

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 441 203**

51 Int. Cl.:

H04L 12/64 (2006.01)
H04L 12/66 (2006.01)
G06F 17/00 (2006.01)
H04L 29/06 (2006.01)
H04M 7/00 (2006.01)
H04L 29/12 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **08.06.2000 E 00947639 (1)**

97 Fecha y número de publicación de la concesión europea: **23.10.2013 EP 1186144**

54 Título: **Aparato y sistema de telefonía de red para telefonía por inter/intranet**

30 Prioridad:

08.06.1999 US 138832 P

45 Fecha de publicación y mención en BOPI de la traducción de la patente:

03.02.2014

73 Titular/es:

**THE TRUSTEES OF COLUMBIA UNIVERSITY IN
THE CITY OF NEW YORK (100.0%)
BROADWAY AND WEST 116TH STREET
NEW YORK, NY 10027-6699, US**

72 Inventor/es:

**SCHULZRINNE, HENNING y
YIN, JIANQI**

74 Agente/Representante:

ZUAZO ARALUZE, Alexander

ES 2 441 203 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

DESCRIPCIÓN

Aparato y sistema de telefonía de red para telefonía por inter/intranet

5 **Campo de la invención**

La presente invención se refiere en general al campo de la telefonía por Internet e intranet. Más particularmente, la presente invención se refiere a un aparato y a un sistema de telecomunicaciones de red para comunicaciones por Internet/intranet.

10

Antecedentes de la invención

Durante los últimos años, Internet ha evolucionado de un medio de comunicaciones adicional conveniente a una herramienta de comunicación esencial en los ámbitos comercial, técnico y educativo. A este respecto, un segmento cada vez mayor de Internet está relacionado con la telefonía por Internet que proporciona varias ventajas con respecto a una red conmutada por circuitos convencional controlada por una red de señalización separada. Por una parte, se permite a las partes seleccionar y usar de una manera más fácil la codificación y otras técnicas de compresión de datos que sean las más apropiadas para sus requisitos de calidad. Por ejemplo, las partes pueden decidir que para llamadas internacionales, cambiarían un menor coste por una plena calidad de llamada de larga distancia, mientras que un reportero que llama para dar su historia a una estación de radio puede decantarse por una plena calidad de FM sin preocuparse mucho por el precio. Incluso sin degradación de la calidad, son suficientes de 5,3 kb/s (G.723.1) a 8 kb/s (G.729) para soportar casi la calidad de llamada de larga distancia en contraposición con 64 kb/s para redes de telefonía de línea terrestre convencionales. Esta flexibilidad también tiene la ventaja de que durante una sobrecarga de red grave, por ejemplo, tras una catástrofe natural, los clientes de telefonía todavía pueden comunicarse a aproximadamente 3 kb/s, aumentando por tanto la capacidad de red por veinte.

Aunque es lógico extender los servicios de telefonía a las redes de datos existentes, tales como Internet, debido a la inteligencia requerida en los sistemas finales, el coste representa una desventaja importante. Antes resultaba difícil construir "teléfonos" de voz por paquetes sin necesidad de potencia externa que operasen a través de cables de par trenzado de baja calidad a varias millas de distancia al coste de un teléfono analógico básico.

Además, la mayoría de los productos de telefonía por Internet conocidos están diseñados para operar según el protocolo de señalización y control H.323. El protocolo H.323 es un protocolo complejo que es difícil de usar e implementar. Como resultado de esta complejidad, diferentes implementaciones de dispositivos H.323 pueden verse afectados de manera adversa por problemas de compatibilidad. Además, los dispositivos que operan con el protocolo H.323 no pueden comunicarse directamente entre sí, las llamadas deben procesarse y encaminarse por un servidor de telefonía.

Por consiguiente, sigue siendo necesario un aparato de telefonía de red que sea de bajo coste, que funcione usando un protocolo de señalización sencillo y que ofrezca un amplio conjunto de características de telefonía avanzada.

Sumario de la invención

Las limitaciones y deficiencias descritas anteriormente de sistemas de telefonía convencionales y sistemas de telefonía por Internet conocidos se superan sustancialmente mediante la presente invención, en la que un objeto principal es proporcionar un sistema de comunicación de voz basado en paquetes para su uso a través de las redes de telecomunicaciones de Internet e intranet.

Otro objeto de la presente invención es proporcionar un aparato de telefonía de datos por paquetes para su uso a través de una red de datos, tal como una red de Ethernet.

Todavía otro objeto de la presente invención es proporcionar un protocolo de comunicación para su uso en un sistema de telecomunicación basado en paquetes.

Aún otro objeto de la presente invención es proporcionar una arquitectura de protocolo de Internet que soporta servicios de telefonía y otros servicios de medios continuos o de medios en flujo continuo tales como "radio por Internet" y "TV por Internet".

Aún otro objeto de la presente invención es proporcionar un aparato de telefonía de red autónomo, de bajo coste, que puede llamar directamente a otra estación de telefonía de red o llamar indirectamente a otra parte de estación de telefonía de red, tal como a través de un servidor de redirección.

Según una primera realización de la presente invención, se proporciona un aparato de telefonía de datos por paquetes de red que incluye: un controlador de red, tal como un subsistema de controlador de Ethernet, acoplado a una red de datos para proporcionar y recibir paquetes de datos a y desde la red. Un subsistema de procesamiento de señal digital está acoplado al subsistema de controlador de red y opera con un programa informático para

detectar llamadas entrantes, iniciar sesiones de llamada e implementar características de telefonía. Un subsistema de conversión de señal está acoplado al subsistema de procesamiento de señal digital para convertir información en paquetes digitales en señales analógicas y viceversa. Un subsistema de interfaz de usuario está acoplado tanto al subsistema de conversión de señal como al subsistema de procesamiento de señal digital para proporcionar control de usuario y retroalimentación al aparato. Este dispositivo de telefonía de red autónomo se denomina en el presente documento aparato de telefonía de red.

Preferiblemente, el programa informático del aparato de telefonía de red implementa el protocolo de inicio de sesión (SIP). En este caso, una dirección de SIP única está asociada con el dispositivo y se realizan la iniciación y control de sesión según el protocolo SIP, RFC 2543, marzo de 1999.

El aparato de telefonía de red implementa preferiblemente una funcionalidad de telefonía de alto nivel que incluye una característica de monitorización, reenvío de llamada, modo de audio en flujo continuo, registro de llamantes, registro de destinatarios de la llamada y similares.

Preferiblemente, el aparato de telefonía de red incluye un conjunto de circuitos de interfaz de sensor para recibir señales desde fuentes remotas, tales como sensores. Las señales recibidas desde las fuentes remotas se procesan por el aparato de telefonía de red y se envían a un destino de red apropiado.

En otro aspecto de la presente invención, se proporciona un protocolo de comunicación para su uso en un sistema de telecomunicación basado en paquetes, teniendo el protocolo de comunicación: una capa de protocolo de Ethernet; una capa de protocolo de Internet (IP) apilada encima de la capa de protocolo de Ethernet para actuar de interfaz con la capa de protocolo de Ethernet; una capa de protocolo de resolución de direcciones (ARP) apilada encima de la capa de protocolo de Ethernet para actuar de interfaz con la capa de protocolo de Ethernet y la capa de IP, y para traducir las direcciones IP en direcciones de control de acceso al medio (MAC); una capa de protocolo de datagramas de usuario (UDP) apilada encima de las capas de ARP y de IP para actuar de interfaz con las capas de ARP y de IP y para proporcionar un transporte en tiempo real de datos y controles de aplicación dentro del sistema de telecomunicación; una capa de protocolo de transporte en tiempo real (RTP) apilada encima de la capa de UDP para actuar de interfaz con la capa de UDP y para proporcionar un transporte de datos de audio en tiempo real dentro del sistema de telecomunicación; una o más capas de protocolo de control apiladas encima de la capa de UDP para actuar de interfaz con la capa de UDP y para señalar y proporcionar un registro de los datos de audio en tiempo real; y uno o más protocolos de aplicación apilados encima de la capa de RTP para actuar de interfaz con el RTP y para formatear los datos de audio en tiempo real.

En otro aspecto de la presente invención se proporciona una arquitectura de sistema de telefonía de red. El sistema incluye al menos dos dispositivos de telefonía de red, tales como el presente aparato de telefonía de red y/o un ordenador personal (PC) de uso general con un conjunto de circuitos de interfaz y un software adecuados para hacer que el PC funcione como teléfono de red. También se proporciona un servidor de redirección que está acoplado a la red de datos junto con los dispositivos de telefonía de red. En el sistema, los dispositivos de telefonía de red pueden dirigirse directamente el uno al otro para establecer una conexión de audio en tiempo real. Alternativamente, los dispositivos de telefonía de red pueden acceder al servidor de redirección con el fin de identificar, localizar e iniciar una sesión de llamada con una parte a la que se llama. El servidor de redirección también puede usarse para implementar funciones de telefonía de alto nivel, tales como reenvío de llamada, llamada entre múltiples partes, correo de voz y similares.

Objetos, características y ventajas adicionales de la invención resultarán evidentes a partir de la siguiente descripción detallada tomada junto con las figuras adjuntas que muestran realizaciones ilustrativas de la invención.

Breve descripción de los dibujos

Para una comprensión completa de la presente invención y las ventajas de la misma, se hace referencia ahora a la siguiente descripción tomada junto con los dibujos adjuntos en los que números de referencia similares indican características similares y en los que:

la figura 1 es un diagrama ilustrativo de un sistema de telecomunicaciones que presenta una red de voz conmutada por circuitos convencional acoplada de manera operativa a una red de paquetes de voz;

la figura 2 es un diagrama de bloques de un sistema de telefonía de red de datos por paquetes;

la figura 3 es un diagrama que muestra una pila de protocolos para dispositivos de telefonía que operan en el sistema de telefonía de red de datos por paquetes de la figura 2;

la figura 4 es un diagrama de bloques de una arquitectura de hardware preferida de un aparato de telefonía de red según la presente invención;

la figura 5 es un diagrama de bloques que ilustra adicionalmente el aparato de telefonía de red de la figura 4;

