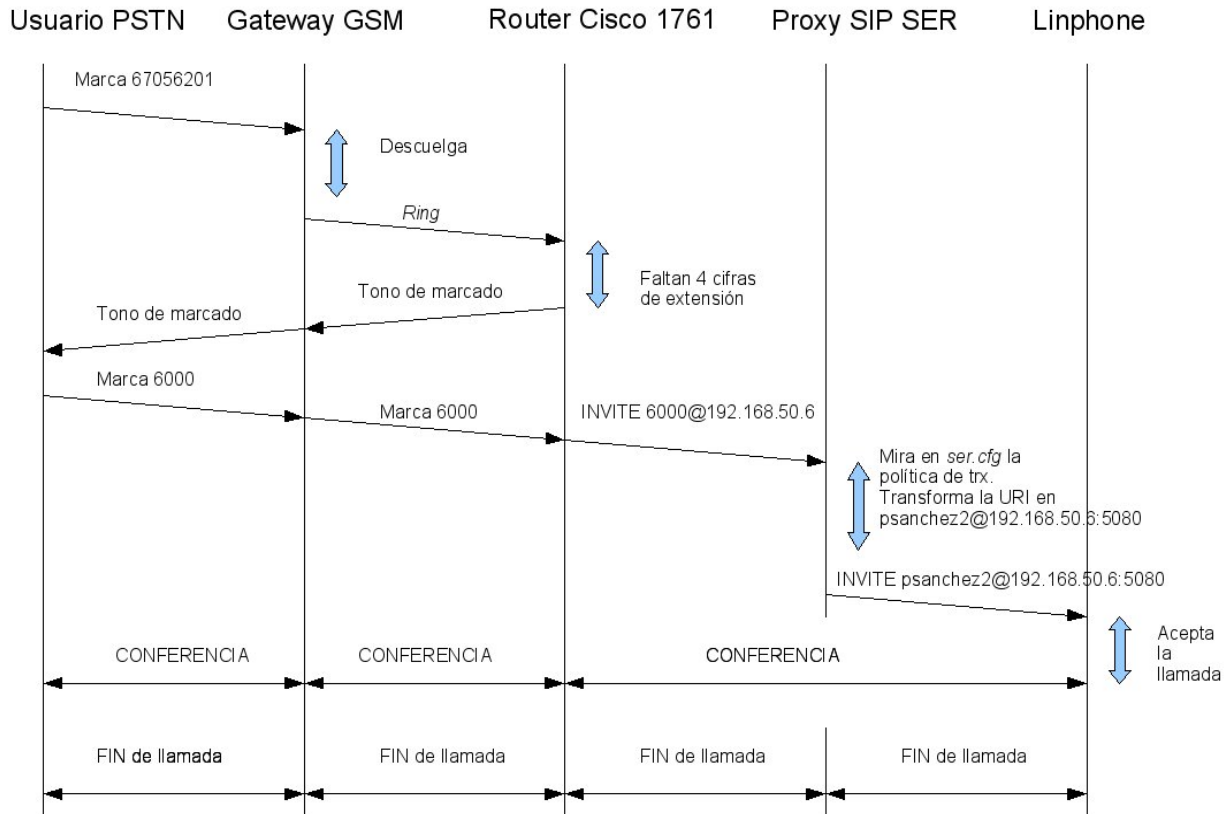


Figura 4.19: Envío de mensajes de una llamada SIP entre todos los componentes de la arquitectura

## **5 Análisis de Viabilidad del Proyecto de Inversión.**

### **5.1 Descripción del Proyecto.**

Se pretende la adquisición e instalación de diverso equipamiento, que interactuará para conseguir una reducción del considerable coste telefónico que afrontará la empresa una vez alcanzada la cuota de mercado prevista. En concreto, se estimará la cantidad de llamadas entre los diferentes integrantes del modelo de negocio, el tiempo y coste de llamada, así como las tarifas “fijas” que tendrá que afrontar la compañía. Con esto, se intentará dar la visión más real posible de la repercusión que tendrá la instalación de los equipos y su configuración.

