2. Fundamento teórico sobre el estándar VoIP

2.1. Introducción

Para entender mejor el alcance de la tecnología IP, es necesario tener un mínimo conocimiento teórico. Se va a tratar de explicar de forma sencilla y sin entrar en demasiados detalles en qué consiste el mundo de la VoIP, porqué surge y sus posibilidades.
En primer lugar hay que tener en cuenta la naturaleza de la señal que se quiere transmitir. La voz humana es una señal analógica que, por tanto, debe ser digitalizada y codificada para poder tratarla como una señal puramente digital. Después hay que empaquetar la información de la señal digital de voz para poder encaminarla por las redes de conmutación de paquetes de la tecnología IP. Estos procesos han ido mejorando y, junto con el aumento del ancho de banda de transmisión de los canales de comunicación, han provocado que la VoIP sea una tecnología útil.

2.2. Voz sobre IP

Voice over Internet Protocol (VoIP). Su objetivo es dividir en paquetes los flujos de audio para transportarlos sobre redes basadas en IP. Para comenzar a hablar de la VoIP, primero se tiene que tener en cuenta que la voz, señal analógica, debe ser digitalizada y codificada para poderla empaquetar y transmitir por las redes IP. Siendo así, se hablará en este apartado de los procesos de digitalización y codificación más comunes.
VoIP es un estándar desarrollado para establecer comunicaciones de voz en tiempo real a través de redes IP. Estas redes se diseñaron para el transporte de datos y están muy difundidas a lo largo de todo el mundo. La transmisión en las redes IP se hace mediante conmutación de paquetes que han ido evolucionando para adaptarse al tráfico de voz hasta realizar las funciones de una red tradicional de conmutación de circuitos. Lo que se espera es que desaparezcan la mayoría de las redes de transmisión de voz que existen en la actualidad y sean reemplazadas por redes de conmutación de paquetes basadas en VoIP. Una de las principales ventajas de la conmutación de paquetes es que pueden soportar un mayor volumen de tráfico gracias a que aprovechan mejor el ancho de banda disponible. Esto supone, además de una reducción clara y amplia de los costes, un desarrollo de una serie de servicios nuevos para voz y datos complicados de implementar en las redes de circuitos conmutados.
Para conseguir que el tráfico de voz y datos coexistan en las redes de telecomunicaciones actuales, se desarrolla el estándar VoIP. Este modelo permite empaquetar la voz para que pueda ser transmitida junto con los datos por las redes IP (Internet). Más adelante se verán los distintos protocolos usa esta tecnología.

2.2.1. Digitalización

Para convertir la voz humana en su estado natural al formato digital que necesita la tecnología VoIP para su transmisión por las redes de Internet, se necesita hacer una conversión da la señal analógica a digital, DAC (Digital-to-Analog Conversion). En la ***'Figura 2.2.1'*** se presenta el esquema típico de una conversión analógica/digital.

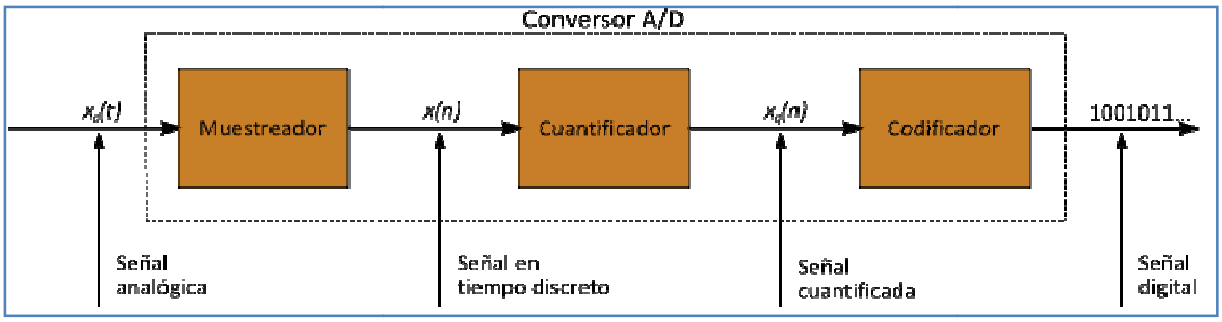

Figura 2.2.1: Esquema de conversión A/D (analógico->digital)

Como se ve, el proceso consta de tres pasos: muestreo, cuantificación y codificación. Además, se incluye una etapa de filtrado y compresión tras la etapa de cuantificación para preservar el ancho de banda y mejorar la eficiencia del mismo.

- ***MUESTREO***

Es la primera etapa que se encuentra la señal analógica en su conversión a digital. Consiste en tomar muestras periódicas de la amplitud de la señal de voz, siendo el intervalo entre las muestras constante. El ritmo de este muestreo, es la frecuencia o tasa de muestreo y determina el número de muestras que se pueden tomar en un intervalo de tiempo. La técnica de muestreo más común para convertir señales de audio a señales digitales es la 'Modulación por Impulsos Codificados' (MIC), donde la señal analógica será muestreada a la vez que su amplitud será discretizada y codificada en formato binario.

- ***CUANTIFICACIÓN***

Una vez realizado el muestreo, la siguiente etapa es la cuantificación de la señal analógica. Para esta parte del proceso los valores continuos de la sinusoide se convierten en series de valores numéricos decimales discretos correspondientes a los diferentes niveles o variaciones de voltajes que contiene la señal analógica original. Por tanto, la cuantificación representa la componente de muestreo de las variaciones de valores de tensiones o voltajes tomados en diferentes puntos de la onda sinusoidal, que permite medirlos y asignarles sus correspondientes valores en el sistema numérico decimal, antes de convertir esos valores en sistema numérico binario.

- ***CODIFICACIÓN***

Tras la cuantificación, los valores tomados en voltios se representan numéricamente por medio de códigos y estándares previamente establecidos. Lo más común es codificar la señal digital en código numérico binario. La codificación permite asignarle valores numéricos binarios equivalentes a los valores de tensión o voltajes que conforman la señal eléctrica analógica original.

En la ***'Figura 2.2.2.'*** se muestra la conversión que sufre la señal al pasar por las distintas etapas del convertidor.

Figura 2.2.2: Etapas de conversión de la señal analógica

Para establecer una llamada telefónica, un teléfono tradicional, sea analógico o digital (RDSI), requerirá un enlace con suficiente capacidad como para transportar un flujo de datos de 64 Kbps, que es la velocidad fijada para cualquier línea de teléfono tradicional. Tanto sistemas de telefonía analógica como digital operan en el mismo rango de muestreo de la señal de voz: 8000 Hz. Esta frecuencia combinada con una resolución de muestreo de 8 bits, da lugar a un régimen binario de 64kbps.
Pero, como todos los tipos de conversiones, existen limitaciones en la señal convertida con respecto a la original que se traducen en inconvenientes. Aun así, las ventajas son mayores, motivo por el que este tipo de tecnología se sigue desarrollando y evolucionando.

- ***VENTAJAS DE LA CONVERSIÓN***
  - No introduce ruidos en la transmisión.
  - Se guarda y procesa mucho más fácilmente que la analógica.
  - Posibilita almacenar grandes cantidades de datos en diferentes soportes.
  - Permite detectar y corregir errores con más facilidad.
  - Las grabaciones no se deterioran con el paso del tiempo como sucede con las cintas analógicas.
  - Permite realizar regrabaciones sucesivas sin que se pierda ninguna generación y, por tanto, calidad.
  - Permite la compresión para reducir la capacidad de almacenamiento.
  - No le afecta las interferencias atmosféricas ni de otro tipo cuando se transmite por vía inalámbrica, como ocurre con las transmisiones analógicas.