- la figura 6 es un mapa de memoria a modo de ejemplo para el DSP del aparato de telefonía de red de la figura 5;
- 5 la figura 7 es un diagrama de bloques de una interfaz de memoria para el DSP del aparato de telefonía de red de la figura 5;
- la figura 8 es un diagrama de bloques de una interfaz de controlador de red para el DSP del aparato de telefonía de red de la figura 5;
- 10 la figura 9 es un diagrama de bloques de una interfaz de códec para el DSP del aparato de telefonía de red de la figura 5;
- la figura 10 es un mapa de memoria a modo de ejemplo para el DSP de la figura 5 que muestra un mapeo de la interfaz de control de LCD con direcciones de memoria de DSP;
- 15 la figura 11 es un diagrama de bloques que muestra la arquitectura de software para el aparato de telefonía de red de la figura 4;
- la figura 12 es un diagrama de bloques que muestra los mecanismos de planificación del software de nivel de proceso de la figura 11;
- 20 las figuras 13A-F son tablas que ilustran definiciones de tareas a modo de ejemplo para operaciones de software de un método preferido para hacer funcionar el teléfono de red de datos por paquetes según las arquitecturas de hardware y software de las figuras 4 y 11;
- 25 la figura 14 es un diagrama de flujo de un procedimiento de salida de petición de ARP según las arquitecturas de hardware y software de las figuras 4, 11 y 13;
- la figura 15 es un diagrama de flujo de un procedimiento de entrada de petición de ARP según las arquitecturas de hardware y software de las figuras 4, 11 y 13;
- 30 la figura 16 es un diagrama que muestra las etapas de procesamiento de IP según las arquitecturas de hardware y software de las figuras 4, 11 y 13;
- 35 la figura 17 es una lista de estructuras de datos de transmisión de Ethernet a modo de ejemplo según la arquitectura de software de la figura 11;
- la figura 18 es un diagrama de flujo de datos de un procedimiento de envío de paquetes según las arquitecturas de hardware y software de las figuras 4, 11 y 13;
- 40 la figura 19 es un diagrama de flujo de datos de un procedimiento de recepción de paquetes según las arquitecturas de hardware y software de las figuras 4, 11 y 13;
- 45 las figuras 20A y 20B muestran el esquema de memoria intermedia "ping-pong" A/D y D/A usado por el software del presente aparato de telefonía de red;
- la figura 21 es un diagrama de transición de estados del proceso *Call_task* del presente aparato de telefonía de red;
- 50 la figura 22 es un diagrama que define los valores de teclado numérico para la realización preferida del teléfono de red de datos por paquetes de la figura 5;
- la figura 23 es una estructura de datos que ilustra definiciones de estado de tecla para la realización preferida del presente aparato de telefonía de red de la figura 5;
- 55 la figura 24 es un mapeo del puerto paralelo de E/S del aparato de telefonía de red de la figura 5;
- la figura 25 es una estructura de datos que define los estados de controlador de Ethernet del aparato de telefonía de red de la figura 5;
- 60 la figura 26 es una estructura de cabecera de RTP a modo de ejemplo para un procesamiento de paquetes de RTP usado en el aparato de telefonía de red de la figura 5;
- la figura 27 es una estructura de datos para su uso con una función de generación de tono del teléfono de red de datos por paquetes de la figura 5;
- 65 la figura 28 es un diagrama de sincronismo para la función de generación de tono del aparato de telefonía de red de

la figura 5;

la figura 29 es una lista de estructuras de datos usadas para procesar las peticiones o respuestas *SIP_task* según el aparato de telefonía de red de la figura 5;

la figura 30 es un diagrama de transición de estados que ilustra el aparato de telefonía de red actuando como cliente (que inicia una llamada) según la figura 5;

la figura 31 es una lista de repuestas *SIP_task* según el aparato de telefonía de red de la figura 5;

la figura 32 es un diagrama de estados que ilustra el diagrama de transición de estados de un UAS de SIP según el aparato de telefonía de red de la figura 5; y

la figura 33 es un diagrama de bloques que ilustra parte de un sistema de telefonía de red de datos por paquetes que incluye uno o más aparatos de telefonía de red según la presente invención.

Descripción detallada de la invención

La figura 1 es un diagrama de bloques que muestra un sistema de telecomunicaciones que tiene componentes de telefonía convencional y de telefonía por paquetes. Tal como se muestra en la figura 1, el sistema incluye una red 20 de voz conmutada por circuitos acoplada a una red 30 de paquetes a través de una primera pasarela 12. La figura muestra al menos tres posibles interacciones entre telefonía por Internet y un sistema de "servicio de telefonía tradicional básico" (POTS) convencional: entrega de paquetes "de extremo a extremo"; entrega de "descarga en extremo distante"; y entrega de paquetes local. Con la entrega de paquetes "de extremo a extremo", sistemas de extremo tales como ordenadores de red, teléfonos con Internet dedicados u ordenadores personales (PC) se usan para empaquetar audio y entregar paquetes de audio a uno o más sistemas de extremo similares para su reproducción. Con la entrega de "descarga en extremo distante", se usan redes de paquetes para una transmisión de voz de largo recorrido, mientras que se usan circuitos de voz conmutados por circuitos convencionales para conectar el equipo en las instalaciones del cliente (CPE), es decir, teléfonos analógicos convencionales, a las pasarelas de telefonía por paquetes. Puede usarse "descarga en extremo distante" tanto para circuitos de voz individuales como para interconexiones de PBX, y permite la derivación de servicios de larga distancia convencionales así como la interconexión de equipos de POTS con sistemas de extremo de audio basado en paquetes. Con la entrega de paquetes local, se generan datos de voz por sistemas de extremo de audio por paquetes, pero se transportan como voz conmutada por circuitos a través de instalaciones arrendadas o públicas.

La figura 2 muestra una realización preferida de un sistema 50 de telefonía de red de datos por paquetes según la presente invención. El sistema de telefonía de red de datos por paquetes incluye: una LAN 52 de Ethernet, teléfonos 54, 56 y 58 de Ethernet, una estación 60 de trabajo, un servidor 62 y una pasarela 64 de Ethernet. Los teléfonos de Ethernet son dispositivos de red que pueden adoptar la forma de dispositivos autónomos, tales como un aparato de red o un sistema de ordenador personal con periféricos de entrada y salida de audio y que pueden funcionar bajo el control de un programa informático apropiado. Con un enfoque de red de datos por paquetes de este tipo, el tráfico de datos de voz se empaqueta cerca del usuario final. El sistema de telefonía de red de datos por paquetes de la figura 2, por ejemplo, puede incluir varias docenas de casas, oficinas o apartamentos que están conectados a una pluralidad de pasarelas de Ethernet (sólo se muestra una en la figura 2), estando ubicada cada una dentro del límite de distancia de cableado CAT-3S de 328 pies desde la unidad de terminación de red. A su vez, las pasarelas pueden conectarse a través de fibra óptica al conmutador de proximidad (no mostrado), o conectarse directamente a la red de telefonía conmutada pública (PSTN) a través de las líneas 66 tal como se muestra en la figura 2. Esta arquitectura tiene la ventaja de que puede darse cabida a una mezcla de clientes de ancho de banda bajo y de ancho de banda alto sin desplegar cables adicionales. Puesto que los costes de conmutador están dominados por número de interfaces en lugar de ancho de banda, este mecanismo ofrece un ancho de banda por usuario mucho mayor (particularmente ancho de banda pico), pero los costes de conmutación son similares a las redes de telefonía de hoy en día. En la arquitectura de la figura 2, cada dispositivo de red incluye una dirección de red y cada dispositivo puede acceder directamente a cada uno de los otros dispositivos de red a través de la dirección de red. Aunque puede desearse un servidor especializado para implementar ciertas características, no es necesario para establecer una sesión de llamada, es decir, comunicaciones de datos de punto a punto entre dos o más dispositivos de red.

El uso de una LAN 52 de red de datos por paquetes es ventajoso porque es una solución relativamente económica en la que pueden usarse interfaces de PC y hardware de red convencionales. La LAN 52 de red de datos por paquetes puede hacerse funcionar a través de una variedad de medios y permite la adición fácil de más dispositivos en una LAN de múltiple acceso. La pasarela 64 puede ser un único DSP que actúa como simple módulo de voz por paquetes y que implementa un reconocimiento de DTMF para una señalización de usuario a red.

La figura 3 es un diagrama de bloques que ilustra un diagrama de pila de protocolos de red de datos por paquetes para proporcionar telefonía por Internet y otros servicios de medios continuos ("medios en flujo continuo") tales como "radio por Internet" y "TV por Internet". Tal como conocen y entienden los expertos en la técnica, un "protocolo" es

generalmente un conjunto de reglas para comunicarse entre ordenadores. Como tal, los protocolos regulan formato, sincronismo, secuenciación y control de errores. El término “pila” se refiere al software real que procesa los protocolos y, por tanto permite el uso de un conjunto o conjuntos específicos de protocolos. El diagrama mostrado en la figura 3 muestra cómo están interrelacionados los diversos protocolos según la presente invención.

La pila 80 de protocolos de la figura 3 incorpora varios protocolos en capas que incluyen: un protocolo 82 base para proporcionar información de sincronismo y formato de mensaje de Ethernet básico; un protocolo 84 de resolución de direcciones (ARP) para actuar de interfaz con el protocolo 82 base y para traducir las direcciones IP en direcciones de control de acceso al medio (MAC); una capa 86 de red de protocolo de Internet (IP) para actuar de interfaz con el protocolo 82 base; un protocolo 88 de configuración dinámica de *host* (DHCP) opcional para actuar de interfaz con el protocolo 82 base; y un protocolo 90 de datagramas de usuario (UDP) para actuar de interfaz con los protocolos ARP 84, IP 86 y DHCP 88 y para un transporte en tiempo real de datos y controles de aplicación. La pila 80 de protocolos incluye además los siguientes protocolos de aplicación específica para codificar información de habla: un protocolo 92 de protocolo de transporte en tiempo real (RTP) para el transporte de datos de audio en tiempo real, actuando el protocolo 92 RTP generalmente de interfaz con el UDP 90 y aplicaciones 94, 96 y 98 de modulación, códec de habla y control, respectivamente. Los protocolos 94 y 96 de aplicación pueden adoptar varias formas, tales como la modulación de códigos de impulsos G.711 y los protocolos de códec de habla G.723, respectivamente. Además, la capa 97 de protocolo de flujo continuo en tiempo real (RTSP) puede incluirse para proporcionar un rendimiento mejorado en aplicaciones de medios en flujo continuo. El protocolo 98 de control se usa para el inicio de sesión y la señalización y adopta preferiblemente la forma del protocolo de inicio de sesión (SIP).

Tal como se muestra en la figura 3, RTP es el protocolo preferido para transportar datos en tiempo real a través de Internet. Véase H. Schulzrinne, S. Casner, R. Frederick y V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications”, Petición de comentarios (norma propuesta, RFC 1889, Grupo de Trabajo de Ingeniería de Internet (enero de 1996). RTP es un protocolo “delgado” que proporciona soporte a aplicaciones con propiedades en tiempo real, que incluyen reconstrucción de sincronismo, detección de pérdida, seguridad e identificación de contenido. Además, RTP proporciona soporte a conferencias en tiempo real para grupos grandes dentro de una intranet, incluyendo identificación de fuente y soporte a pasarelas, tales como puentes de audio y vídeo, y traductores de multidifusión a unidifusión. RTP ofrece retroalimentación de calidad de servicio desde receptores al grupo de multidifusión así como soporte a la sincronización de diferentes flujos de medios.

En la figura 3, la pila combinada de los protocolos 88, 90 y 92 IP, UDP y RTP añade 40 bytes a cada paquete para enlaces de baja velocidad y audio muy comprimido, y 20 bytes para 20 ms de audio de 8 kb/s. Por tanto, es deseable una compresión de cabecera.

Tal como se indicó anteriormente, la pila 80 de protocolos de la figura 3 emplea preferiblemente el protocolo de inicio de sesión (SIP) para establecer intercambios multimedia con una o más partes. En lugar de usar números telefónicos, SIP usa direcciones en forma de usuario@dominio o usuario@host. Esta dirección, por ejemplo, puede ser idéntica a la dirección de correo electrónico de una persona.

SIP proporciona la funcionalidad PBX o CLASS convencional, tal como reenvío de llamada, llamada en espera, llamante M, transferencia de llamada, “llamada retenida”, “puesta en espera” y “captura de llamadas”. “Llamada retenida” permite que una llamada a través de extensión u originada por operadora a una estación de voz de una sola línea ocupada espere automáticamente a la unidad de estación a la que se ha llamado hasta que esté libre mientras que la operadora puede realizar otras llamadas. “Puesta en espera” permite a un usuario poner una llamada en espera y luego recuperar la llamada desde otra estación dentro del sistema. “Captura de llamadas” permite que las estaciones contesten llamadas a otros números de extensión dentro de un grupo de captura de llamadas especificado por el usuario. Muchas de estas características no requieren en realidad soporte de señalización en absoluto, sino que pueden implementarse por el software de sistema final. SIP está diseñado como variante de HTTP/1.1, lo que permite una fácil reutilización de características de seguridad y autenticación de HTTP, etiquetado de contenido y negociación de pago.