### **5.2 Datos Económicos del Estudio.**

#### **5.2.1 Datos Utilizados en la Proyección de los Estados Contables.**

Se parte de los siguientes supuestos:

- Se estima, para simplificar los cálculos un periodo medio de cobro y pago al contado.
- Suponemos devoluciones de IVA al año siguiente al de su devengo, así como pagos también al siguiente año al de su devengo.
- Se han supuesto pagos de IRPF al año siguiente al de su devengo, tomando el tipo máximos de las tablas del impuesto.
- El interés de los préstamos se ha calculado como el índice Euribor a 6 meses más un porcentaje de variabilidad (en el que se computan las posible subidas del Euribor y otros imprevistos) para hacer el análisis en la peor situación posible.

#### **5.2.2 Financiación del Proyecto.**

Aunque en el momento del alcance de la cuota de mercado y actividad esperada del sector inmobiliario del levante español la empresa cuente con fondos propios suficientes, impondremos la situación de afrontar un préstamo para sufragar la inversión de los equipos y su montaje.

En concreto, iremos presentando a lo largo del análisis de viabilidad dos soluciones para la migración, una total, que incluye las pasarelas desde y hacia las redes telefónicas públicas de los tres países en los que se prevé se instalen sendas oficinas, y otra con sólo la infraestructura necesaria para la eliminación de costes de llamada entre delegaciones.

Para la migración total:

<b>COSTE</b>	<b>IMPORTE</b>
3 Routers Cisco 1761	6.123,00 €
3 Tarjetas FXS	949,99 €
3 Tarjetas FXO	949,99 €
3 Gateway GSM	651,45 €
1 Servidor	735,74 €
Beca PFC	1.500,00 €
<b>TOTAL INVERSIÓN</b>	<b>10.910,17 €</b>

Tabla 2: Inversión de la migración total

PRESTAMO	10.910,17 €
FONDOS PROPIOS	-
<b>TOTAL FINANCIACION</b>	<b>10.910,17 €</b>

Tabla 3: Financiación de la migración total

Y para la parcial:

<b>COSTE</b>	<b>IMPORTE</b>
1 Servidor	735,74 €
Beca PFC	1.500,00 €
<b>TOTAL INVERSIÓN</b>	<b>2.235,74 €</b>

Tabla 4: Inversión de la migración parcial

PRESTAMO	2.235,74 €
FONDOS PROPIOS	-
<b>TOTAL FINANCIACION</b>	<b>2.235,74 €</b>

Tabla 5: Financiación de la migración parcial

Como se observa, la diferencia entre las dos migraciones es considerable. Con la primera obtenemos plena funcionalidad, pero la parcial es aconsejable para el estado actual de la empresa (realizando las gestiones para la apertura de la primera oficina fuera de España, concretamente en el Reino Unido).

En ambos casos, el préstamo para la financiación externa tendrá las siguientes características:

<b>Importe Financiado:</b>	10.910,17 €
<b>Plazo (años):</b>	2
<b>Interes [(Euribor a seis meses 2,833%)+2,167] :</b>	5,00%

Tabla 6: Préstamo para la financiación de la migración total

<b>Importe Financiado:</b>	2.235,74 €
<b>Plazo (años):</b>	0,5
<b>Interes [(Euribor a seis meses 2,833%)+2,167]:</b>	5,00%

Tabla 7: Préstamo para la financiación de la migración parcial

Las garantías de los préstamos se materializan con los inmuebles adquiridos y el propio capital social de la compañía.

Como se observa en el siguiente cuadro, la amortización del préstamo de la amortización total se calcula para un período de 2 años (lo más usual para esa cantidad), resultando una cuota mensual que se descontará del ahorro conseguido. Ya que la cuota es menor que el ahorro, como veremos más adelante, la empresa no tiene que reajustar su presupuesto:

<b>Importe</b>	10.910,17 €	<b>Cuota:</b>	5.867,54 €
<b>Interés</b>	5,00%		
<b>Plazo (años)</b>	2	<b>Cuota mensual:</b>	488,96 €