- ***INCONVENIENTES DE LA CONVERSIÓN***
  - Para su transmisión requiere un mayor ancho de banda en comparación con la analógica.
  - La sincronización entre los relojes de un transmisor inalámbrico digital y el receptor requiere que sea precisa.
  - Las transmisiones de las señales digitales son incompatibles con la mayoría de las instalaciones existentes para transmisiones analógicas.

2.2.2. Códecs

Códec es una abreviatura de Codificador-Decodificador. La mayor parte de los códecs provocan pérdidas de información para conseguir un tamaño lo más pequeño posible del archivo destino. Para el caso que se trata, se hablará de los códecs de audio. Un códec de audio es un tipo de códec específicamente diseñado para la compresión y descompresión de señales de sonido audible para el ser humano, y en este caso para señales de voz. Existen gran variedad de códecs disponibles, unos gratuitos, otros bajo licencia de pago, y cada uno comprime la señal de una forma, provocando que unos códecs realicen compresiones mayores que otros, a veces a costa de pérdida de calidad. La mayoría de estos códecs usados en redes VoIP son definidos por recomendaciones de la UIT-T (Unión Internacional de Telecomunicaciones-Sector de Normalización de las Telecomunicaciones). Los códecs de audio para aplicaciones de telefonía se dividen en dos grupos:

1. Los que se basan en la modulación por impulsos codificados (MIC) para transmitir la señal de audio.
2. Los que reestructuran la representación digital de la señal MIC en un formato más adecuado.

Ambos tipos de códecs son básicos de 64 Kbps. Pero aparte de éstos, existen los denominados vocoders, que van un paso más allá del algoritmo PCM, y los códecs híbridos, que reúnen las ventajas de los PCM y vocoders. Lo que persiguen todos estos códecs es reducir la cantidad de datos digitales necesarios para poder enviar la señal de voz y que en el destino ésta se pueda reproducir. Los códecs de audio se caracterizan por los siguientes parámetros:

- **Número de canales**: un flujo de datos codificado puede contener una o más señales de audio simultáneamente.

- **Frecuencia de muestreo**: el teorema de Nyquist, determina la calidad de voz mediante la máxima frecuencia que es capaz de codificar (la mitad de la frecuencia de muestreo). Por tanto, cuanto mayor sea la frecuencia de muestreo, mayor será la calidad del sonido obtenido respecto a la señal de audio original.

- **Número de bits por muestra**: determina la precisión con la que se reproduce la señal original y el rango dinámico de la misma.

- **Pérdida**: algunos códecs pueden eliminar frecuencias de la señal original que, teóricamente, son inaudibles para el ser humano. De esta manera se puede reducir la frecuencia de muestreo.

- **Bit-rate (tasa de bit)**: es el número de bits de información que se procesan por unidad de tiempo, teniendo en cuenta la frecuencia de muestreo resultante, la profundidad de la muestra en bits y el número de canales.

2.2.2.1. Códecs PCM

Transmiten información sobre la forma de onda de la señal de voz. Se caracterizan por tener una tasa de bit de 64 Kbps. Esta tasa es muy elevada para las posibilidades de algunas partes de la red, por lo que cada vez se utilizan menos este tipo de códecs. Quizá, el más representativo de todos sea el códec G.711. En este grupo se encuentran los códecs predictivos, que utilizan los datos previamente descodificados para predecir cómo va a ser el resultado. Después se compara esta predicción y la realidad, y se codifica ese error, junto con la información necesaria para la reproducción. En algunos casos se utilizan ambas, aplicando la transformación al resultado de la codificación. De esta forma codifican con una menor cantidad de bits ya que la señal de error es más pequeña que la muestra en sí. Con estos códecs se puede reducir la tasa de error hasta los 18 Kbps, a cambio de perder un poco de calidad.

2.2.2.2. Códecs VOCODERS

Los vocoders son analizadores/sintetizadores de voz. Estos códecs aprovechan las características de la señal de voz humana. Toman muestras de intervalos de la señal de voz de diferente duración (10ms, 20ms ó 30ms), según el tipo de códec. Las muestras se analizan mediante diferentes algoritmos para sacar los coeficientes del filtro vocal (que hace las veces del tracto vocal de una persona cuando habla) y para crear la señal de excitación (que simula el impulso del aire que pasa por las cuerdas vocales). Con esto se puede reconstruir posteriormente la voz en el receptor. Los vocoders comprimen bastante la información a transmitir y pueden alcanzar tasas de transmisión muy bajas. El problema es que la voz reproducida suena muy sintetizada y poco natural, lejos de la calidad que ofrecen los códecs PCM.

2.2.2.3. Códecs HÍBRIDOS

Estos códecs tienen las ventajas de los vocoders, en cuanto a que se basan en el modelo de excitación, más un filtro vocal para conseguir bajas tasas de bit a transmitir, y además poseen las ventajas de los predictivos porque comparan la muestra generada mediante la señal de excitación y el filtro calculados, con la original, para transmitir también el error cometido con muy pocos bits y conseguir más naturalidad en la voz reproducida en el destino. Con esto se consiguen tasas de transmisión también muy bajas y una calidad de voz considerablemente buena. Alguno de los actuales códecs híbridos que de momento han sido estandarizados por la UIT-T son el G.729 y el G. 723.

2.2.2.4. Tabla resumen de códecs

A modo de resumen, se presenta esta tabla de códecs en la que se indican algunas de las características a tener en cuenta.

Nombre	Estandarizado	Descripción	Bit rate (kb/s)	Sampling rate (kHz)	Frame size (ms)	Observaciones	MOS (Mean Opinion Score)
G.711 *	ITU-T	Pulse code modulation (PCM)	64	8	Muestreada	Tiene dos versiones u-law (US, Japan) y a-law (Europa) para muestrear la señal	4.1
G.721	ITU-T	Adaptive differential pulse code modulation (ADPCM)	32	8	Muestreada	Obsoleta. S e ha transformado en la G.726.	
G.722	ITU-T	7 kHz audio-coding within 64 kbit/s	64	16	Muestreada	Divide los 16 Khz en dos bandas cada una usando ADPCM	
G.722.1	ITU-T	Codificación a 24 y 32 kbit/s para sistemas sin manos con baja perdida de paquetes	24/32	16	20		
G.723	ITU-T	Extensión de la norma G.721 a 24 y 40 kbit/s para aplicaciones en circuitos digitales.	24/40	8	Muestreada	Obsoleta por G.726. Es totalmente diferente de G.723.1.	
G.723.1	ITU-T	Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s	5.6/6.3	8	30	Parte de H.324 video conferencing. Codifica la señal usando linear predictive analysis-by-synthesis coding. Para el codificador de high rate utiliza Multipulse Maximum Likelihood Quantization (MP-MLQ) y para el de low-rate usa Algebraic-Code-Excited Linear-Prediction (ACELP).	3.8-3.9
G.726	ITU-T	40, 32, 24, 16 kbit/s adaptive differential pulse code modulation (ADPCM)	16/24/32/40	8	Muestreada	ADPCM; reemplaza a G.721 y G.723.	3.85
G.727	ITU-T	5-, 4-, 3- and 2-bit/sample embedded adaptive differential pulse code modulation (ADPCM)	var.		Muestreada	ADPCM. Relacionada con G.726.	
G.728	ITU-T	Coding of speech at 16 kbit/s using low-delay code excited linear prediction	16	8	2.5	CELP.	3.61
G.729 **	ITU-T	Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)	8	8	10	Bajo retardo (15 ms)	3.92
GSM 06.10	ETSI	RegularPulse Excitation LongTerm Predictor (RPE-LTP)	13	8	22.5	Usado por la tecnología celular GSM	
LPC10	Gobierno de USA	Linear-predictive codec	2.4	8	22.5	10 coeficientes.La voz suena un poco "robotica"	
Speex			8, 16, 32	2.15-24.6 (NB) 4-44.2 (WB)	30 (NB) 34 (WB)		
iLBC			8	13.3	30		
DoD CELP	American Department of Defense (DoD) Gobierno de USA		4.8		30		
EVRC	3GPP2	Enhanced Variable Rate CODEC	9.6/4.8/1.2	8	20	Se usa en redes CDMA	
DVI	Interactive Multimedia Association (IMA)	DVI4 uses an adaptive delta pulse code modulation (ADPCM)	32	Variable	Muestreada		
L16		Uncompressed audio data samples	128	Variable	Muestreada		