SIP emplea además un gestor de llamadas basado en calendario. El software de procesamiento de llamadas accede al calendario de citas personales del usuario y contesta el teléfono en consecuencia. El usuario puede definir categorías de llamantes y establecer previamente, basándose en la entrada del calendario, si y dónde se reenviarán sus llamadas. La información facilitada al llamante, si las llamadas no se reenvían, puede variar, por ejemplo, desde “no está disponible actualmente” a “John Smith está en una reunión hasta las 3 p.m. en la sala 5621 con Jane Doe”, dependiendo de la identidad del llamante. El gestor de llamadas también puede integrarse con un lenguaje de procesamiento de llamadas, un lenguaje de escritura de comandos basado en estados que permite construir sistemas de correo de voz o sistemas de gestión de llamadas automáticos en unas pocas líneas de código. El gestor de llamadas también gestiona la traducción entre llamadas de RDSI y llamadas de telefonía por Internet.

La figura 4 es un diagrama de bloques de hardware de alto nivel que muestra una realización preferida de un teléfono 100 de red de datos por paquetes según la presente invención. Tal como resultará evidente a lo largo de toda esta descripción, el dispositivo 100 es un producto de interfaz de coste relativamente bajo para colocar voz y datos en una red de datos por paquetes, tal como LAN de Ethernet, intranet e Internet. Por tanto, el dispositivo 100

se denominará generalmente aparato de red para reflejar la amplia aplicabilidad de este dispositivo autónomo.

El aparato 100 de red proporciona comunicaciones de audio y vídeo a través de una red de área local (LAN), Internet u otra red de Ethernet, e incluye generalmente: un subsistema 110 de controlador de red (por ejemplo, Ethernet); un subsistema 120 de procesamiento de señal digital; un subsistema 130 de conversión de señal; y un subsistema 160 de interfaz de usuario acoplado tanto al subsistema 130 de conversión de señal como al subsistema 120 de procesamiento de señal digital. El teléfono 100 incluye además una fuente de alimentación, una ROM 142 y una RAM 152. El subsistema de interfaz de usuario puede incluir un altavoz 161, un micrófono 162 y otros controles 169 de usuario tal como se comenta más adelante y con referencia a la figura 5. Un conjunto 135 de circuitos de interfaz para funciones de adquisición de datos y de control también puede acoplarse al subsistema 130 de conversión de señal. Alternativamente, tal conjunto de circuitos de E/S puede acoplarse directamente al DSP 120.

El subsistema 110 de controlador de red está interpuesto entre el DSP 120 y la red de datos externa y como tal proporciona y recibe paquetes de datos a y desde la red de datos (Ethernet). El subsistema 110 de controlador de Ethernet también ordena al subsistema 120 de procesamiento digital que acepte datos recibidos desde o proporcione datos a la red de Ethernet. Además, el subsistema de controlador de red puede actuar como controlador de acceso inicial rechazando y descartando paquetes de datos corrompidos o no deseados recibidos desde la red de Ethernet.

La figura 5 es un diagrama de bloques que ilustra el presente aparato de red en mayor detalle. Tal como se muestra en la figura 5, una realización preferida del subsistema 110 de controlador de red incluye un controlador 112 de Ethernet, un filtro 114 de servicio (transformador 10Base-T) y al menos un enchufe 116 RJ-45. Entre otras cosas, el subsistema 110 de controlador de red realiza las siguientes funciones: actuar de interfaz para el aparato de red con la red de Ethernet; enviar y recibir paquetes de Ethernet; informar al subsistema 120 de DSP para que acepte los datos cuando hay datos disponibles desde la Ethernet; recibir los paquetes desde el subsistema 120 de DSP y enviarlos a la Ethernet; y rechazar y descartar paquetes no deseados desde la Ethernet.

Tal como se muestra en la figura 5, el controlador 112 de Ethernet es preferiblemente el control de acceso al medio para Ethernet (MACE) AM79C940 disponible de Advanced Micro Device (AMD). El dispositivo de MACE es un periférico basado en registro esclavo. Todas las transferencias a y desde el sistema se realizan usando una memoria sencilla u órdenes de lectura y escritura de E/S. Junto con un motor de DMA definido por el usuario, el chip de MACE proporciona una interfaz de IEEE 802.3 adaptada a una aplicación específica.

FIFO de transmisión y recepción individuales disminuyen la latencia del sistema y soportan las siguientes características: retransmisión automática sin recarga de FIFO; eliminación de recepción automática y espaciado de transmisión; rechazo automático de paquetes pequeños; borrado automático de tramas de colisión; acceso de lectura/escritura de FIFO directo para una interfaz sencilla con controladores de DMA o procesadores de E/S; alineación de bytes arbitrarios e interfaz de memoria pequeña/grande/mediana soportada y velocidad de reloj de sistema de 5 MHz-25 MHz.

Haciendo referencia de nuevo a la figura 5, el subsistema 120 de procesamiento de señal digital incluye un procesador 122 de señal digital (DSP) y circuitos lógicos relacionados, que incluyen una memoria 142 de sólo lectura (ROM), una memoria 52 de acceso aleatorio (RAM) y un dispositivo 124 lógico programable borrable (EPLD). El subsistema 120 de procesamiento de señal digital proporciona las siguientes funciones: procesamiento de señal digital, tal como compresión de habla; generación de tono de progreso de llamada y generación de señal de timbre; lógica de "interconexión" general para interconectar DSP, memoria y dispositivos de E/S; procesamiento de protocolo de red; control de flujo de llamadas e implementación de máquina de estados finitos; detección y decodificación de actividad de teclado numérico; y control de presentación visual.

Tal como se muestra en la figura 5, el DSP 122 usado en una realización preferida del aparato de red puede ser cualquier DSP comercialmente disponible adecuado, tal como TMS320C32 de Texas Instruments. El DSP TMS320C32 tiene las siguientes características: operaciones de multiplicación paralela y de unidad de lógica aritmética (ALU) en datos de número entero o coma flotante en un único ciclo; archivo de registros de uso general; almacenamiento en caché de programa; unidades aritméticas de registro auxiliar (ARAU) dedicadas; memorias de doble acceso internas (512 palabras dobles); dos canales de acceso directo a memoria (DMA); un puerto serie; dos temporizadores; un puerto de memoria externa y una estructura de múltiples interrupciones.

Además, el DSP TMS320C32 incluye recursos de cuatro interrupciones externas y seis interrupciones internas. La interrupción externa puede activarse directamente por los pin externos. La interrupción interna puede activarse programando los periféricos individuales, tales como puerto serie, controlador de DMA y temporizadores. Además, todas estas fuentes de interrupción pueden programarse como interrupción de canal de DMA a través de un registro habilitado por CPU/DMA, IE. El DSP TMS320C32 también incluye un cargador de arranque flexible que habilita el programa de control principal para el aparato de red cargado automáticamente desde uno de tres espacios de memoria externa diferentes o el puerto serie, lo que sea apropiado según se determine por la actividad de las interrupciones externas de INTO a INT3 cuando se inicia el DSP 122, tal como en el encendido.

El DSP 122 está configurado generalmente para incluir las siguientes asignaciones de recursos. Interrupciones externas incluyen: INT0: indicación de “arranque de sistema desde 0x1000”. Cuando el sistema se enciende e int0 está activo, el DSP arrancará el programa desde el espacio de memoria externa 0x1000; INT1: señal de interrupción externa DMA0, usada para recibir paquetes desde el controlador 112 de red; INT2: señal de interrupción externa DMA1, usada para enviar paquetes al controlador 112 de red; INT3: interrupción de estado de paquete y de mensaje de error AM79C940. Un mapa de memoria de DSP de muestra para su uso en una realización del presente aparato de red se muestra en la figura 6.

Haciendo referencia de nuevo a la figura 5, el presente aparato de red tiene el subsistema 160 de interfaz de usuario que incluye: un codificador 166 de claves, una pantalla 164 de cristal líquido (LCD) y un aparato 163 de teléfono, que incluye un teclado 165 numérico, un micrófono 162 y un altavoz 161. Los componentes del subsistema 160 de interfaz de usuario permiten una interacción de usuario con el aparato de red proporcionando las siguientes funciones: interfaz de usuario para entrada (teclado numérico) y salida (LCD); interfaz de voz; salida de alerta de timbre a través de altavoz y comunicación alternativa de aparato de teléfono o manos libres (micrófono y altavoz). A través de esta interfaz 160 se introducen órdenes del usuario y se envía y recibe audio al usuario.

Además, la LCD puede tener botones adyacentes a la pantalla, tales como al lado y debajo. La función de estos botones puede actuar como “teclas programables” cuya función depende del estado actual del sistema. Por ejemplo, cuando no se contestan llamadas, la pantalla puede mostrar una rápida de marcación lista y la hora del día. Además, después de haber dejado sin contestar llamadas o haberse reenviado al correo de voz, la pantalla puede mostrar una lista de llamadas recibidas. Durante la llamada, se presenta visualmente cualquier otra llamada entrante, permitiendo al abonado conmutar entre llamadas o asociar la llamada a la llamada existente.

Alternativamente, la interfaz 160 de usuario del presente aparato 100 de red puede estar configurada con una pequeña pantalla táctil (no mostrada) para sustituir o complementar la pantalla LCD y los botones. La pantalla táctil, que presenta visualmente de manera gráfica funciones y operaciones disponibles y responde al contacto por parte del usuario en la pantalla, proporciona una interfaz de usuario mejorada, tal como para la introducción de direcciones de red alfanuméricas y otras operaciones de telefonía.

La figura 5 también muestra el sistema 130 de procesamiento de señal, que incluye codificador y decodificador de PCM que realiza una conversión analógico a digital (A/D) y digital a analógico (D/A), y un amplificador de audio 134 acoplado al aparato de teléfono y los correspondientes altavoz 161 y micrófono 162. También se proporciona una fuente de alimentación para proporcionar niveles de tensión de 5V positiva y negativa desde un único adaptador de fuente de alimentación de CA o CC (“enchufe de pared”). En la realización preferida de la figura 5, se requieren niveles de tensión negativos por la LCD 164 y el códec 132 de PCM.

La figura 7 es un diagrama de bloques que ilustra una interfaz 700 de memoria adecuada para su uso en el aparato de red de la figura 5. La interfaz 700 de memoria incluye módulos 142 y 152 de memoria externa, que a su vez incluyen 128 Kbytes de memoria 142 de sólo lectura (ROM) para almacenamiento de programas y al menos 32 Kbytes de memoria 702, 704, 706 y 708 de acceso aleatorio (RAM) estática de palabras dobles (32 bits). Debido a la velocidad relativamente lenta de la ROM 142, es preferible que el aparato de red inicialice el programa principal desde la ROM y almacene este programa en la RAM relativamente rápida durante la ejecución en el tiempo de ejecución.

La figura 8 es un diagrama de bloques que muestra una interfaz a modo de ejemplo entre el DSP 122 y el controlador 124 de Ethernet según una realización preferida de la presente invención. Los 32 registros del controlador 124 de Ethernet se mapean en memoria en el espacio de memoria 0x810000 del DSP 122 tal como se muestra en la figura 6. Preferiblemente, los primeros dos registros están recibiendo y transmitiendo colas “primero en entrar, primero en salir” (FIFO). El DSP 122 intercambia los datos con el controlador 124 de Ethernet a través de un bus 802 de datos de 16 bits.

La figura 9 es un diagrama esquemático que ilustra una interfaz entre el DSP 122 y el códec 132 de PCM según una realización preferida de la presente invención. Tal como se muestra en la figura 9, el DSP 122 se conecta al códec 132 de PCM a través de un puerto 902 serie interno. El puerto serie en el DSP 122 es un puerto serie bidireccional independiente.