Año	Cuota	Capital	Interés	Capital Pte	Capital Amort	Gasto Neto
2007	5.867,54 €	5.322,04 €	545,51 €	5.588,14 €	5.322,04 €	545,51 €
2008	5.867,54 €	5.588,14 €	279,41 €	0,00 €	10.910,17 €	279,41 €
<b>TOTALES</b>	<b>11.735,09 €</b>	<b>10.910,17 €</b>	<b>824,92 €</b>			<b>824,92 €</b>

Tabla 8: Amortización del préstamo de la migración total

Y el de la migración parcial, para un crédito a 6 meses:

<b>Importe</b>	2.235,74 €	<b>Cuota:</b>	4.638,48 €
<b>Interés</b>	5,00%		
<b>Plazo (años)</b>	0,5	<b>Cuota mensual:</b>	386,54 €

Año	Cuota	Capital	Interés	Capital Pte	Capital Amort	Gasto Neto
Mes 1	386,54 €	377,22 €	9,32 €	1.858,52 €	377,22 €	9,32 €
Mes 2	386,54 €	377,22 €	9,32 €	1.481,29 €	754,45 €	9,32 €
Mes 3	386,54 €	377,22 €	9,32 €	1.104,07 €	1.131,67 €	9,32 €
Mes 4	386,54 €	377,22 €	9,32 €	726,84 €	1.508,90 €	9,32 €
Mes 5	386,54 €	377,22 €	9,32 €	349,62 €	1.886,12 €	9,32 €
Mes 6	358,93 €	349,61 €	9,32 €	0,00 €	2.235,74 €	9,32 €
<b>TOTALES</b>	<b>2.291,63 €</b>	<b>2.235,74 €</b>	<b>55,89 €</b>			<b>55,89 €</b>

Tabla 9: Amortización del préstamo de la migración parcial



### 5.2.3 Amortización y Tarifas de las Distintas Migraciones.

En el estudio de los períodos de amortización se ha partido de los siguientes supuestos para los cálculos:

- Se utilizará como coste de establecimiento de llamada el más barato de la oferta de tarifas internacionales de operadoras de telefonía móvil de España, correspondiente a Vodafone con un coste de 0,30 € .
- Se despreciará en llamadas locales o provinciales dentro del mismo país el establecimiento de llamada.
- El precio del minuto se fija a una media de 0,36 €, también la más barata de la oferta de tarifas internacionales, correspondiente a Vodafone.
- El precio del minuto de la llamada local o provincial dentro del mismo país se adecúa a una media de 0,02 € .
- Se aplica una tarifa plana de empresas que permite las llamadas sin coste entre móviles de la misma cuenta empresarial por un precio medio de 7 € al mes por línea.
- El coste que supone la manutención de una dirección IP fija para las conexiones a *Internet* de las oficinas, lo fijaremos al precio español de 12 € por conexión.

Así, el coste del tráfico telefónico sin la aplicación de este PFC quedaría descrito en la siguiente tabla, con los tiempos promedio de duración de los distintos tipos de llamada:

## **COSTES TOTALES SIN APLICAR PFC**

### **PROMEDIO LLAMADAS**

ORIGEN Y DESTINO	LLAMADAS DIA	LLAMADAS MES	MEDIA MINUTOS
Comercial -> Cliente	3	66	6
Entre Oficinas	15	330	3
Oficinas -> Cliente (Sólo Locales)	20	440	5
Comercial -> Oficina mismo País	5,33	117,26	6
Comercial -> Oficina otro País	10,66	234,52	6

TOTAL MINUTOS MES	COSTE ESTABLECIMIENTO	COSTE MINUTO	COSTE TOTAL
396	0,30 €	0,36 €	162,36 €
990	0,30 €	0,36 €	455,40 €
2200	0,00 €	0,02 €	44,00 €
703,56	0,00 €	0,00 €	0,00 €
1407,12	0,30 €	0,36 €	576,92 €

### **OTROS COSTES**

COSTES	CANTIDAD	IMPORTE
Tarifa Plana Empresas por Línea	5	35,00 €
<b>TOTAL OTROS COSTES</b>		<b>35,00 €</b>