* El códec g711 tiene dos versiones conocidas como ley A (usado en Europa) y ley μ (usado en USA y Japón). La ley μ se corresponde con el estándar T1 usado en Estados Unidos y la ley A con el

estándar E1 usado en el resto del mundo. La diferencia es el método que se utiliza para muestrear la señal. La señal no se muestrea de forma lineal sino de forma logarítmica.
** Existen varias versiones del códec g729 que es interesante explicar por su extendido uso:

- G729: es el códec original.
- G729A: es una simplificación de G729 y es compatible con G729. Es menos complejo pero tiene algo menos de calidad.
- G729B: Es G729 pero con supresión de silencios y no es compatible con las anteriores.
- G729AB: Es g729A con supresión de silencios y sería compatible solo con G729B.

Aparte de esto el códec G729 (en todas sus versiones) en general tienen una tasa de bit de 8Kbps pero existen versiones de 6.4 Kbps y 11.4 Kbps.

2.3. Arquitectura de protocolos

En los últimos años, los protocolos de señalización para el servicio de transmisión de voz han experimentado una fuerte evolución junto con la tendencia a transportar dicho tráfico desde las redes de conmutación de circuitos hacia las redes de conmutación de paquetes. La telefonía IP necesita de elementos y protocolos nuevos para conseguir transmitir la voz por las redes de datos hasta sus destinos con una calidad aceptable, lo que resulta, en principio, algo complejo. Un protocolo de señalización es un lenguaje común hablado por teléfonos, centralitas digitales implementadas en servidores, centralitas tradicionales y por cualquier otro elemento que pueda interferir en una comunicación telefónica. Actualmente existen dos importantes protocolos de señalización en el mundo de la telefonía IP: el Protocolo de Inicio de Sesión (SIP), desarrollado por el IETF (Internet Engeneering Task Force), y H.323, desarrollado por la UIT-T. Ambas arquitecturas básicas de protocolos tienen bastantes aspectos en común, aunque existen diferencias considerables en cuanto a sus arquitecturas de servicios suplementarios. H.323 está más orientada al mundo de la telefonía y las telecomunicaciones, mientras que SIP está más orientado a Internet y es bastante más sencillo que H.323.
Existen otra serie de protocolos de señalización, desarrollados por compañías privadas, como pueden ser: SCCP, desarrollado por la compañía Cisco, o IAX (actualmente evolucionado a IAX2), propiedad de la empresa Digium.
En la ***'Figura 2.3.1.'*** se puede ver la configuración de los protocolos usados en función de las necesidades.

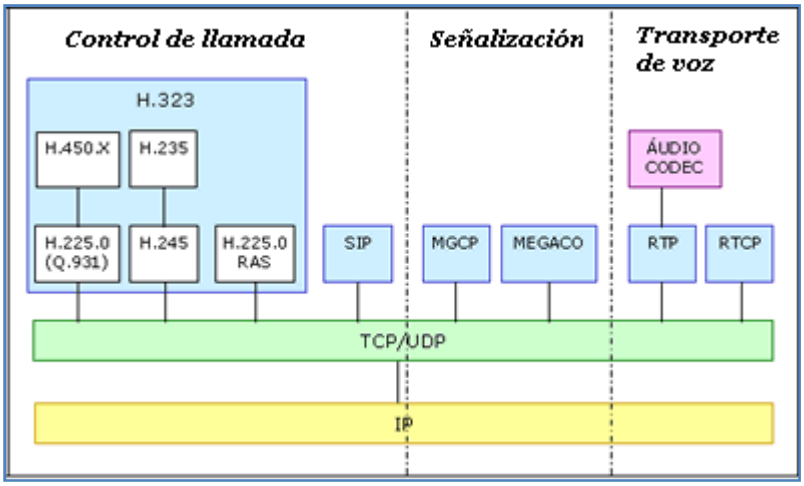

Figura 2.3.1: Torre de protocolos para VoIP

- ***Protocolos de Señalización (Control de Llamada)***

Protocolos H.323 y SIP. Aquí también se incluyen los protocolos IAX2 y SCCP. Utilizados por los teléfonos IP, computadoras, adaptadores IP, controladores de señalización y gateways para establecimientos, control y término de las llamadas.

- ***Protocolos de Control de Gateway (Señalización)***

MGCP: Media Gateway Control Protocol (IETF). Utilizado por los controladores de gateways y gateways para establecimiento, control y término de las llamadas.
MEGACO: Media Gateway Control Protocol (IETF/UIT-T). Tiene la misma finalidad del MCGP, sin embargo fue desarrollado para ser una alternativa a ese protocolo, adecuándose también a controladores distribuidos de gateways, a controladores multipunto (Conferencia) y a unidades interactivas de respuesta audible.

- **Protocolos de Media (Transporte de Voz)**

RTP: Real-Time Transport Protocol o Protocolo de Transporte en Tiempo Real (IETF). Protocolo responsable del transporte de voz en tiempo real entre los servidores y gateways. Es el estándar más utilizado actualmente para esta finalidad.
RTCP: Real-Time Transport Control Protocol o Protocolo de Control de Transporte en Tiempo Real (IETF). Protocolo responsable del control del transporte de voz realizado por el RTP en los sistemas VoIP.

En este apartado se van explicar algunos de estos protocolos y su estructura de red.

2.3.1. H.323

H.323 es la especificación establecida por la UIT (Unión Internacional de Telecomunicaciones) en 1996 que fija los estándares para la comunicación de voz y vídeo sobre redes de área local. Para la conferencia de datos se apoya en la norma T.120, con lo que en conjunto soporta las aplicaciones multimedia. Los terminales y equipos conforme a H.323 pueden tratar voz en tiempo real, datos y vídeo, incluida videotelefonía.

El protocolo TCP/IP utilizado en múltiples comunicaciones es un protocolo de transferencia seguro, gracias a TCP, lo que asegura la transmisión libre de errores. Sin embargo, no hay garantía de que los paquetes lleguen ordenados a su destino en tiempo real, lo que causa problemas para la voz o el vídeo. Para evitar este efecto, el IETF ha propuesto el protocolo denominado RTF (Real-time Transfer Protocol) que facilita las comunicaciones multimedia.