Tal como se muestra en la figura 5, el DSP 122 también está acoplado de manera operativa a la LCD 164. La interfaz de control de LCD se mapea con las direcciones de DSP mostradas en la figura 10. En una realización de la presente invención, la LCD 164 es una LCD de 120 x 32 píxeles tal como la LCD MGLS-12032AD, fabricada por Vazitronics. Puesto que la velocidad de acceso de la LCD es generalmente baja, los datos presentados visualmente por la LCD pueden mapearse con el espacio de memoria STRBO (1X1000) del DSP 122, que es el mismo espacio de memoria que el espacio de memoria de ROM. Preferiblemente, la lógica de sincronismo de LCD es la misma que la lógica de sincronismo para el DSP 122. Sin embargo, cuando la LCD está compuesta por una mitad izquierda y una mitad derecha, tal como en la MGLS-12032, es necesario controlar y programar ambas mitades de la LCD cuando se presenta visualmente un mensaje de línea completa.

La figura 11 es un diagrama de bloques que muestra la arquitectura de software para el presente aparato de red. Tal como se muestra en la figura 11, la arquitectura de procesamiento para el presente aparato de red se organiza generalmente en tres niveles: el nivel 1110 de ISR (rutina de servicio de interrupción); el nivel 1120 de proceso o sistema operativo y el nivel 1130 de tarea o aplicación. Una lista a modo de ejemplo de las funciones y tareas que pueden realizarse en cada uno de los niveles de software se proporciona en la figura 13.

El nivel inferior, el nivel 1110 de ISR, incluye gestores de interrupción y funciones de interfaz de E/S. El nivel 1110 de ISR sirve como interfaz entre el nivel 1120 de proceso y el hardware de aparato de red mostrado en las figuras 4 y 5.

Por encima del nivel 1110 de ISR está el nivel 1120 de proceso, o sistema operativo, que es preferiblemente un micronúcleo multitarea en tiempo real, tal como el micronúcleo en tiempo real incrustado CRTX de StarCom. Generalmente, el software 1120 de nivel de proceso (micronúcleo) realiza la gestión de memoria, gestión de procesos y tareas y funciones de gestión de disco. En una realización preferida de la presente invención tal como se muestra en la figura 12, el micronúcleo soporta tres mecanismos de planificación: un gestor 1222 de etiquetas de eventos en tiempo real; un gestor 1224 de tareas retardadas y un gestor 1226 de planificación. El micronúcleo tiene tres colas separadas para los tres mecanismos diferentes anteriores, respectivamente.

El gestor 1222 de etiquetas de eventos en tiempo real se usa para activar la ejecución de eventos en tiempo real a modo de etiquetas de establecimiento. Si una etiqueta se establece en una condición "ON", la tarea asociada con la etiqueta se ejecuta inmediatamente. Por ejemplo, una rutina de servicio de interrupción establecería una etiqueta particular cuando se produce un determinado evento. Los eventos de etiquetas se introducen en una cola de etiquetas con una dirección de tarea asociada.

El gestor 1224 de tareas retardadas es responsable de eventos sincronizados. Una tarea sincronizada, tal como una tarea a prueba de fallos o "guardián", puede ejecutarse tras un determinado retardo de tiempo. Si no se produce un determinado evento dentro de una determinada trama de tiempo, el temporizador activa la tarea haciendo que se ejecute. Otro ejemplo es la ejecución repetida de una tarea controlada por un temporizador periódico. En una realización a modo de ejemplo, hay 10 entradas de temporizador. Cada temporizador está cargado con un contador de tics y luego se disminuye a cada tic del temporizador a partir del temporizador de intervalo del hardware. Cuando el recuento llega a cero, la tarea asociada con el temporizador se planifica en la cola de tareas. El gestor 1226 de planificación explora la cola de planificación de tareas buscando tareas planificadas. Al descubrir una entrada en la cola, el control pasa a una tarea planificada.

Las figuras 13a-f son tablas que enumeran tareas y funciones de software a modo de ejemplo que pueden ser parte del software de nivel de tarea (figuras 13a-c), software de nivel de proceso (figura 13d) y software de nivel de ISR (figuras 13e-f). Para los fines de la presente invención, los términos "tarea" y "función" tal como se denominan con respecto a la arquitectura de software se consideran sinónimos. Sin embargo, las "tareas" se ejecutan generalmente por el gestor 1226 de planificación, mientras que las "funciones" se llaman generalmente por las tareas o por otras funciones. Las tareas de aplicación, tales como las tareas de procesamiento de llamada (*Call_task*) y de procesamiento de IP (*IP_Send_task* y *ERcv_task*, etc.), se planifican por el software 1120 de nivel de proceso. La ejecución de tales tareas es un resultado de una planificación previa por una ISR, otra tarea, o por la propia tarea actual.

Las figuras 13 A-F ilustran definiciones de función y procedimiento a modo de ejemplo llamadas en una operación motivada por eventos realizada por el presente software de teléfono de red de datos por paquetes de la figura 11. Las funciones, que se llaman cuando se producen diversos eventos, habilitan la operación del teléfono/sistema de red de datos por paquetes e incluyen operaciones ordinarias tales como: iniciar el teléfono/sistema de red de datos por paquetes; procesar los datos de ARP; codificar datos de voz; procesar datos de mensajes; procesar datos de IP; decodificar datos de voz; transferir datos analógicos y digitales a y desde correspondientes memorias intermedias; y realizar funciones de "guardián".

La inicialización del aparato de teléfono de red de datos por paquetes incluye las etapas de inicialización de hardware y planificación de tareas. Después del encendido, el DSP 122 transferirá automáticamente el programa principal desde la ROM 142 hasta la RAM 152 (operación de arranque). La inicialización de hardware se produce de una manera tradicional, incluyendo las etapas de: inicializar el puntero de pila, registro de control de interfaz de bus externo, registro de control global del DSP, vector de interrupción para la ISR y similares.

Una vez finalizadas la inicialización de hardware y la planificación de tareas preliminares, se devuelve el control de procesamiento al nivel 1120 de proceso (micronúcleo). El micronúcleo 1120 CRTX y las tareas planificadas controlan entonces el procesamiento posterior.

Haciendo referencia a la figura 13a, el software de nivel de tarea del presente aparato de red incluye un procesamiento de protocolo de resolución de direcciones (ARP). El ARP es un protocolo TCP/IP conocido usado para convertir una dirección IP en una dirección física (denominada dirección de control de enlace de datos (DLC)), tal como una dirección de Ethernet. Un ordenador *host* que desea obtener una dirección física difunde una petición

de ARP en la red TCP/IP. El ordenador *host* en la red que tiene la dirección IP en la petición contesta entonces con su dirección de hardware físico.

La figura 14 es un diagrama de flujo que ilustra un procedimiento 1400 de salida de petición de ARP, *ARP_out()*. Tal como se ilustra en la figura 13B, *ARP_out()* es un componente del software de nivel de tarea que recibe una dirección IP que debe resolverse, y emite una correspondiente dirección de MAC. Cuando comienza una petición de ARP (etapa 1402) la función *ARP_out()* comprueba en primer lugar la dirección IP solicitada en una tabla en caché de ARP local, *arptable* (etapa 1404). Si la correspondiente entrada SE RESUELVE en la etapa 1406, entonces *ARP_out()* copia la dirección de MAC desde *arptable* al parámetro solicitado y devuelve una etiqueta de estado ARPOK (etapa 1408). De lo contrario, el procedimiento asignará una entrada en *arptable* y planifica una petición de ARP (etapa 1410). Tal como se muestra adicionalmente por la etapa 1410, una dirección de MAC, es decir, "gestión", de la *arptable* se devuelve al programa principal (*c_int00()*). Según la gestión, el software comprueba entonces el correspondiente *ae_state* de la entrada.

La figura 15 es un diagrama de flujo de un procedimiento 1500 de entrada de petición de ARP a modo de ejemplo, *ARP_In_task()*, que es un componente del software de nivel de tarea enumerado en la figura 13A. La *ARP_In_task* recibe un paquete de ARP, y o bien modifica la *arptable* o bien pone en cola una contestación de ARP si el paquete entrante es una petición de ARP. Cuando se recibe un paquete de ARP (etapa 1502) el software comprobará si los tipos de protocolo y hardware de ARP del paquete coinciden (etapa 1504). Si los tipos no coinciden, se devuelve el control al programa principal (etapa 1506). Si uno o los dos tipos coinciden, entonces el software comprueba si el *host* de destino es el presente *host* (etapa 1510). Si el *host* de destino no es el presente *host*, entonces se devuelve el control al programa principal (etapa 1508).

Tal como se muestra adicionalmente en la figura 15, si el *host* de destino es el *host* actual, entonces la *ARP_In_task* comprueba a continuación la tabla de ARP para determinar si hay una entrada de ARP correspondiente para el paquete entrante (etapa 1512). Si se halla una entrada (etapa 1514), entonces la nueva dirección de MAC se copia a la entrada existente y modifica el "periodo de vida" (TTL) de la entrada a un nuevo valor (etapa 1516). Los expertos en la técnica entienden el TTL como un campo en el protocolo de Internet (IP) que especifica cuántos saltos más puede dar un paquete antes de descartarse o devolverse al remitente. Sin embargo, si no se haya tal entrada de MAC según la etapa 1513, entonces la *ARP_In_task* añade una nueva entrada de MAC en la tabla de ARP (etapa 1518). Si la entrada de MAC está en un estado PENDIENTE (etapa 1520), entonces se cambia a un estado RESUELTO y la dirección de MAC se copia a la entrada objetivo (etapa 1522). Si el paquete de ARP entrante es una petición de ARP desde otro *host*, se envía un paquete de contestación de ARP poniendo en cola la *IP_Send_task*, etapas 1524 y 1526. El control se devuelve entonces al programa principal (etapa 1528).

Además de los procesos de entrada y salida de ARP, el procesamiento de ARP en el nivel de tarea incluye una *ARPTimer_task()*, que es una tarea en bucle retardada usada para mantener la tabla de entradas de ARP *arpenry*. De manera nominal, la *ARPTimer_task()* se genera una vez por segundo. El propósito principal de la *ARPTimer_task()* es disminuir el "periodo de vida" (TTL) de la entrada de ARP y reenviar la petición de ARP durante el estado pendiente en caso de que la petición de ARP previa se pierda.

El procesamiento de nivel de tarea también puede incluir operaciones de procesamiento asociadas con la codificación y decodificación de paquetes de audio. La *Codec_task* incluye generalmente una función de *SpeechEncode()*, que codifica datos de habla de la memoria intermedia *ADBuf* a la *EncodeBuf* según el algoritmo indicado por el parámetro "tipo". Los datos codificados se emiten entonces a través de la cola *IP_Send_task*, con el conjunto de parámetros "RTP".

Las operaciones de nivel de tarea también pueden incluir procesamiento de protocolo de Internet (IP). Las operaciones de procesamiento de IP general están ilustradas en el diagrama de bloques de la figura 16. Tal como se muestra en la figura 16, el procesamiento de IP incluye las etapas de: transmitir y recibir paquetes de Ethernet, etapa 1602; multiplexación y demultiplexación de paquetes de IP, etapa 1604; y empaquetar y desempaquetar paquetes de Ethernet, de protocolo de Internet (IP), de protocolo de datagramas de usuario (UDP), de protocolo de transporte en tiempo real (RTP) y de protocolo de resolución de direcciones (ARP), etapa 1606.

Según la etapa 1602 de la figura 16, puede realizarse una transmisión de paquetes de Ethernet usando canales de acceso directo a memoria (DMA) del controlador 112 de Ethernet. DMA es una técnica para transferir datos desde una memoria principal a un dispositivo sin pasarlos a través de la CPU. Puesto que los canales de DMA pueden transferir datos a y desde dispositivos de una manera mucho más rápida que con medios convencionales, el uso de canales de DMA es especialmente útil en aplicaciones en tiempo real, tal como el presente sistema de telefonía de red.