**TOTAL COSTES SIN APLICAR PFC** 1.273,68 €

Tabla 10: Resumen de costes sin aplicar el PFC

Con la migración parcial a VoIP que, recordamos, sólo incluye las llamadas entre oficinas, sin pasarela de ningún tipo a las redes telefónicas públicas, el estudio del coste telefónico de SunConexion quedaría:

## **COSTES TOTALES MIGRACION PARCIAL**

### **PROMEDIO LLAMADAS**

<b>ORIGEN Y DESTINO</b>	<b>LLAMADAS DIA</b>	<b>LLAMADAS MES</b>	<b>MEDIA MINUTOS</b>
Comercial -> Cliente	3	66	6
Entre Oficinas	15	330	3
Oficinas -> Cliente (Sólo mismo País)	20	440	5
Comercial -> Oficina mismo País	5,33	117,26	6
Comercial -> Oficina otro País	10,66	234,52	6

<b>TOTAL MINUTOS MES</b>	<b>COSTE ESTABLECIMIENTO</b>	<b>COSTE MINUTO</b>	<b>COSTE TOTAL</b>
396	0,30 €	0,36 €	162,36 €
990	0,00 €	0,00 €	0,00 €
2200	0,00 €	0,02 €	44,00 €
703,56	0,00 €	0,00 €	0,00 €
1407,12	0,30 €	0,36 €	576,92 €

Tarifa Plana Empresas por Línea	5	35,00 €
Coste IP Fija	3	36,00 €
<b>TOTAL OTROS COSTES</b>		<b>71,00 €</b>

**TOTAL COSTES MIGRACIÓN PARCIAL 854,28 €**

*Tabla 11: Resumen de costes aplicando migración parcial*

Y, por último, la migración total, que corresponde a la inclusión en la infraestructura de las llamadas entre oficinas, las llamadas de y hacia los comerciales móviles y las de y hacia clientes extranjeros:

## **COSTES TOTALES MIGRACION TOTAL**

### **PROMEDIO LLAMADAS**

<b>ORIGEN Y DESTINO</b>	<b>LLAMADAS DIA</b>	<b>LLAMADAS MES</b>	<b>MEDIA MINUTOS</b>
Comercial -> Cliente	3	66	6
Entre Oficinas	15	330	3
Oficinas -> Cliente (Cualquier Oficina)	20	440	5
Comercial -> Oficina mismo País	5,33	117,26	6
Comercial -> Oficina otro País	10,66	234,52	6

<b>TOTAL MINUTOS MES</b>	<b>COSTE ESTABLECIMIENTO</b>	<b>COSTE MINUTO</b>	<b>COSTE TOTAL</b>
396	0,00 €	0,02 €	7,92 €
990	0,00 €	0,00 €	0,00 €
2200	0,00 €	0,02 €	44,00 €
703,56	0,00 €	0,00 €	0,00 €
1407,12	0,00 €	0,00 €	0,00 €

<b>COSTES</b>	<b>CANTIDAD</b>	<b>IMPORTE</b>
Tarifa Plana Empresas por Línea	5	35,00 €
Coste IP Fija	3	36,00 €
<b>TOTAL</b>		<b>71,00 €</b>

**TOTAL COSTES MIGRACIÓN TOTAL** 122,92 €

*Tabla 12: Resumen de costes de la migración total*

Como se puede ver claramente en los resúmenes de coste anteriores, la migración total a VoIP es la que consigue reducir a un valor menor al 10% el coste telefónico que se prevé para la empresa, si bien, esta opción es también la que una mayor inversión requiere.