El estándar H.323 proporciona la base para la transmisión de voz, datos y vídeo sobre redes no orientadas a conexión como son las basadas en IP, incluida Internet, de manera que las aplicaciones puedan 'convivir', permitiendo la comunicación entre los usuarios sin necesidad de que éstos se preocupen por la compatibilidad de sus sistemas. La topología de la LAN sobre la que los terminales H.323 se comunican puede variar de más simple a más compleja, lo que puede resultar en un grado variable de rendimiento.

Esta arquitectura contempla el control de la llamada, gestión de la información y ancho de banda para una comunicación punto a punto y punto a multipunto dentro de la LAN, así como define interfaces entre la LAN y otras redes externas (como por ejemplo la RDSI). Este estándar es una parte de una serie de especificaciones para videoconferencia sobre distintos tipos de redes, que incluyen desde la H.320 a la H.324, que son válidas para RDSI y RTC respectivamente.

H.323 establece los estándares para la compresión y descompresión de audio y vídeo, asegurando que los equipos de distintos fabricantes se entiendan. Así, los usuarios no se tienen que preocupar de cómo el equipo receptor actúe, siempre y cuando cumpla este estándar. La gestión del ancho de

banda disponible para evitar que la LAN se colapse con la comunicación de audio y vídeo (por ejemplo limitando el número de conexiones simultáneas) también está contemplada en el estándar.

La norma H.323 hace uso de los procedimientos de señalización de los canales lógicos contenidos en la norma H.245, en los que el contenido de cada uno de los canales se define cuando se abre. Estos procedimientos se proporcionan para fijar las prestaciones del emisor y receptor, el establecimiento de la llamada, intercambio de información, terminación de la llamada y cómo se codifica y decodifica. Por ejemplo, cuando se origina una llamada telefónica sobre Internet, los dos terminales deben negociar cuál de los dos ejerce el control, de manera tal que sólo uno de ellos origine los mensajes especiales de control. Una cuestión importante es, como se ha dicho, que se deben determinar las capacidades de los sistemas, de forma que no se permita la transmisión de datos si no pueden ser gestionados por el receptor. En la ***'Figura 2.3.2'*** se esquematiza el uso de de los distintos protocolos en los que se apoya la arquitectura H.323.

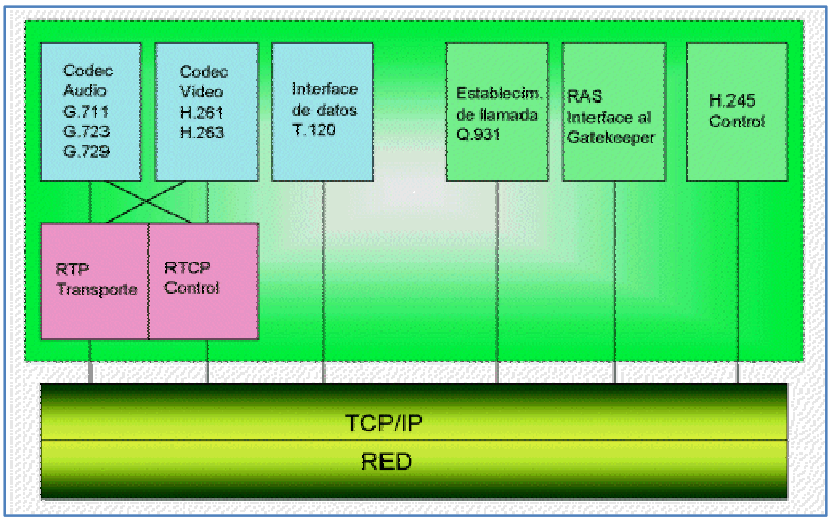

Figura 2.3.2: Arquitectura del estándar H.323

Una característica de la telefonía sobre Internet es que se permite la información de vídeo sobre la de audio (videoconferencia), que se formatea de acuerdo con el estándar H.261 o H.263, formando parte de la carga útil del paquete RTP. El problema es que la transmisión de video resulta muy sensible a la pérdida de paquetes ya que origina distorsión en la imagen recibida.

2.3.1.1. Componentes

La especificación define cuatro componentes principales para un sistema de comunicaciones en red: Terminales, Gateways, Gatekeepers y MCUs como se precia en la ***'Figura 2.3.3.'***

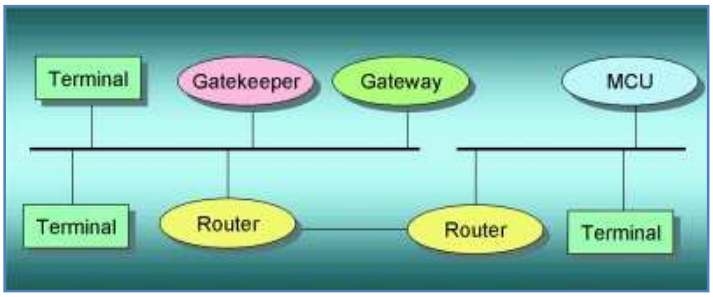

Figura 2.3.3: Componentes del estándar H.323

- **TERMINALES**

Son los clientes finales en la LAN, que proporcionan una comunicación bidireccional en tiempo real. Todos los terminales deben soportar la comunicación de voz, mientras que la de vídeo y datos son opcionales. Además, deben soportar la norma H.245 que se emplea para la negociación del uso del canal y sus prestaciones y que consta de: Q.931 para el establecimiento de la llamada y la señalización, RAS (Registration/ Admission/Status) un protocolo utilizado para la comunicación con el Gatekeeper sólo si éste está presente en la red, soporte para RTP/RTCP (Real-time Transport Protocol/Real-time Transport Control Protocol) que fija la secuencia de los paquetes de audio y vídeo). Opcionalmente los terminales pueden incorporar un códec para videoconferencia según T.120 y MCU (Multipoint Control Unit o Unidad de Control Multipunto).

- **GATEWAY**

El Gateway es un elemento opcional en una conferencia H.323, que proporciona entre otros servicios la adaptación con otras normas del UIT. En general, su misión es establecer un enlace con otros terminales ubicados en la RTC o RDSI. Integra en el mundo VoIP, terminales que no lo son.

- **GATEKEEPER**

El Gatekeeper realiza dos funciones de control de llamadas que preservan la integridad de la red corporativa de datos. La primera es la traslación de direcciones de los terminales de la LAN a las correspondientes IP. La segunda es la gestión del ancho de banda, fijando el número de conferencias que pueden estar dándose simultáneamente en la LAN y rechazando las nuevas peticiones por encima del nivel establecido, de manera tal que se garantice ancho de banda suficiente para las aplicaciones de datos sobre la LAN. El Gatekeeper proporciona todas las funciones anteriores para los terminales, Gateways y MCUs, que están registrados dentro de la denominada 'Zona de control' H.323.

- **MCU (Multipoint Control Unit)**

La Unidad de Control Multipunto está diseñada para soportar la conferencia entre tres o más puntos, bajo el estándar H.323, llevando la negociación entre terminales para determinar las capacidades comunes para el proceso de audio y vídeo y controlar la multidifusión.