El controlador 110 de red soporta preferiblemente una pluralidad de canales de DMA, tales como el canal DMA1 del controlador 112 de Ethernet que puede usarse para transmisión de paquetes. Cuando un paquete de Ethernet está listo para su transmisión, se llama la función *DMA1()*, una función de nivel de ISR, estableciendo la dirección de origen (memoria intermedia de paquetes de Ethernet, *ESend*), la dirección de destino (FIFO de transmisión del controlador de Ethernet) y un contador (la longitud de paquete). Ejemplos de estructura de datos de transmisión de

Ethernet se proporcionan en la figura 17. La función *DMA1()* inicia entonces el canal DMA1. Cuando el contador llega a cero, el DMA1 se detiene y espera la siguiente llamada.

La figura 18 es un diagrama de bloques que muestra el flujo de datos entre una memoria 1802 intermedia de entrada de datos, una memoria 1804 intermedia de UDP y una tabla 1806 de ARP en la interfaz de Ethernet (FIFO de transmisión de Ethernet) del controlador 112 de red de Ethernet. Tal como se muestra adicionalmente en la figura 18, los datos desde la memoria 1802 intermedia de entrada de datos, la memoria 1804 intermedia de UDP y la tabla 1806 de ARP se envían a una cola 1810 de salida de IP, y están dispuestos para indicar el tipo de protocolo, el puntero de origen y la longitud de datos. En lugar de poner en cola los datos de envío, la *IP_Send_task* se pone en cola por un software 1120 de nivel de proceso (micronúcleo). Los tipos de protocolo soportados por la *IP_Send_task* incluyen generalmente UDP, RTP, ARP_REQUEST, ARP_REPLY. *IP_Send_task* se usa para transmisión de paquetes y empaquetado de Ethernet. Preferiblemente, *IP_Send_task* se planifica por otras tareas o funciones tales como *SIP_task*, *ARP_Out()*, *SpeechEncode()*, etc. Una vez que se ejecuta la *IP_Send_task*, ésta comprueba el tipo de protocolo de los datos. Esta tarea encapsula entonces los datos de salida en el correspondiente paquete de Ethernet en la memoria intermedia *ESend*. Finalmente, el paquete se emite a través del canal DMA asignado (DMA1).

La figura 19 es un diagrama de flujo de datos que ilustra adicionalmente operaciones de recepción y demultiplexación de paquetes. La demultiplexación se realiza planificando diferentes tareas para diferentes protocolos en la *ERcv_task*. De acuerdo adicionalmente con la etapa 1602 de la figura 16, los paquetes de Ethernet se reciben en la memoria FIFO de datos de recepción (etapa 1902) y se procesan adicionalmente por un controlador de canal DMA0 (etapa 1904). Puesto que el DSP 122 no sabe cuándo llegarán los paquetes, el canal DMA0 está activo todo el tiempo (es decir, no se detiene incluso aunque el contador llegue a cero). Cuando llega un paquete, el canal DMA0 lo copiará automáticamente desde la FIFO de recepción del controlador de Ethernet a la memoria intermedia de recepción de Ethernet, *Ercv* (etapa 1906). El canal DMA0 se detiene cuando no hay datos disponibles en la FIFO.

ERcv_task es una tarea de activación de etiqueta para un desempaqueado de paquetes de Ethernet y una demultiplexación de paquetes de IP (etapa 1908). La *ERcv_task* funciona tal como sigue: en primer lugar, se llama una función *PacketCheck()* para comprobar el paquete entrante. La *PacketCheck()* devolverá el tipo de protocolo del paquete, o será NULO si el paquete es no válido. En segundo lugar, dependiendo del tipo de protocolo devuelto, la *ERcv_task* activará las diferentes tareas para procesar el paquete recibido, *RTP_In_task* para un paquete de "RTP" (etapa 1910), *ARP_in_task* para un paquete de "ARP" (etapa 1912) o tareas de procesamiento de UDP (etapa 1912) para paquetes de UDP, por ejemplo.

Haciendo referencia a la figura 13C, *SpeechDecode()* es una función de decodificación de voz asociada con el procesamiento de RTP de la etapa 1910. En primer lugar, una tarea de *SpeechDecode()* comprueba si hay datos disponibles en la memoria intermedia de decodificación, *DecodeBuf*. Si hay datos disponibles, por ejemplo, *RcvFlag* está ESTABLECIDA, luego *SpeechDecode()* los decodifica según el tipo de datos de datos de recepción, PCM (G.711), G.723, G.729, por ejemplo. Los datos decodificados se envían a la memoria intermedia D/A, *DABuf*.

La rutina de interrupción de A/D y D/A puede activarse mediante una fuente de interrupción interna, por ejemplo, *Rint0()*. Preferiblemente, la rutina de interrupción A/D y D/A se activa por una frecuencia de muestreo de 8 kHz proporcionada por el DSP. Puesto que esta rutina se denomina frecuentemente, *Rint0()* se escribe preferiblemente en un lenguaje de ensamblaje. Las etapas realizadas por *Rint0()* incluyen las etapas de: leer una muestra D/A desde una memoria intermedia D/A, *DABuf*, enviar la muestra al puerto serie D/A; obtener una muestra desde el puerto serie A/D; guardar la muestra A/D a una memoria intermedia A/D, *ADBuf*; y aumentar los punteros de memoria intermedia A/D y D/A, *ADPnt* y *DAPnt*, por uno.

Las figuras 20A y 20B son diagramas de bloques que muestran un esquema de memoria intermedia "ping-pong" A/D y D/A usado por el software de la presente invención. Además, si el valor de puntero A/D actual (*ADPnt*) supera un umbral de memoria intermedia (*ADTh*) predeterminado entonces se establece una etiqueta en la cola de tareas de etiqueta que indica que se requiere un servicio.

Las memorias intermedias A/D y D/A pueden dividirse en dos partes, memoria 2002a intermedia superior y memoria 2002b intermedia inferior, respectivamente. Ambas memorias intermedias pueden diseñarse como memorias intermedias circulares. De esta manera, cuando el puntero actual alcanza la parte inferior de la memoria intermedia, vuelve al principio. Sin embargo, desde el punto de vista del codificador y decodificador, se usa como esquema de memoria intermedia ping-pong de dos tramas (definido como trama superior y trama inferior). La operación de este proceso se muestra en las figuras 20A y 20B. Para la conversión A/D, cuando se llena la parte superior (o parte inferior), los datos en la memoria intermedia superior (o inferior) pasarán por un conmutador 2004 ping pong y se copiarán a la memoria intermedia de codificación de habla, *EncodeBuf*, 2006. Para una conversión D/A, si la memoria intermedia superior (o inferior) se completa, se copiará una nueva trama de datos desde la memoria intermedia de decodificación de habla, *DecodeBuf*, 2010 a la memoria intermedia superior 2008a (o inferior 2008b). Este mecanismo garantiza que mientras que el algoritmo de codificación (o decodificación) lee (escribe) de una parte de la memoria intermedia, la ISR de muestreo A/D (o D/A) puede escribir (leer) la otra parte de la memoria

intermedia sin conflicto.

La figura 21 es un diagrama de transición de estado de una subrutina de *Call_task* usada en una realización a modo de ejemplo del presente aparato de red. *Call_task* es una tarea en bucle que gestiona el procedimiento de llamada. Tal como se muestra en la figura 21, el estado 2102 "inactivo" se produce cuando no se realiza ninguna llamada y no hay ninguna llamada entrante. Cuando existe esta condición, la *Call_task* reinicia el bucle en el estado 2102 "inactivo". El estado 2104 "tono de marcación" existe cuando el estado receptor está DESCOLGADO, o el estado de aparato de teléfono indica MANOS LIBRES, y por tanto el estado de *Call_task* cambiará del estado 2102 "inactivo" al estado 2104 "tono de marcación" cuando existe una condición DESCOLGADO o MANOS LIBRES. Estos estados se introducen generalmente por una entrada por un usuario a través de los controles 160 de usuario indicando que va a iniciarse una llamada. Cuando el estado de *Call_task* está en un estado 2104 "tono de marcación", la *Codec_task* se configurará como "modo de tono, tono de marcación" y se envía un tono de marcación a los componentes de aparato de teléfono de la interfaz 160 de usuario.

Haciendo referencia de nuevo a la figura 21, mientras se encuentra en el estado 2104 "Tono de marcación", si se mantiene presionada cualquier tecla de dígitos ('0 ... '9', '*' y '#') o el botón de remarcación, el estado de llamada cambia desde el estado 2104 "Tono de marcación" al estado 2106 "conseguir dígito". En el estado 2106 "conseguir dígito", el tono de marcación se detiene en el aparato de teléfono.

Después de que se ha introducido el número del destinatario y se ha presionado un botón INTRODUCIR por el usuario para indicar que la marcación está completa, la *Call_task* comprobará si la entrada es válida. Si el número es válido, se crea una entrada de llamada por una función *CreateSipCall()* y la *Call_task* se dirigirá a un estado 2108 "SIP". Por el contrario, si el número de entrada es no válido, el número se solicita de nuevo y el estado permanece en el estado 2106 "conseguir dígito".

Mientras se espera un procesamiento de *SIP_task*, pueden tomarse varias decisiones dependiendo del estado 2108 "SIP". El estado 2108 "SIP" es un *SIP_status*, variable global que se modifica por la *SIP_task* según su transición de estado. Si el estado 2108 "SIP" cambia a *SIP_Ring*, la *Call_task* cambiará al estado 2114 "retorno de llamada" y la *Codec_task* se configurará como modo "modo de tono, retorno de llamada". Cuando la *Codec_task* está en el modo "modo de tono, retorno de llamada", se envía un tono de retorno de llamada al aparato de teléfono.

A partir del estado 2108 "SIP", si el estado 2108 "SIP" cambia a *SIP_busy*, la *Call_task* y por tanto la llamada cambiarán al estado 2120 "tono de ocupado" y el tono de ocupado se reproducirá en el aparato de teléfono. Si el estado 2108 "SIP" cambia a *SIP_Refused*, se presentarán visualmente mensajes apropiados en la pantalla LCD en relación con el estado *SIP_Refused*.

A partir del estado 2118 "retorno de llamada", si el estado "SIP" se vuelve *SIP_Connected*, el estado *Call_task* cambia al estado 2116 "hablar". Cuando el estado *Call_task* está en el estado 2116 "hablar", la *Codec_task* se configurará como modo *SpeechEncode* y *SpeechDecode*.

Para llamadas entrantes, mientras está en el estado 2102 "inactivo", si el estado 2108 "SIP" es *SIP_Invite*, el estado *Call_task* cambia al estado 2114 "timbre" y la *Codec_task* se configurará como "modo de tono, tono de timbre". Cuando la *Codec_task* está configurada como "modo de tono, tono de timbre", se reproducirá un tono de timbre en el altavoz. Después de que el estado SIP se vuelve *SIP_Connected*, el estado *Call_task* cambiará al estado 2116 "hablar". Por el contrario, si el estado SIP se vuelve *SIP_Cancel*, lo que sucede si el llamante abandona la llamada, el estado *Call_task* vuelve al estado 2108 "inactivo".

Mientras se encuentra en el estado 2102 "inactivo", si se presiona el botón INTRODUCIR, la *Call_task* llama a la *Setting_task*. Cuando el programa de ajuste de parámetros está terminado, volverá a la *Call_task*.

Durante la ejecución de *Call_task*, si el estado de cuelgue indica que el receptor está COLGADO, o se halla un error de sistema, la *Call_task* cambia al estado 2102 "inactivo", independientemente de cuál sea el estado previo (excepto el estado 2114 "timbre").