Relacionamos coste e inversión en el siguiente gráfico:

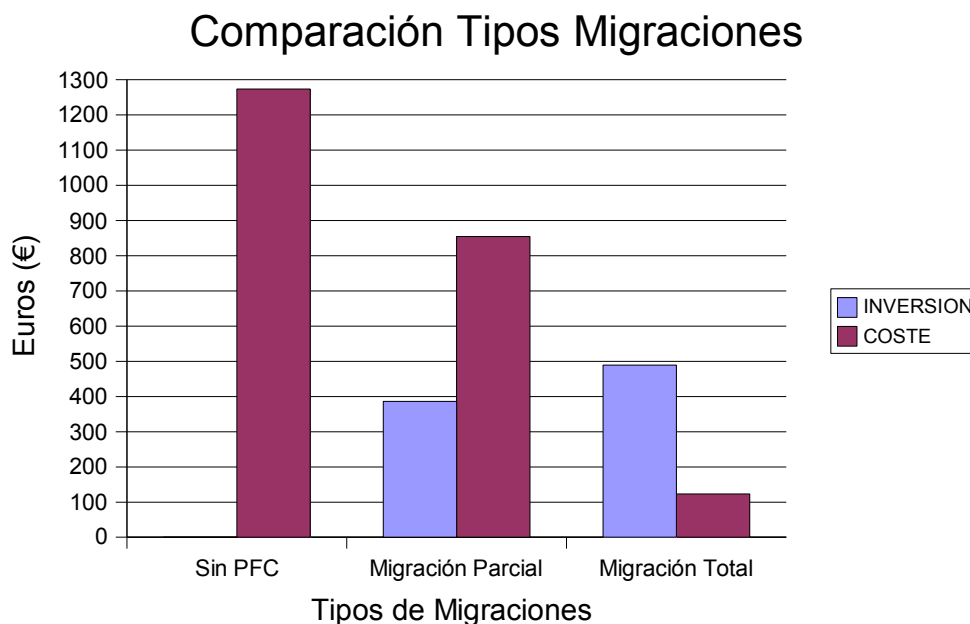


Figura 5.1: Comparación Coste-Inversión de los tipos de migración

También es necesario reseñar que es distinta la situación mientras la empresa está pagando el préstamo de la inversión y cuando ya ha finalizado ese período. En lo sucesivo, entenderemos como el 100% el coste telefónico total de la opción sin PFC, es decir, 1273,68 € .

En los siguientes diagramas comprenderemos esta diferencia de situación, siendo capaces de observar cómo el ahorro de costes es muy considerable en la opción de migración total y nada despreciable en la de migración parcial.

Comenzamos con el coste total sin ninguna infraestructura encaminada a la migración:

### Coste Telefónico Sin Aplicar PFC

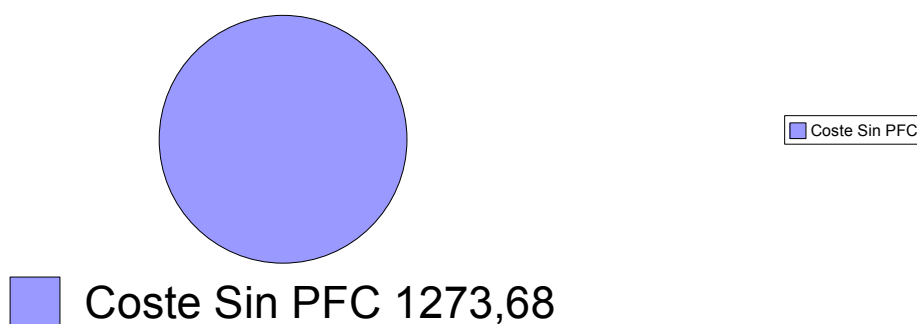


Figura 5.2: Gráfico del coste telefónico sin migración

La comparación entre la migración total y la parcial durante el tiempo que la empresa paga el préstamo (2 años para la total y 6 meses para la parcial):