La comunicación bajo H.323 contempla las señales de audio y vídeo. La señal de audio se digitaliza y se comprime bajo uno de los algoritmos soportados, tales como el G.711 o G.723, y la señal de vídeo (opcional) se trata con la norma H.261 o H.263. Los datos (opcional) se manejan bajo el estándar T.120 que permite la compartición de aplicaciones en conferencias punto a punto y punto a multipunto.

2.3.2. SIP

El protocolo de inicio de sesión, SIP (Session Initiation Protocol), es un protocolo para el establecimiento de sesiones sobre una red IP. Una sesión que puede soportar desde una llamada telefónica hasta una multiconferencia multimedia. Está siendo desarrollado por el IETF, con la misma filosofía de sencillez y mínimo esfuerzo de siempre. Está pensado como un mecanismo para el establecimiento, la terminación y la modificación de sesiones. SIP es un estándar emergente para establecer, enrutar y modificar sesiones de comunicaciones a través de las redes de Internet. Utiliza el modelo de Internet y lo convierte al mundo de las telecomunicaciones, utilizando protocolos ya existentes tales como HTTP y SMTP. También usa una estructura de dirección URL. Usa las direcciones de tipo correo electrónico para identificar a los usuarios en lugar de los dispositivos que los utilizan. De esta forma SIP no depende del dispositivo y no hace distinción alguna entre voz y datos, teléfono u ordenador. Como se describe a continuación, SIP es usado más para el manejo de servicios, mientras que H.323 se usa prácticamente para la conversión del número telefónico en paquetes IP.

Se trata de un protocolo basado en petición-respuesta (request-response), al igual que HTTP o SMTP. SIP maneja mensajes de petición que se estructuran en tres bloques:

1) *Request Line*
2) *Cabecera*
3) *Cuerpo*

y mensajes de respuesta, también en tres:

1) *Status Line*
2) *Cabecera*
3) *Cuerpo*

En ambos casos el cuerpo es independiente de SIP y puede contener cualquier cosa. A efectos de estandarización se definen métodos para describir las áreas de especificación. SIP define los siguientes métodos:

- ***INVITE***: este método es usado para establecer sesiones y anunciar las capacidades de los nodos SIP.
- ***ACK:*** es usado para confirmar que el cliente solicitante ha recibido una respuesta final desde un servidor a una solicitud INVITE, reconociendo la respuesta como afirmativa.
- ***OPTIONS:*** es usado para preguntar a un nodo SIP por sus capacidades, sin que ningún canal multimedia haya sido establecido aún.
- ***BYE:*** cuando la llamada es completada, es decir, cuando alguno de los extremos involucrados en la comunicación desea finalizar la llamada.
- ***CANCEL:*** cancela una solicitud pendiente, pero no afecta a una solicitud ya completada. Este método finaliza una solicitud de llamada incompleta.
- ***REGISTER:*** notifica al servidor SIP en qué terminal SIP puede ser alcanzado un usuario.
- ***INFO:*** es usado para transmitir señales de aplicación de telefonía a través del canal usado por la señalización SIP (por ejemplo dígitos marcados).
- ***PRACK:*** este método es usado en lugar de ACK para notificar al otro extremo que se está estableciendo una llamada.
- ***SUBSCRIBE:*** este método provee una forma de establecer manejadores de eventos dentro de aplicaciones de telefonía SIP.
- ***NOTIFY:*** este método entrega mensajes entre extremos SIP, tales como eventos ocurridos durante la llamada.

En la ***'Figura 2.3.4.'*** se ilustra un establecimiento de llamada en SIP, con algunos de los menajes que se intercambian.

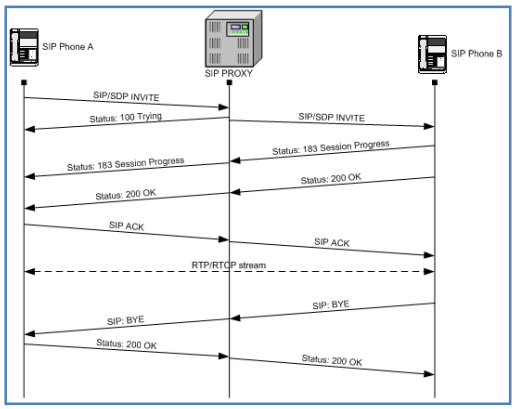

Figura 2.3.4: Establecimiento de llamada en SIP

Las respuestas son del tipo HTTP:

- 1xx Informational (100 Trying, 180 Ringing, 181 Call is being forwarded)
- 2xx Successful (200 OK, 202 Accepted)
- 3xx Redirection (300 Multiple choices, 301 Moved Permanently, 302 Moved Temporarily)
- 4xx Client Error (400 Bad Request, 404 Not Found, 482 Loop Detected, 486 Busy here)
- 5xx Server Failure (500 Server Internal Error, 501 Not Implemented)
- 6xx Global Failure (600 Busy Everywhere, 603 Decline).

SIP se puede definir como un protocolo de control, pensado para la creación, modificación y terminación de sesiones, con uno o más participantes. Esas sesiones pueden comprender conferencias multimedia, llamadas telefónicas sobre Internet (o cualquier otra red IP), distribución de contenidos multimedia, etc. Los participantes pueden negociar los contenidos y capacidades que van a utilizar. Soporta movilidad de los usuarios, mediante utilización de proxys.
Las funcionalidades que se le exigen a un protocolo de estas características, son básicamente:

- La traducción de nombres y la ubicación de usuarios.
- La negociación de capacidades de cada usuario.
- La gestión de los usuarios que toman parte en una conferencia (sesión).
- La gestión de los cambios en las capacidades de cada participante.

Está pensado para ser independiente de los niveles inferiores; sólo necesita un servicio de datagramas no fiable, con lo cual se puede montar sobre UDP o TCP. Sobre ese servicio no fiable se monta un transporte con RTP/RTCP.

2.3.2.1. Arquitectura SIP

SIP necesita dos componentes básicos: un agente de usuario (UA, User Agent) y un servidor (NS, Network Server). El agente de usuario, comprende un elemento cliente (UAC, User Agent Client) y un elemento servidor (UAS, User Agent Server). El cliente inicia las llamadas, y el servidor las responde: la idea es realizar llamadas (establecer sesiones 'peer-to-peer', P2P) con un protocolo Cliente/Servidor.
Las funciones principales de los servidores SIP son la resolución de nombres y la ubicación de usuarios. Los servidores pueden guardar o no información de estado, dando lugar a dos modos de funcionamiento: 'statefull' o 'stateless' respectivamente.

Otras funcionalidades importantes de los servidores son la redirección (de una petición) y la distribución (pueden pasar una llamada a un grupo de usuarios, apropiándose de la sesión el primero que conteste).
Con esos componentes, UAC, UAS y NS, se puede montar una infraestructura básica de SIP, sobre la cual se pueden montar servidores de aplicaciones que podrían alojar módulos de servicio (de mensajería instantánea, de presencia, de control de llamada, perfiles de usuario...).