En la realización preferida del aparato de red tal como se muestra en la figura 5, el teclado numérico del teléfono tiene 17 teclas para proporcionar entradas y órdenes de usuario. El teclado numérico del teléfono incluye 10 teclas de dígitos, dos teclas especiales y cinco teclas de función se definen tal como se muestra en la figura 22.

La *Key_task* es una tarea retardada de bucle que se ejecuta periódicamente, tal como cada 0,1 segundos. Cuando se inicia, la *Key_task* llama en primer lugar a la función *key()*. Si el valor de retorno no es "-1", significa que se ha presionado una tecla. Entonces, la función *KeyMap()* mapea la palabra clave binaria de entrada con la palabra clave ASCII. La *Key_task* establece entonces el correspondiente elemento de la estructura de *FuncKey*. Si el sistema está listo para aceptar la entrada de tecla (se indica la *KeyRegEnable*), la palabra de tecla de entrada se almacena en la *KeyBuf*.

Además, *Key_task* soporta preferiblemente cuatro modos de entrada diferentes: modo de entrada de dígitos, modo

de entrada de dirección IP, modo de entrada de alfabeto y modo de entrada de dirección de lista. Puede realizarse una conmutación entre los cuatro modos presionando el botón INTRODUCIR antes de marcar cualquier número o alfabeto cuando se levanta el aparato de teléfono y se escucha un tono de marcación. Después de que la entrada está completa y se presiona el botón INTRODUCIR, los números de entrada se transferirán a la tarea actual (*Call_task* o *Setting_task*) mediante una canalización de mensaje. Si se presiona la tecla llamada, la tarea copiará la entrada previa desde la memoria intermedia de copia de seguridad, *KeyBackup*, a la *KeyBuf*. Entonces, los datos se transferirán a la *Call_task*.

El sistema operativo del presente aparato de red soporta preferiblemente un esquema de planificación de tarea retardada. La tarea retardada es similar a la función *sleep()* en UNIX. Sin embargo, una tarea retardada también puede ser una ejecución de tareas persistente desde un temporizador periódico cuando se establece la etiqueta de repetición de la tarea. Para tareas retardadas, el software 1120 de nivel de proceso requiere un temporizador de intervalo para proporcionar un tic de sistema. El sistema de la figura 5 usa el temporizador1 de TMS320C32, TCLK1, como base de temporizador de sistema.

La *Clock_task* es una tarea retardada en bucle que realiza funciones de calendario y de reloj en tiempo real. Sirve como reloj general para calcular y presentar visualmente el tiempo actual, incluyendo la hora, minuto y segundo. Cuando se conecta una llamada, puede presentar visualmente la duración de llamada. Cuando el teléfono está colgado, también puede presentarse visualmente el año, mes y fecha actuales en la LCD.

Haciendo de nuevo referencia a la figura 11, el software de teléfono de red de la presente invención incluye varias funciones de nivel bajo que están incluidas como parte del nivel de ISR de software. Algunas de estas funciones de nivel bajo son funciones relacionadas con E/S, que se usan con el puerto paralelo E/S de 8 bits del teléfono definido en la figura 24. Las funciones relacionadas con E/S, de nivel bajo incluyen: el monitor de estado "Cuelgue", *Hookst()*; la comprobación y lectura de la disponibilidad de entrada de tecla, *Key()*; aparato de teléfono y control de manos libres, *HandSet()*; reinicio de controlador de Ethernet, *ENET_reset()*; control de volumen, *AmpControl()*; y reinicio de software del sistema.

El chip 136 de interfaz de audio, que adopta preferiblemente la forma de LM4830, puede usarse para controlar la conmutación entre el aparato de teléfono y el modo de manos libres. Por ejemplo, la función *HandSet()* puede escribir un '0' en el puerto E/S cuando se requiere el modo "manos libres" o escribir un '1' en el puerto apropiado cuando se requiere el modo "aparato de teléfono".

Las funciones de nivel bajo de la presente invención también incluyen la ISR de interrupción de controlador de Ethernet, *c_int03()*. La estructura de mensaje global para su uso con *c_int03()* se define para el estado del controlador de Ethernet tal como se muestra en la figura 25. Siempre que se envíe un paquete, o un paquete recibido esté completo, el controlador de Ethernet interrumpirá el DSP 122 para indicar la interrupción. El *DSP 122* leerá los estados de transmisión y recepción desde el registro del controlador de Ethernet y luego almacenará el estado en la estructura de estado anterior. Esta información puede comprobarse mediante otras tareas. Además, estos mensajes se leen tras cada transmisión de paquetes. De otro modo, se bloqueará el controlador de Ethernet.

Tal como se indicó anteriormente, se prefiere que el presente aparato de red de la presente invención use el protocolo RTP para transmitir y recibir paquetes de habla en tiempo real. El paquete de RTP está encapsulado en un paquete de UDP. Los módulos *IP_Send_task* y *RTP_In_task* operan para crear y analizar sintácticamente paquetes de RTP. La figura 26 muestra una estructura de cabecera de RTP para un procesamiento de paquetes de RTP.

Cuando la *IP_Send_task* consigue una petición para enviar un paquete de RTP, en primer lugar genera una cabecera de Ethernet y UDP. A continuación, añade la cabecera de RTP en la memoria intermedia de transmisión de paquetes de Ethernet. Finalmente, los datos de RTP se copian al área de datos de RTP y se envían a través de la red de datos.

La figura 27 muestra una estructura de datos para su uso con una función de generación de tono, *Tone_task()*. Los parámetros descritos en la figura 27 están ilustrados en el diagrama de sincronismo de generación de tono de la figura 28.

Tone_task es una tarea retardada que puede ejecutarse aproximadamente cada 0,1 segundos. Se usa para contar el tono activo y la duración de detención definidos en la estructura *ToneType*. La *Tone_task* establece el *ToneState* a ACTIVO durante una ráfaga y PARADA durante un silencio. Una duración de activo y de parada diferente genera diferentes tonos. Son: tono de marcación, tono continuo (sin parada); tono de ocupado, ráfaga de 0,5 s y silencio de 0,5 s; tono de retorno de llamada, ráfaga de 2 s y silencio de 4 s; señal de timbre, ráfaga de 0,8 s dos veces en dos segundos, luego silencio de 4 s.

Preferiblemente, un módulo *ToneGenerate()* genera un tono de una trama de 400 Hz o una señal de timbre de 2400 Hz definida por un parámetro "modo" cuando el *ToneState* está ACTIVO. Por el contrario, se proporciona una señal de silencio de trama.

El aparato de red de la presente invención usa UDP como su protocolo de transporte para SIP. *SIP_task* es una tarea en bucle que permite la señalización de SIP. Puesto que el presente aparato de red puede usarse o bien como llamante o bien como destinatario de la llamada, *SIP_task* opera como UAC (cliente de agente de usuario) y como UAS (servidor de agente de usuario).

La figura 29 es un código fuente que muestra estructuras de datos usadas para procesar las peticiones o respuestas de SIP según el protocolo de SIP. *Tstate* es la estructura de transición de estado usada en *SIP_In_task* y *SIP_task* para una transición de estado de SIP. Los mensajes de SIP analizados sintácticamente están en el mensaje_t de estructura de datos. La llamada de estructura se define para cada llamada y las entradas de llamada totales se definen por `msg[MaxSipEntry]`.

La figura 30 muestra un diagrama de transición de estado de la *SIP_task* que opera como cliente (por ejemplo, un llamante). Cuando el teléfono de SIP comienza a llamar, funciona como cliente. Se creará una llamada a través de las siguientes etapas: se asigna un `msg[CurrentIndex]` de entrada de llamada cuando se levanta el teléfono y la etiqueta de la llamada se ESTABLECE; *CreateSipCall()* crea un paquete de SIP según las entradas de marcación y de ajuste actuales, donde se usa el paquete de SIP como referencia de la llamada y el `us_state` se establece a UAC; *SIPParse()* genera la estructura de mensaje (`msg[CurrentIndex].m`) para la llamada desde el paquete anterior; la *SIP_task* comprobará si hay alguna llamada activa, si hay una llamada (`msg[i].flag` se ESTABLECE), *SIP_task* creará la correspondiente petición según la especificación de SIP y los estados de SIP se actualizarán en *SIP_task* tal como se muestra en la figura 30.

La figura 30 muestra un diagrama de estado a modo de ejemplo con respecto a las operaciones del cliente (llamante), denominado diagrama de transición de estado de UAC de *SIP_task*. Desde un estado inicial (etapa 3002) se introduce un estado de llamada y una *SIP_task* retransmite una petición de INVITACIÓN DE SIP periódicamente (T1) hasta que se reciba una respuesta (etapa 3004). Nominalmente, T1 es 500 ms inicialmente y se duplica después de cada transmisión de paquete. (Etapa 3006) T2 es nominalmente 32 segundos. Si el cliente no recibe respuesta, la *SIP_task* deja de retransmitir cuando el temporizador T2 expira y el estado de SIP se cambiará a Cancelar (etapa 3008). Si la respuesta es provisional, el cliente continúa retransmitiendo la petición hasta siete veces. Cuando se recibe una respuesta final, el estado cambiará a Completado y se generará un ACK (etapa 3010). Cuando el llamante abandona la llamada, el estado cambia al estado Adiós (etapa 3012). Las peticiones ADIÓS también se retransmiten durante el intervalo de T1 hasta que T2 expira con el fin de una transmisión fiable. La variable, `SIP_Status`, cambiará según la respuesta recibida tal como se muestra en la figura 31. Por ejemplo, si se recibe una respuesta 3xx, *SIP_task* iniciará otra llamada a la dirección redirigida. Otras respuestas finales pueden presentarse visualmente en la LCD.

Cuando el aparato de red recibe una llamada, la *SIP_task* funciona como UAS (servidor) de SIP. Los paquetes entrantes se procesan tal como sigue: *UDP_In_task* acepta el paquete de UDP entrante y envía los paquetes a la *SIP_In_task* junto con su dirección IP de origen y número de puerto. *SIP_In_task* procesa el paquete según la especificación de SIP y actualiza los estados por consiguiente. *SIP_task* monitorizará el estado de receptor, establecerá y disminuirá el temporizador T1 y T2 de cada llamada y actualizará los estados de SIP si es necesario.

La figura 32 ilustra un diagrama de transición de estado a modo de ejemplo de un UAS de SIP. Mientras que la *SIP_task* permanece en un estado inicial (etapa 3205), escucha los paquetes de SIP entrantes. Si se recibe una petición de INVITACIÓN, genera una respuesta de llamada (180) y su estado cambia a Invitación y el módulo *SIP_task* avanzaría a una etapa de En Curso (etapa 3210). Si una parte a la que se llamó levanta el teléfono, el estado cambia a Levanta y el proceso avanza a Éxito (etapa 3215), indicando que se ha establecido una sesión de llamada satisfactoria. Si la parte a la que se llamó no levanta el teléfono, el estado cambia a Fallo y el proceso avanza al estado de Fallo (etapa 3220). Tras el éxito o fallo, el cliente dará acuse de recibo del estado actual y avanzará en el proceso al estado Confirmado (etapa 3225). Cuando la parte que llama termina la sesión, el estado cambia a Colgado y el proceso avanza a Adiós (etapa 3230) indicando que se ha completado la sesión actual.

Tal como se expone en el presente documento, el aparato de red es un dispositivo autónomo que puede iniciar y recibir llamadas de telefonía en una red de datos por paquetes. Mientras que la arquitectura autónoma descrita en el presente documento ofrece muchas ventajas asociadas, tales como su coste relativamente bajo de implementación, también pueden proporcionarse definiciones funcionales y de arquitectura de software similares descritas en conexión con el aparato 100 autónomo en un dispositivo de teléfono basado en PC. En tal caso, un ordenador personal convencional que tiene un micrófono, altavoces y una tarjeta de interfaz de red adecuada, está dotado de software para operar de manera consistente con la manera descrita anteriormente. Naturalmente, se efectúan cambios obvios en esta realización, tales como los componentes y funciones de interfaz de usuario que se realizan por elementos convencionales del PC, por ejemplo, el teclado, monitor, ratón y similares. Se proporciona una interfaz GUI con la funcionalidad de telefonía por el software para permitir las funciones de telefonía deseadas.