### Coste Migración Total Durante Pago Préstamo

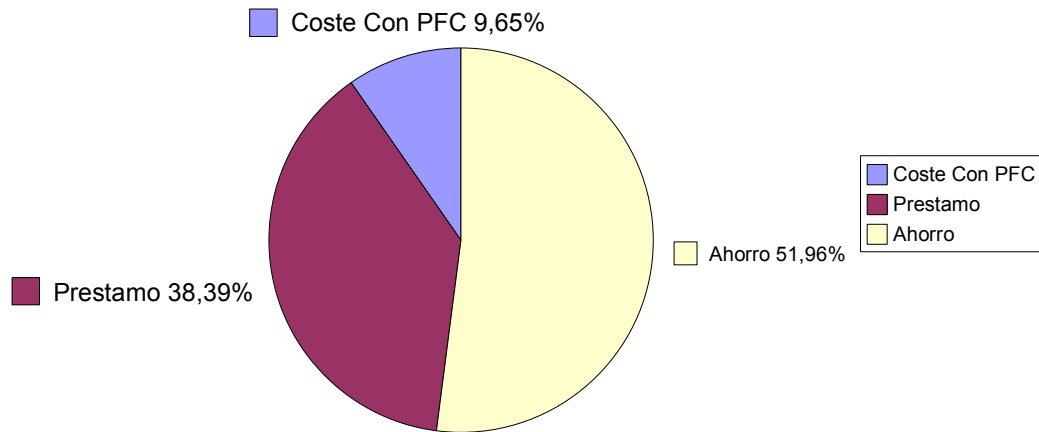


Figura 5.3: Comparación entre pagos por préstamo de inversión, coste telefónico y ahorro con la migración total

### Coste Migración Parcial Durante Pago Préstamo

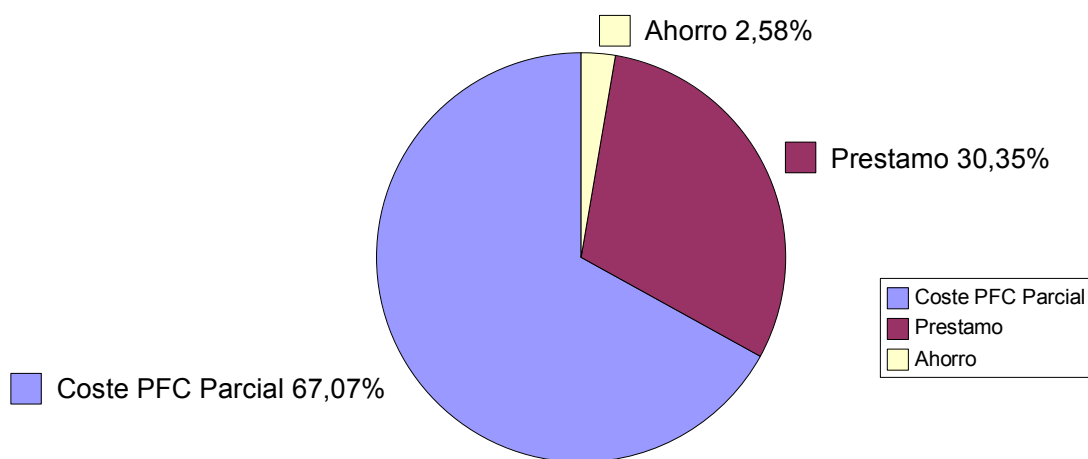


Figura 5.4: Comparación entre pagos por préstamo de inversión, coste telefónico y ahorro con la migración parcial

Y es tras pagar el préstamo cuando se aprecia en su totalidad el ahorro conseguido con el empleo de las nuevas tecnologías:

## Coste Migración Total Tras Pagar Préstamo

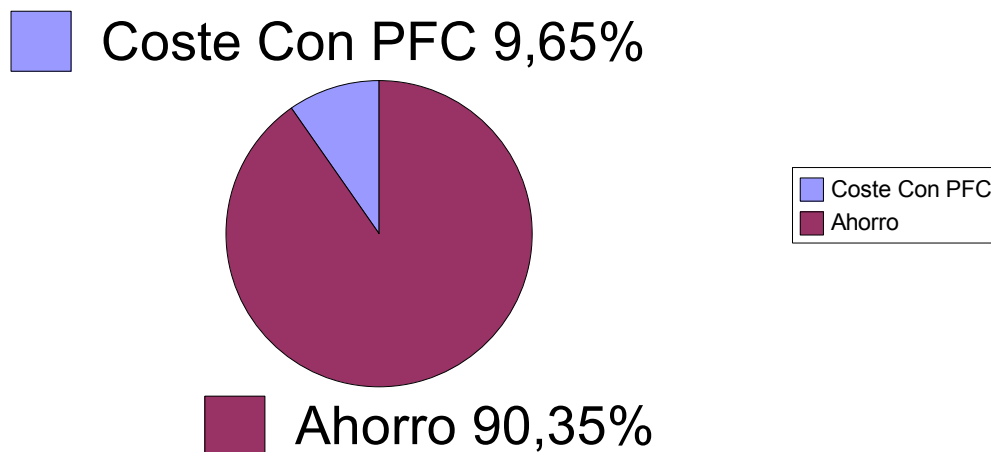


Figura 5.5: Comparación entre ahorro y coste telefónico con migración total

## Coste Migración Parcial Tras Pago Préstamo

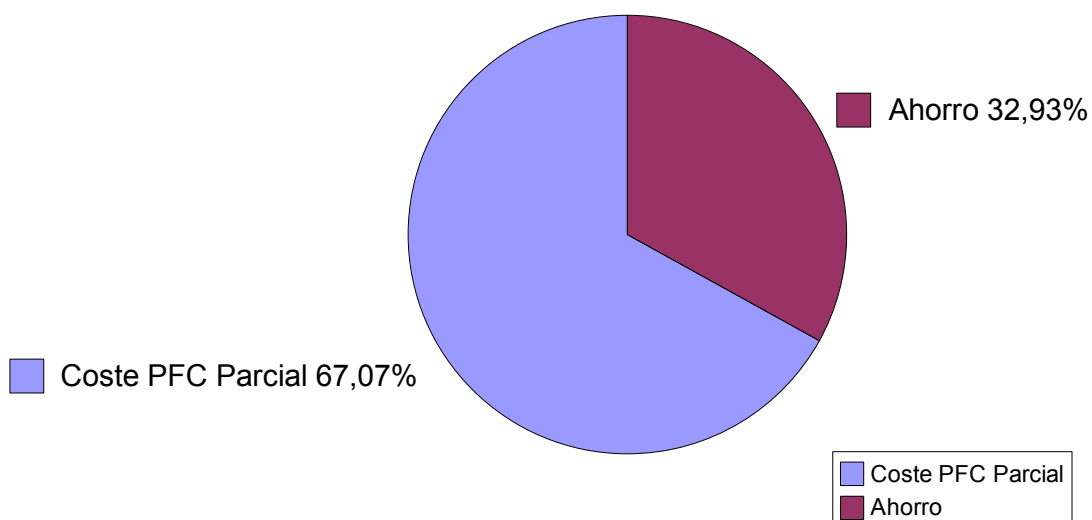


Figura 5.6: Comparación entre ahorro y coste telefónico con migración parcial

### 5.3 Conclusiones del estudio.

Como hemos visto en el análisis, la inclusión de las nuevas tecnologías descritas en capítulos anteriores posibilita que empresas que no se dedican al sector tecnológico disfruten de reducción de costes que llegan en casos a más del 90%. En el estado actual de la empresa, se recomienda la instalación de la infraestructura de la migración parcial, y cuando se alcance la cuota de mercado y la implantación deseadas en el mercado inmobiliario del turismo residencial europeo, se termine la migración incluyendo a los comerciales y las pasarelas a las redes telefónicas públicas.

También hay que plantear la posibilidad de no pedir un préstamo que sufrague la inversión a realizar. Si la empresa cuenta con inmovilizado suficiente para hacer frente a los gastos, el período de amortización de las migraciones, dedicando todo el ahorro generado a la devolución a las arcas de la empresa, queda:

	<b>COSTE DE LA INVERSIÓN</b>	<b>AHORRO MES</b>	<b>PLAZO DE AMORTIZACIÓN (MESES)</b>
Migración Parcial	2.235,74 €	419,40 €	5,33
Migración Total	10.910,17 €	1.150,76 €	9,48

Tabla 13: Tiempos de amortización con financiación propia

## 6 Conclusiones y Líneas Futuras.

A lo largo de todo el documento hemos ido comprobando la rentabilidad del proyecto, así como su pequeña curva de aprendizaje una vez configurado adecuadamente. Esto hace que las alternativas aquí presentadas para la comunicación telefónica de una empresa como SunConexion, que prevé un crecimiento que incluye la apertura de varias oficinas en Europa, sean aplicables a cualquier empresa que cuente con varias delegaciones. Si bien es cierto que si todas las delegaciones se encuentran en el mismo país, la amortización es más lenta pero igual de interesante.