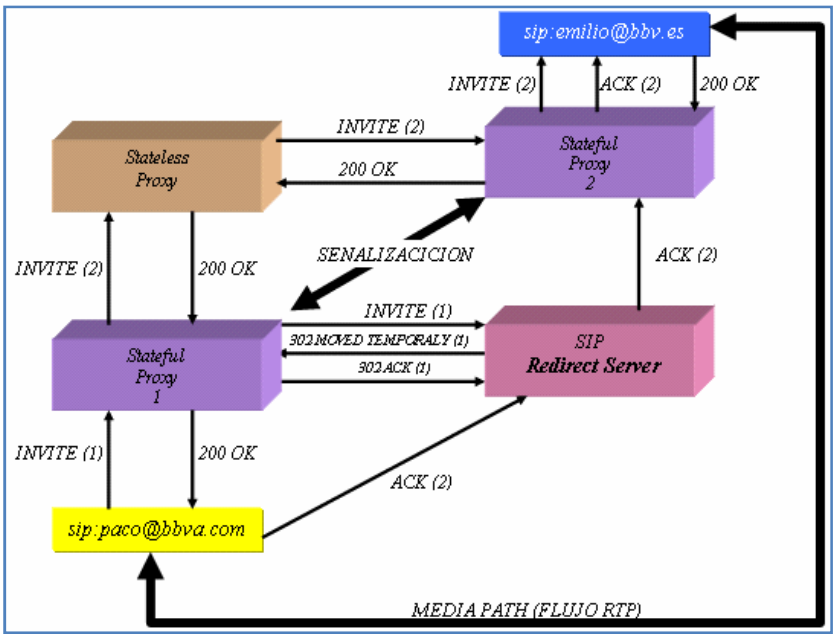

Figura 2.3.5: Elementos básicos del establecimiento de llamada

En la ***'Figura 2.3.5.'*** se pretende ilustrar el establecimiento de una llamada para mostrar cómo interactúan los elementos básicos que se han mencionado más arriba.
En este ejemplo, el usuario *paco@bbva.com* quiere hablar con *emilio@bbva.com,* es decir, con un usuario que habitualmente está en su mismo dominio, pero por algún motivo, que desconocemos, hoy no está en *bbva.com*, sino en *bbv.es* aunque *paco* no lo sabe. Siendo así, manda una invitación (INVITE) para el usuario *emilio@bbva.com* al servidor responsable de su dominio (en este caso es un servidor proxy con estado: 'Stateful Proxy 1'). El servidor enviará la invitación a un servidor de redirección para tratar de averiguar la localización actual de *emilio*. Es este servidor de redirección el que determina que el usuario *emilio* está en el dominio *bbv.es* y le contesta al proxy con un '302 MOVED TEMPORARILY que incluye la nueva dirección de *emilio* (*sip:emilio@bbv.es*). El proxy responde con un '302 ACK', puesto que aquí termina la secuencia de la invitación inicial (INVITE (1) de la ***Figura 2.3.5***).
A partir de esta situación, el Proxy 1 con estado (Stateful Proxy 1) podría mandarle la dirección de *emilio* a *paco* para que él tratara de comunicarse directamente con *sip:emilio@bbv.es*. En el ejemplo, lo que hace el proxy 1 es modificar la invitación y tratar de encontrar a *sip:emilio@bbv.es*. Como no conoce a ningún otro servidor con estado que se responsabilice del dominio *bbv.es*, pasará la invitación a un servidor sin estado ('Stateless proxy') que conocerá el siguiente salto que debe seguir para llegar hasta *sip:emilio@bbv.es*.
Para simplificar el ejemplo el primer proxy sin estado conoce a un servidor proxy que controla el dominio *bbv.es* ('Stateful Proxy 2'). Ese segundo proxy completa la entrega de la invitación para *sip:emilio@bbv.es*, momento en el que *emilio* acepta la llamada enviando un mensaje de respuesta (200 OK), que recorre el mismo camino de vuelta de la invitación hasta llegar a *sip:paco@bbva.com*. Ahora *paco* debería mandarle un ACK de esta respuesta a *emilio*; y aunque en principio podría hacerlo directamente, en nuestro ejemplo hemos decidido que toda la señalización pase por los proxys de cada dominio (se supone que así lo habrán indicado en los mensajes de invitación que se han cruzado). De esta forma queda establecida la comunicación.

SIP proporciona los mecanismos necesarios para ofrecer una serie de servicios:

- **Usuarios:**
  - Localización.
  - Disponibilidad y capacidades (servicio de presencia y terminal asociado).

  - Perfil.
- **Llamadas**
  - Establecimiento.
  - Mantenimiento.
  - Desvíos.
  - Traducción de direcciones.
  - Entrega de los números llamado y llamante.
  - Movilidad: direccionamiento único independiente de la ubicación del usuario.
  - Negociación del tipo de terminal
  - Negociación de las capacidades del terminal.
  - Autenticación de usuarios llamado y llamante.
  - Transferencias ciegas y supervisadas.
  - Incorporación a conferencias multicast.

Este es el protocolo que más utilidad ha dado a los proyectos desarrollados en Clever Tecnología S.L., además de por su simplicidad y fácil manejo, por la implantación conjunta que tiene con el software Asterisk. Esta aplicación también incorpora el protocolo IAX2 que todavía simplifica más su implementación, pero al no estar estandarizado, dispone de pocos terminales en uso para poder ser comercializado.

2.3.3. IAX

2.3.3.1. Arquitectura

El protocolo IAX se corresponde con Inter-Asterisk eXchange protocol. Como indica su nombre fue diseñado como un protocolo de conexiones VoIP entre servidores Asterisk aunque hoy en día también sirve para conexiones entre clientes y servidores que soporten el protocolo.

La versión actual es IAX2 ya que la primera versión de IAX ha quedado obsoleta. Es un protocolo diseñado y pensado para su uso en conexiones de VoIP aunque puede soportar otro tipo de conexiones (como video).

Los objetivos de IAX son:

- Minimizar el ancho de banda usado en las transmisiones de control y multimedia de VoIP.
- Evitar problemas de NAT (Network Address Translation).
- Soporte para transmitir planes de marcación.

Entre las medidas para reducir el ancho de banda cabe destacar que IAX o IAX2 es un protocolo binario en lugar de ser un protocolo de texto como SIP y que hace que los mensajes usen menos ancho de banda.

Para evitar los problemas de NAT el protocolo IAX2 usa como protocolo de transporte UDP, normalmente sobre el puerto 4569 (el IAX usaba el puerto 5036), y tanto la información de señalización como los datos viajan conjuntamente y por tanto lo hace menos vulnerable a problemas de NAT y le permite pasar los routers y firewalls de manera más sencilla.

IAX fue creado por Mark Spencer para paliar una serie de problemas e inconvenientes que se encontró al utilizar SIP en VoIP y que pensó que debían ser mejorados.