El aparato de red de la presente invención, además de realizar funciones de telefonía tradicionales, también puede proporcionar una interfaz económica entre la red y el entorno. Aunque no es viable equipar sensores con interfaces de Ethernet, debido al gran número de puertos requeridos y el coste del hardware mínimo requerido, el aparato de red de la presente invención puede convertirse en el punto de reunión para un número de sensores digitales y

analógicos. Esto se consigue generalmente acoplado el sensor externo al aparato de red a través del conjunto 135 de circuitos E/S convencional que está acoplado al DSP 122. El conjunto de circuitos E/S puede adoptar la forma de memorias intermedias sencillas, convertidores A/D, registros y similares. Esta característica es particularmente útil en entornos que tienen teléfonos por motivos de seguridad, por ejemplo, elevadores, vestíbulos, pisos de tiendas, garajes, etc. Ejemplos incluyen: un sensor digital de infrarrojos pasivo (PIR) para detectar la presencia de personas, esto puede usarse para reenviar automáticamente llamadas si no hay nadie en la oficina o como parte de un sistema de gestión de seguridad o energía; sensor de luz analógica o digital para detectar si la oficina está ocupada; sensor de temperatura analógico; detectores de humo, de monóxido de carbono y de radiación; y cierres de contacto para sistemas de seguridad. Por tanto, el presente aparato de red proporciona un punto de integración de sistema.

Para proporcionar adicionalmente una capacidad E/S mejorada, el conjunto de circuitos E/S puede ser compatible con protocolos de control locales tales como los protocolos X10 y CEBus que son normas reconocidas para controlar dispositivos alimentados por cable tales como iluminación o aparatos. La adición de una interfaz de este tipo al teléfono proporciona un control basado en red de tales dispositivos.

La figura 33 ilustra un sistema que emplea el presente aparato de red para establecer llamadas entre dos o más partes en la red. El sistema incluye generalmente uno o más aparatos 100 de red autónomos, tal como se describió anteriormente. Además, el sistema también puede incluir dispositivos 3320 de telefonía basados en PC, tales como un PC habilitado por red que opera un software de telefonía de red adecuado que es un protocolo conforme con el aparato 100 de red. Cada punto extremo de telefonía puede denominarse nodo y tiene una dirección de SIP específica. Empleando esta dirección específica, cualquier nodo que actúa como parte que llama (cliente) puede iniciar directamente una sesión de llamada con cualquier otro nodo en la red (servidor).

El sistema también incluye preferiblemente un servidor 3325 de redirección al que puede accederse mediante diversos nodos en la red para proporcionar servicios mejorados, tales como un servicio de directorio, reenvío de llamada, derivación de llamada, mensajería de llamada y similares. Por ejemplo, una parte que llama que desea iniciar una llamada a JOHN SMITH puede introducir la dirección de SIP para esa persona si se conoce, tal como sip:john.smith@work.com. Si, por otro lado, la parte que llama no conoce la dirección de SIP de la parte, la parte que llama puede entrar en contacto con el servidor 3325 de redirección con una petición para comenzar una sesión con JOHN SMITH. El servidor de redirección incluye bases de datos con información de registro para diversas partes y puede devolver la dirección de SIP a la parte que llama o reenviar la petición de llamada a la dirección de SIP apropiada. Además, la parte a la que se llamó puede tener múltiples direcciones de SIP tales como john.smith@home, john.smith@office, john.smith@lab y similares. El servidor de redirección puede proporcionar una señal de inicio de sesión a cada una de estas direcciones y establecer una conexión entre la parte que llama y el primer nodo de contacto que responde a la petición de inicio. De manera similar, las partes pueden registrarse periódicamente con el servidor de redirección para indicar la dirección de SIP actual en la que pueden contactarse (característica de reenvío de llamada).

El aparato 3305 de red puede estar configurado para actuar de interfaz con uno o más sensores 3310. Se reciben señales desde los sensores por el aparato 3305 de red y pueden enviarse a lo largo de la red a un nodo de red deseado. Las señales desde los sensores pueden detectarse periódicamente por un temporizador en el aparato de red y enviarse a una dirección de SIP almacenada en la memoria. Alternativamente, las señales de sensor pueden medirse por el aparato 100 de red basándose en una orden recibida desde otro nodo (interrogada por un nodo de red remoto) o puede medirse basándose en una señal de interrupción recibida que indica un cambio de estado del sensor (accionamiento por interrupción). Por ejemplo, el aparato 100 de red puede usarse como dispositivo de comunicación de sistema de seguridad que informa del estado de diversos puntos de sensor de seguridad a una estación de monitorización central. En tal caso, el aparato puede comprobar periódicamente el estado de los sensores conectados, tales como sensores de puertas, sensores de incendio, sensores infrarrojos pasivos y similares, e informar a un nodo de estación central del estado actual. En el caso de un cambio de estado que indicaría una condición de alarma, el aparato 100 podría generar una sesión de llamada con la estación central e informar también de esta condición. Naturalmente, el mismo aparato que actúa como comunicador de alarma también puede proporcionar funciones de telefonía completas. Además, aunque se describió una aplicación de seguridad sencilla, también se apreciará que pueden implementarse diversas otras aplicaciones de recopilación y control de datos generalmente como SCADA (control de sitio y adquisición de datos), usando el presente aparato 100 de red.

Para mantener el servicio vital durante los cortes de energía, el aparato de red de la presente invención puede estar equipado con una batería recargable, integrada posiblemente en un transformador de pared.

Como muchas ubicaciones están equipadas actualmente con sólo una interfaz de Ethernet, el aparato de red de la presente invención debe proporcionar un concentrador de Ethernet de dos puertos, con una interfaz de RJ-45 externa. Esto proporciona una operación simultánea tanto del dispositivo de telefonía como del ordenador habilitado por red.

Además de los datos de audio, el presente aparato de red también puede recibir y transportar datos de vídeo. Por ejemplo, una interfaz de entrada de vídeo, o bien analógica o bien a través de un USB (bus universal en serie)

puede acoplarse de manera operativa al DSP 122 para implementar esta característica.

El presente aparato 100 de red también puede acoplarse a una interfaz de Ethernet inalámbrica adecuada para permitir el equivalente de un teléfono inalámbrico.

Los siguientes protocolos pueden añadirse al presente aparato 100 de red para proporcionar una funcionalidad expandida: DHCP y RARP para una asignación automática de direcciones IP; IGMP para suscribirse a grupos de multidifusión; RTSP para recuperar señales de llamada distintivas y de correo de voz; SAP para escuchar anuncios de eventos de "radio" de multidifusión; y DNS para una resolución de nombre (sometido a un espacio de memoria de programa disponible).

Además de operaciones de telefonía básica, el presente aparato de red también puede proporcionar funciones de telefonía de alto nivel. Por ejemplo puede proporcionarse una característica "No molestar" que reenvíe automáticamente llamadas durante un periodo dado a una ubicación designada tal como se especifica por una dirección de SIP introducida por el usuario. Cada vez que se selecciona la característica, tal como presionando un botón en la interfaz de usuario, el tiempo aumenta por un intervalo predeterminado (por ejemplo, 15 minutos).

También puede proporcionarse un "Registro de llamada" en el que la dirección de SIP e información relacionada con respecto a llamadas entrantes se registran almacenando la información en la memoria, con la capacidad de llamar de vuelta a la parte que llama desplazándose por la lista y seleccionando la dirección de SIP del llamante desde el registro mediante interacción del usuario a través del subsistema 160 de interfaz de usuario.

El aparato de red también puede incluir un "libro de direcciones automático". A través de la entrada de usuario o a través de un servidor conectado en la red, el aparato de red puede adquirir una lista de marcación rápida o una lista de nombres almacenados en su memoria local por la que un usuario puede desplazarse (usando la respuesta "múltiples elecciones" de SIP);

Una característica "interfaz para sistema de correo de voz" puede presentar visualmente todas las llamadas no contestadas que han entrado, incluyendo el tiempo de la llamada, el llamante, el asunto y urgencia de la llamada y si el llamante dejó un correo de voz. Las llamadas pueden ordenarse de manera cronológica o por urgencia. La pantalla de llamada presenta preferiblemente cinco botones programables: para borrar la entrada, avanzar y retroceder a través de la lista, devolver la llamada y recuperar el mensaje.

"Llamada distintiva" es una característica en la que el aparato 100 está programado para anunciar a ciertos llamantes mediante un clip de sonido distinto, tal como un timbre distintivo, melodía o el nombre del llamante. En este caso una base de datos pequeña asocia un llamante, o una clase de llamantes (por ejemplo, amigo, cliente, urgente) con una respuesta de timbre seleccionada particular. El clip de sonido o bien se reproduce desde la memoria o bien se recupera desde un servidor;

"Reenvío de llamada" es una característica adicional que puede implementarse en el aparato 100. Normalmente, las llamadas se reenvían por el servidor de redirección proxy. Sin embargo, el aparato 100 de red también puede realizar un simple reenvío por sí mismo, tal como se describió anteriormente para el botón "no molestar". La redirección puede adoptar la forma de llamar al teléfono desde otro teléfono con una orden de REGISTRO, para implementar llamadas de seguimiento. Además, el reenvío automático de llamadas desde ciertos dominios o durante ciertas horas se implementa inmediatamente sin el uso de un servidor de redirección.

Modo "Intercom" es una característica en la que las llamadas entrantes se "contestan" automáticamente, con el micrófono deshabilitado hasta que se presiona el botón pulsar para hablar o se levanta el receptor. Esto también puede usarse como parte de un sistema de dirección pública de seguridad.

Las características de "monitorización de bebé" permiten que el aparato de red actúe como dispositivo de monitorización de audio remoto. Por ejemplo, con la recepción de una llamada entrante, el aparato 100 de red se activa con el altavoz deshabilitado pero con el micrófono habilitado automáticamente de manera que la parte que llama puede escuchar el entorno en el que está ubicado el aparato al que se llama. Esta característica puede contratarse selectivamente, tal como por un código predeterminado o identidad del llamante;

Una característica "radio por Internet" permite que el aparato 100 de red reproduzca automáticamente estaciones de radio proporcionadas por un servidor de multidifusión de RTP local u otra fuente de medios en flujo continuo cuando no está recibiendo o iniciándose una llamada. El aparato 100 puede escuchar anuncios de SAP y puede reproducir la lista de estación en la pantalla, con botones programables. Cualquier llamada de telefonía entrante interrumpe el programa de radio actual.

El presente aparato de red también puede mantener una "lista de destinatarios de la llamada". Si una llamada previa fue satisfactoria, la dirección del destinatario de la llamada se introduce automáticamente en una parte de la memoria usada como lista de marcación de guía local. Cuando esta parte debe marcarse de nuevo, el destinatario de la llamada puede seleccionarse mediante la tecla hacia arriba o hacia abajo de la lista del destinatario de la

llamada. Esto es generalmente una estructura de memoria de tipo FIFO que elimina automáticamente entradas antiguas y las sustituye por entradas más actuales; y

5 "Marcación repetida", que permite una marcación de tecla sencilla o bien del último número marcado o bien del último destinatario de la llamada.

Además, la "mejora de procesamiento de habla", tal como la supresión de silencio, la generación de ruido agradable y la cancelación de eco también pueden incluirse en el presente aparato de red de una manera que es bastante conocida en la técnica de telefonía.