En el proyecto no se ha usado ninguna tecnología especialmente novedosa o puntera, sino que todas cuentan con varios años de existencia, por lo que es previsible, vista la madurez que han demostrado, su implantación progresiva en todos los sectores empresariales así como en las Administraciones de varios países. Esto es especialmente interesante para países en vías de desarrollo, que han de ahorrar costes en la medida de lo posible, dado el bajo PIB (Producto Interior Bruto) del que disfrutan.

Las operadoras de datos o los proveedores de servicios comienzan a mostrar intenciones de regular el tráfico de VoIP, viendo el mercado que se les escapa. Esto sólo afectará a las compañías que se dedican a ofrecer VoIP con coste, es decir, que emplean tarificación para las llamadas a precios muy por debajo de los de las operadoras tradicionales. El entorno empresarial para el que va dedicado este PFC no se verá afectado por esa posible regulación y tarificación adicional del tráfico VoIP, ya que en estos proyectos la baja latencia y alta calidad de la línea contratada de acceso a *Internet* han de ser requisitos esenciales que exigir al proveedor de servicios. Por esto, en medida de lo posible, se instalarán conexiones de banda ancha, prestando especial atención a la velocidad de subida, ya que los proveedores de servicios basan sus ofertas en gran capacidad de bajada mientras que para el tráfico de voz tan importante es recibir como enviar paquetes. Con unos 300 kb de velocidad de subida y bajada podemos asegurar proyectos de VoIP de mediana envergadura.

Mención aparte merece el cifrado de los paquetes de voz para garantizar el anonimato y privacidad de las llamadas, ya sea mediante túneles IP creados por la propia empresa (evitando también la tarificación adicional del tráfico de voz) o creados por el proveedor de servicios. Los túneles y el cifrado SSL garantizan totalmente la seguridad de las comunicaciones de la compañía, si bien introducen una pequeña cantidad de latencia que desaconsejará su instalación en conexiones a poca velocidad.

Para completar las infraestructuras descritas en el documento, se puede, quizás como objeto de un futuro Proyecto Fin de Carrera, incluir las centralitas, túneles IP y otras funciones avanzadas de SIP.

Las centralitas no se han tratado en este PFC porque sólo aportan la posibilidad de incluir un mayor número de extensiones por oficina. Estos aparatos están muy presentes en las oficinas de todo el mundo, existiendo analógicas (que sólo disponen de encaminamiento de llamadas salientes, por lo que no son muy interesantes como complemento a la VoIP) y digitales (que sí permiten encaminamiento de llamadas entrantes y salientes, si bien su configuración es más complicada).

Sería también interesante tratar los túneles IP por la seguridad que proporcionan a las comunicaciones entre la empresa, ya sea para la transmisión de datos o voz.

En cuanto a las funciones avanzadas del protocolo SIP, podemos destacar la integración de la URI de cada usuario con su *e-mail*, permitiendo mandarle un mensaje de correo desde el mismo agente de usuario si no está disponible, o las funciones de *clic-to-dial* en la página *web* de la empresa para que los usuarios cuenten con una vía de comunicación gratuita y accesible.



## **7 Bibliografía y Referencias.**

### **7.1 Bibliografía.**

1. Jonathan Davidson y James Peters, Voice Over IP Fundamentals, Cisco Press, 2000
2. Paul J. Fong y otros, Configuring Cisco Voice Over IP, Syngress Publishing, 2002

### **7.2 Referencias.**

1. <http://www.recursosvoip.com>
2. <http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/vcl.htm>
3. <http://www.iptel.org>
4. <http://www.linphone.org>
5. <http://www.columbia.edu/kermit/ckermi.html>
6. <http://www.ietf.org/rfc.html>
7. <http://www.rfc-es.org>
8. <http://www.guia-ubuntu.org>
9. <http://www.dit.upm.es>
10. <http://www.pcbox.es>