2.3.3.2. Comunicación IAX2

Para poder entender el protocolo IAX vamos a ver un ejemplo del flujo de datos de una comunicación IAX2:

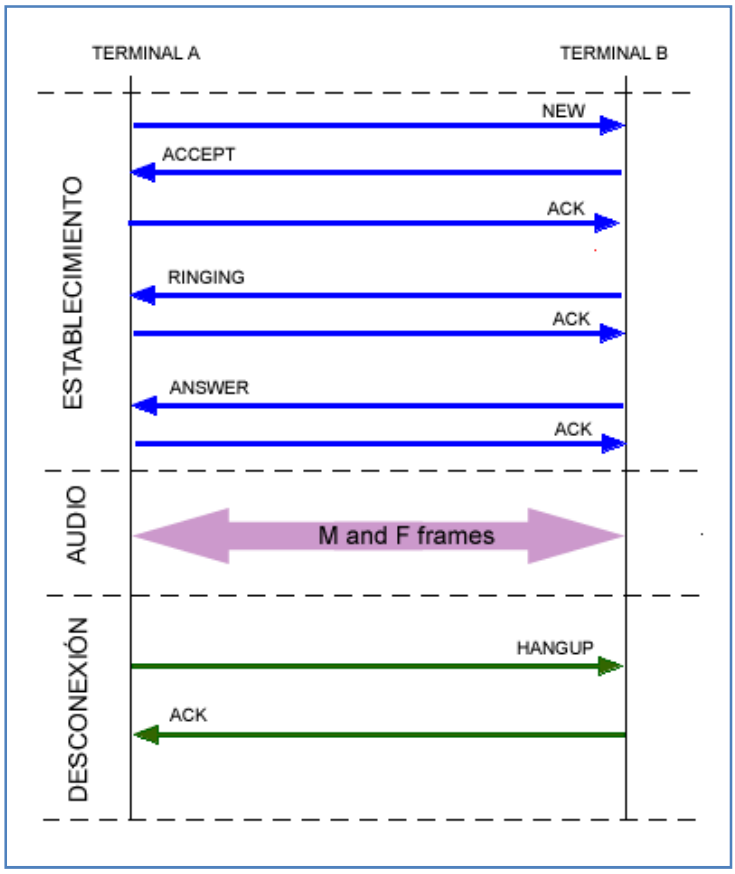

Figura 2.3.6: Flujo de datos del protocolo IAX2

Una llamada IAX2 tiene tres fases:

1. ***Establecimiento de la llamada***

El terminal llamante (A) inicia una conexión y manda un mensaje 'new'. El terminal llamado (B) responde con un 'accept' y A le responde con un 'Ack'. A continuación B da las señales de 'ringing' y A contesta con un 'ack' para confirmar la recepción del mensaje. Por último, B acepta la llamada con un 'answer' y A confirma ese mensaje.

2. ***Flujo de datos o flujo de audio***

Se mandan las tramas M y F en ambos sentidos con la información vocal. Las tramas M son 'mini-tramas' que contienen solo una cabecera de 4 bytes para reducir el uso en el ancho de banda. Las tramas F son tramas completas que incluyen información de sincronización. Es importante volver a resaltar que en IAX el flujo utiliza el mismo protocolo UDP que usan los mensajes de señalización evitando problemas de NAT.

3. ***Liberación de la llamada o desconexión***

La liberación de la conexión se realiza mediante el mensaje de 'hangup' y confirmación de dicho mensaje.

2.3.3.3. Diferencias entre IAX y SIP

La primera diferencia considerable es la forma de registro con la centralita principal, debido a que no existen muchos teléfonos IP que implementen SIP. La disposición de los terminales en función del protocolo elegido se muestra en la '***Figura 2.3.7***':

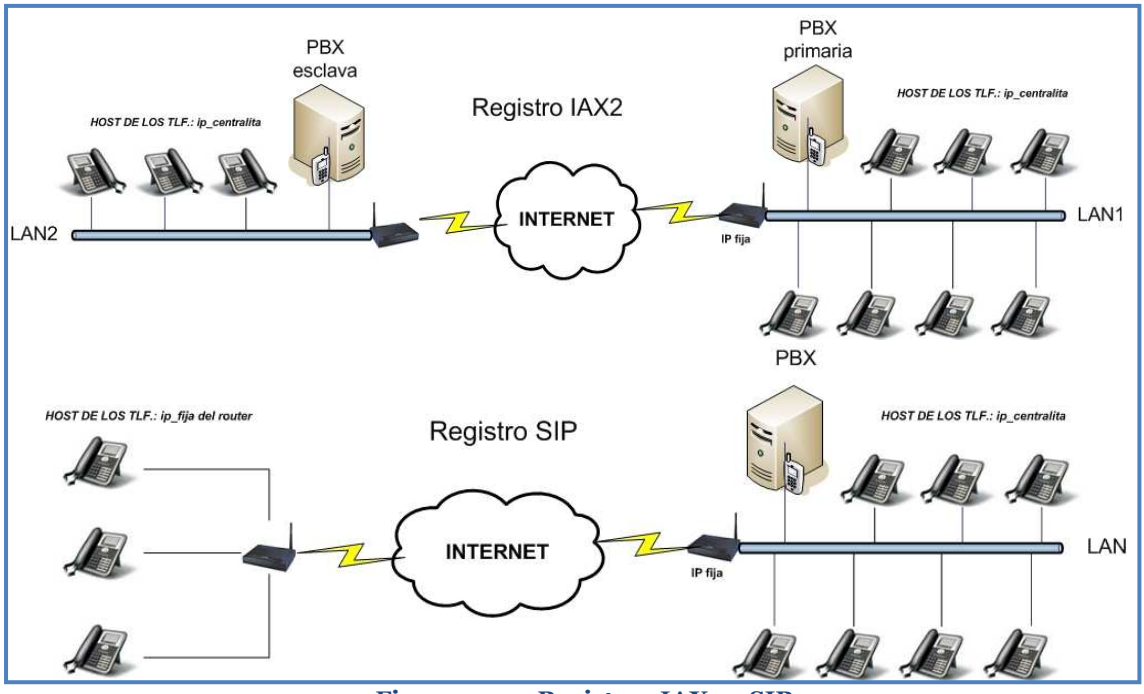

Figura 2.3.7: Registros IAX2 y SIP

Las principales diferencias ente IAX y SIP son las siguientes:

- ***Ancho de banda***

IAX utiliza un menor ancho de banda que SIP ya que los mensajes son codificados de forma binaria mientras que en SIP son mensajes de texto. Asimismo, IAX intenta reducir al máximo la información de las cabeceras de los mensajes reduciendo también el ancho de banda.

- ***NAT***

En IAX la señalización y los datos viajan conjuntamente con lo cual se evitan los problemas de NAT que frecuentemente aparecen en SIP. En SIP la señalización y los datos viajan de manera separada y por eso aparecen problemas de NAT en el flujo de audio cuando éste debe superar los routers y firewalls.

- ***Estandarización y uso***

SIP es un protocolo estandarizado por la IETF hace bastante tiempo y que es ampliamente implementado por todos los fabricantes de equipos y software. IAX está aun siendo estandarizado y es por ello que no se encuentra en muchos dispositivos existentes en el mercado.

- ***Utilización de puertos***

IAX utiliza un solo puerto (4569) para mandar la información de señalización y los datos de todas sus llamadas. Para ello utiliza un mecanismo de multiplexión o 'trunking'. SIP, sin embargo utiliza un puerto (5060) para señalización y 2 puertos RTP por cada conexión de audio (como mínimo 3 puertos). Por ejemplo para 100 llamadas simultáneas con SIP se usarían 200 puertos (RTP) más el puerto 5060 de señalización. IAX utilizaría sólo un puerto para todo (4569).

- ***Flujo de audio al utilizar un servidor***

En SIP si utilizamos un servidor la señalización de control pasa siempre por el servidor pero la información de audio (flujo RTP) puede viajar extremo a extremo sin tener que pasar necesariamente por el servidor SIP. En IAX al viajar la señalización y los datos de forma conjunta

todo el tráfico de audio debe pasar obligatoriamente por el servidor IAX. Esto produce un aumento en el uso del ancho de banda que deben soportar los servidores IAX sobre todo cuando hay muchas llamadas simultáneas.

- ***Otras funcionalidades***

IAX es un protocolo pensado para VoIP y transmisión de video y presenta funcionalidades interesantes como la posibilidad de enviar o recibir planes de marcado (dialplans) que resultan muy interesante al usarlo conjuntamente con servidores Asterisk. SIP es un protocolo de propósito general y podría transmitir sin dificultad cualquier información y no sólo audio o video.

2.3.4. Zaptel

Zaptel no es exactamente un protocolo, sino un driver de conexión de hardware con el sistema Asterisk. Es la abreviatura de ZAPata TELephony. Sirve para poder conectar las tarjetas Digium (o similares), como la TDM400P que es capaz de interconectar la RTC con el sistema de VoIP Asterisk. El driver dispone de un archivo editable mediante el que se pueden configurar los distintos modos y puertos que disponga la tarjeta de comunicaciones. Ese archivo se denomina zaptel.conf y se ubica en el directorio *etc/*. Hay dos archivos de configuración con nombres similares, pero tratan aspectos distintos:

- */etc/zaptel.conf*: Configuración de las interfaces del hardware
- */etc/asterisk/zapata.conf*: Configuración del sistema Asterisk para utilizar las interfaces del hardware que se configuraron en zaptel.conf.

Los ajustes de la configuración establecidos en zaptel.conf son:

1. Un número de identificación para una tarjeta TDM de Digium que será utilizado en el archivo extensions.conf para referir a una tarjeta de interfaz específica. Además, se debe asignar un número único.
2. Idioma del tono que debe asociarse con la tarjeta de TDM (español, inglés, francés, alemán, etc.). Esto sirve para tener un tono de llamada familiar del país en el que se esté.

2.3.4.1. Señalización FXO y FXS

Antes de configurar las tarjetas y sus canales, hay que identificar los tipos de señalización FXO y FXS:

- **FXO**

Un Foreign Exchange Office (FXO) es cualquier dispositivo que, desde el punto de vista de la central telefónica, actúa como un teléfono tradicional. Debe ser capaz de aceptar señales de llamada o ring, ponerse en estado de colgado o descolgado, y enviar y recibir señales de voz. Es como un 'teléfono' o cualquier otro dispositivo que 'suena' (como una máquina de fax o un módem).

- **FXS**

Un Foreign Exchange Station (FXS) es lo que está situado al otro lado de una línea telefónica tradicional (la estación). Es el que envía el tono de marcado, la señal de llamada que hace sonar los teléfonos y los alimenta. En líneas analógicas, un FXS alimenta al FXO. El FXS utiliza alrededor de 48 voltios DC para alimentar al teléfono durante la conversación y hasta 80 voltios AC (20 Hz) cuando genera el tono de llamada (ring).

En resumen, como se aprecia en la '***Figura 2.3.8***':

1. Un FXS necesita estar conectado a un FXO (como una línea telefónica necesita estar conectada a un teléfono) o viceversa.
2. Un FXS suministra energía (elemento activo) a un teléfono FXO (elemento pasivo).

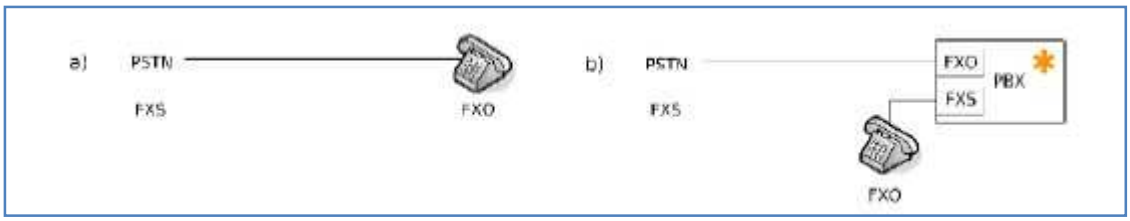

Figura 2.3.8: Señalización FXO y FXS

2.3.4.2. Configuración de las tarjetas

En la '***Figura 2.3.9***' se muestra la tarjeta TDM400P de Digium, que es la que va a servir de ejemplo de configuración.

Figura 2.3.9: TDM400P de Digium

Una vez colocada la tarjeta en su puerto respectivo, se debe saber si es reconocida tecleando en línea de comandos:

```
ProxyAFR:/usr/src# lspci | grep Communication

0000:06:05.0 Communication controller: Tiger Jet Network Inc. Tiger3XX
Modem/ISDN interface
0000:08:02.0 Communication controller: Tiger Jet Network Inc. Tiger3XX
Modem/ISDN interface
```

El siguiente paso es instalar zaptel para que la tarjeta pueda trabajar:

```
aptitude install zlib1g-dev
cd /usr/src/zaptel-1.2.12
make clean; make; make install
```

Y recompilar asterisk:

```
<code>
cd /usr/src/asterisk-1.2.14
make; make install
```

Se cargan los módulos:

```
modprobe zaptel
```

```
modprobe wcfxo
modprobe wctdm
```

Lo siguiente será modificar los archivos /etc/zaptel.conf y /etc/asterisk/zapata.conf:

- ***zaptel.conf***

Ejecutando en línea de comandos,

```
nano /etc/zaptel.conf
```

se accede al fichero editable de zaptel y se ve,

```
fxsks=1-4
loadzone=us
defaultzone=us
```

La línea fxsks=1-4 (1-4 es lo mismo que decir 1 2 3 y 4) indica que los cuatro módulos son FXO y por ello deben llevar señalización FXS. Si hubiese dos tarjetas TDM400P debería haber una sola línea fxsks=1-8. Es importante zaptel.conf porque es donde se define la señalización que cada puerto va a tener. Supongamos que esta tarjeta es TDM22B que tiene dos puertos FXS y dos puertos FXO en este orden FXS FXS FXO FXO, la línea en zaptel.conf debería quedar algo así:

```
fxoks=1-2
fxsks=3-4
```

Ahora hay que ejecutar ztcfg para confirmar que la configuración es la correcta, de manera que, si ha habido alguna equivocación con la señalización de cada puerto saldrá un mensaje de error que lo indicará para poder corregirlo. Tecleando en línea de comandos:

```
ztcfg –vvv
```

Y aparecerá algo como:

```
Zaptel Configuration
======================

Channel map:

Channel     01:     FXS     Kewlstart     (Default)     (Slaves:     01)
Channel     02:     FXS     Kewlstart     (Default)     (Slaves:     02)
Channel     03:     FXS     Kewlstart     (Default)     (Slaves:     03)
Channel     04:     FXS     Kewlstart     (Default)     (Slaves:     04)

4 channels configured
```

- ***zapata.conf***

Ahora hay que editar /etc/asterisk/zapata.conf para los cuatro canales FXO:
Ejecutando en línea de comandos,

```
nano /etc/asterisk/zapata.conf
```

se accede así al fichero editable de zapata en el que se ve,

```
[channels]
context=pstn
usecallerid=yes
hidecallerid=no
inmediate=no

; configuración del primer canal FXO
signalling=fxs_ks
echocancel=yes
group=1
channel=1

; configuración del segundo canal FXO
signalling=fxs_ks
echocancel=yes
group=2
channel=2

; configuración del tercer canal FXO
signalling=fxs_ks
echocancel=yes
group=3
channel=3

; configuración del cuarto canal FXO
signalling=fxs_ks
echocancel=yes
group=4
channel=4
```

Con esto se dispone de 4 canales de comunicación para tener interconectados la centralita Asterisk con la red telefónica conmutada.