10 Por tanto, se ha dado a conocer un teléfono basado en red que es un "aparato de Internet" autónomo que permite al usuario realizar llamadas de telefonía dentro de una red de área local (LAN) o a través de Internet. Su núcleo es un procesador de señal digital (DSP) sencillo (un microcontrolador optimizado para procesar datos de audio y vídeo). Proporciona servicios que son un superconjunto de aquéllos de una telefonía regular, pero conecta una red de datos
15 de Ethernet en lugar de a PSTN (red pública de telefonía de conmutación). Puesto que Ethernet que se ejecuta a 10 Mb/s puede usar el mismo cableado de par trenzado usado para teléfonos analógicos y digitales, el teléfono de red de datos por paquetes no requiere cablear de nuevo las instalaciones del cliente. Un sistema mínimo consiste en dos teléfonos de red de datos por paquetes conectados por un cable cruzado de Ethernet. Puede implementarse un PBX básico de múltiples líneas que consiste en cualquier número de teléfono de red de datos por paquetes
20 conectado al concentrador o conmutador de Ethernet. Este "PBX" puede escalarse a cualquier número de teléfonos, simplemente añadiendo capacidad y puertos de Ethernet. El teléfono de red de datos por paquetes comparte la Ethernet con otros servicios de LAN. En casi todos los casos, el tráfico de voz será una fracción pequeña de la capacidad de red. (Una única llamada de voz consume aproximadamente 16 kb/s de la capacidad de 10 Mb/s.) El teléfono de red de datos por paquetes ofrece comunicaciones de voz, que implementan las características
25 habituales de PBX. Sin embargo, el presente aparato de red puede usar un servidor ubicado en la LAN o Internet para proporcionar una funcionalidad adicional, tal como servicios de directorio y de ubicación de usuario, reenvío de llamada, correo de voz, servicios asociados.

30 Un PBX basado en el aparato de red actual puede alcanzar teléfonos tradicionales a través de una pasarela de telefonía por Internet (ITG). Una pasarela de este tipo se conecta a la PSTN usando o bien líneas analógicas, interfaces de velocidad primaria o básica de RDSI o bien líneas principales digitales (tal como T1/E1). Se han introducido recientemente ITG como productos comerciales, con capacidades de desde una a aproximadamente 240 líneas.

35 Aunque la presente invención se ha descrito en conexión con realizaciones particulares de la misma, debe entenderse que pueden realizarse diversas modificaciones, alteraciones y adaptaciones por los expertos en la técnica sin apartarse del alcance de la invención. Se pretende que la invención esté limitada sólo por las reivindicaciones adjuntas.

REIVINDICACIONES

1. Aparato de red autónomo de usuario final para proporcionar datos por paquetes a través de una red de datos por paquetes, que comprende:
- 5 un subsistema de controlador de red acoplado a dicha red de datos por paquetes;
- un subsistema de procesamiento de señal digital acoplado a dicho subsistema de controlador de red, comprendiendo además el subsistema de procesamiento de señal digital un programa informático para detectar llamadas entrantes e iniciar sesiones de llamada;
- 10 un subsistema de conversión de señal acoplado a dicho subsistema de procesamiento de señal digital; y
- un subsistema de interfaz de usuario acoplado tanto al subsistema de conversión de señal como a dicho subsistema de procesamiento de señal digital.
- 15
2. Aparato de red según la reivindicación 1, en el que dicho subsistema de procesamiento de señal digital comprende un procesador de señal digital (DSP) y uno o más dispositivos de memoria acoplados a dicho procesador de señal digital.
- 20
3. Aparato de red según la reivindicación 2, en el que dicho programa informático implementa el protocolo de inicio de sesión para detectar e iniciar sesiones de llamada y realizar un control de sesión de llamada.
4. Aparato de red según la reivindicación 3, en el que una dirección de SIP única está asociada con el aparato, almacenándose dicha dirección en al menos uno de dichos dispositivos de memoria.
- 25
5. Aparato de red según la reivindicación 4, en el que los datos por paquetes incluyen datos de audio y en el que el subsistema de interfaz de usuario comprende:
- 30 un aparato de teléfono que comprende un dispositivo de entrada, un micrófono y un altavoz; y
- un dispositivo de presentación visual.
6. Aparato de red según la reivindicación 5, en el que dicho programa informático implementa una característica de monitor, en el que al detectarse una llamada dirigida al aparato desde un llamante, se inicia automáticamente una sesión de llamada con dicho micrófono habilitado y dicho altavoz deshabilitado durante la sesión de llamada.
- 35
7. Aparato de red según la reivindicación 6, en el que se almacenan criterios de identificación de al menos un llamante aprobado en al menos uno de dichos dispositivos de memoria y en el que dicho procesador de señal digital recibe criterios de identificación desde el llamante y activa la característica de monitor sólo si los criterios de identificación recibidos coinciden con al menos uno de los criterios de identificación almacenados de dicho al menos un llamante aprobado predeterminado.
- 40
8. Aparato de red según la reivindicación 7, en el que dichos criterios de identificación se seleccionan del grupo que incluye nombre, dirección de SIP y contraseña.
- 45
9. Aparato de red según la reivindicación 3, en el que el programa informático implementa una característica de reenvío de llamada, en el que al menos una dirección de SIP de reenvío se almacena en al menos uno de dichos dispositivos de memoria, pudiendo seleccionarse al menos una de dichas direcciones de SIP de reenvío por un usuario a través de dicho subsistema de interfaz de usuario, y en el que al detectarse una llamada dirigida al aparato desde un llamante, dicha llamada se redirige a la dirección de SIP de reenvío seleccionada.
- 50
10. Aparato de red según la reivindicación 9, en el que dicha característica de reenvío de llamada se activa durante un tiempo predeterminado en respuesta a una entrada desde el usuario.
- 55
11. Aparato de red según la reivindicación 9, que comprende además un sensor acoplado a dicho aparato para detectar la ausencia de un ser humano, en el que dicha característica de reenvío de llamada se activa en respuesta a una señal desde dicho sensor.
- 60
12. Aparato de red según la reivindicación 1, en el que el subsistema de interfaz de usuario incluye un dispositivo de salida y en el que el programa informático implementa un modo de medios en flujo continuo en el que se reciben datos en flujo continuo desde la red y se convierten en señales perceptibles proporcionadas a dicho dispositivo de salida.
- 65
13. Aparato de red según la reivindicación 12, en el que el dispositivo de salida incluye un altavoz y en el que

cuando no hay ninguna sesión de llamada en curso se reciben datos en flujo continuo desde la red y se convierten en señales de audio proporcionadas a dicho altavoz.

5 14. Aparato de red según la reivindicación 13, en el que el programa sale del modo multimedia en flujo continuo en caso de que se inicie una sesión de llamada nueva.

10 15. Aparato de red según la reivindicación 12, en el que el dispositivo de salida incluye un altavoz y en el que se reciben selectivamente datos en flujo continuo desde la red y se convierten en señales de audio proporcionadas a dicho altavoz.

16. Aparato de red según la reivindicación 12, en el que los datos en flujo continuo se reciben desde la red y se reenvían selectivamente a otro dispositivo durante una sesión de llamada pudiendo convertirse los datos en señales perceptibles por dicho dispositivo.

15 17. Aparato de red según la reivindicación 12, en el que el dispositivo de salida incluye una presentación visual de vídeo y en el que los datos en flujo continuo incluyen datos de vídeo en flujo continuo que se reciben selectivamente desde la red y se convierten en señales de vídeo proporcionadas a dicha presentación visual.

20 18. Aparato de red según la reivindicación 3, en el que el subsistema de interfaz de usuario incluye un dispositivo de presentación visual y en el que el procesador de señal digital detecta la dirección de SIP de llamantes y almacena una pluralidad de direcciones de SIP de llamante en al menos uno de dichos dispositivos de memoria, pudiendo presentarse visualmente dicha pluralidad de direcciones de SIP de llamante en dicho dispositivo de presentación visual y pudiendo seleccionarse en respuesta a una entrada desde el subsistema de interfaz de usuario.

25 19. Aparato de red según la reivindicación 3, en el que el subsistema de interfaz de usuario incluye un dispositivo de presentación visual y en el que el procesador de señal digital almacena una pluralidad de direcciones de SIP a las que se ha llamado en dicho dispositivo de memoria, correspondiendo dichas direcciones de SIP a las que se ha llamado a la dirección de sesiones de llamada iniciadas satisfactoriamente y pudiendo presentarse visualmente en dicho dispositivo de presentación visual y pudiendo seleccionarse en respuesta a una entrada desde el subsistema de interfaz de usuario.

35 20. Aparato de red según la reivindicación 1, en el que dicho subsistema de controlador de red comprende un controlador de Ethernet y un filtro de servicio.

21. Aparato de red según la reivindicación 2, en el que dicho subsistema de procesamiento de señal digital comprende además:

40 un convertidor analógico a digital (A/D) para convertir datos de audio entrantes en datos de audio entrantes digitales;

un codificador acoplado a dicho convertidor A/D para codificar dichos datos de audio entrantes digitales;

45 un decodificador para decodificar datos de audio salientes digitales proporcionados por dicho subsistema de procesamiento de señal digital;

un convertidor digital a analógico (D/A) acoplado a dicho decodificador para convertir datos de audio salientes digitales en datos de audio salientes; y

50 un amplificador de audio acoplado al aparato de teléfono y al altavoz y al micrófono correspondientes para acondicionar dichos datos de audio entrantes y salientes.

22. Aparato de red según la reivindicación 1, en el que el programa informático comprende además:

55 una capa de protocolo de Ethernet;

una capa de protocolo de Internet (IP) apilada encima de dicha capa de protocolo de Ethernet para actuar de interfaz con dicha capa de protocolo de Ethernet;

60 una capa de protocolo de resolución de direcciones (ARP) apilada encima de dicha capa de protocolo de Ethernet para actuar de interfaz con dicha capa de protocolo de Ethernet y dicha capa de IP, y para traducir las direcciones IP en direcciones de control de acceso al medio (MAC);

una capa de protocolo de datagramas de usuario (UDP) apilada encima de dicho ARP y

65 capas de IP para actuar de interfaz con dichas capas de ARP y de IP y para proporcionar transporte en tiempo real de datos y controles de aplicación dentro de dicho sistema de telecomunicación;

una capa de protocolo de transporte en tiempo real (RTP) apilada encima de dicha capa de UDP para actuar de interfaz con dicha capa de UDP y para proporcionar un transporte de datos de audio en tiempo real dentro de dicho sistema de telecomunicación;

5 una o más capas de protocolo de control apiladas encima de dicha capa de UDP para actuar de interfaz con dicha capa de UDP y para señalar y proporcionar un registro de dichos datos de audio en tiempo real; y uno o más protocolos de aplicación apilados encima de dicha capa de RTP para actuar de interfaz con dicho RTP y para formatear dichos datos de audio en tiempo real.

10 23. Aparato de red según la reivindicación 22, en el que dichos protocolos de aplicación incluyen una capa de protocolo RTSP.

15 24. Aparato de red según la reivindicación 1, que comprende además al menos un circuito de interfaz de sensor, estando dicho circuito de interfaz de sensor acoplado de manera operativa al subsistema de procesamiento de señal digital y teniendo un puerto para acoplar de manera operativa el aparato a un sensor remoto.

20 25. Aparato de red según la reivindicación 24, en el que el subsistema de procesamiento de señal digital adquiere datos desde el circuito de interfaz de sensor a intervalos de tiempo predeterminados, formatea los datos adquiridos como datos por paquetes de red y transmite los datos a un destino predeterminado en la red.

25 26. Aparato de red según la reivindicación 24, en el que el subsistema de procesamiento de señal digital adquiere datos desde dicho conjunto de circuitos de interfaz de sensor de una manera sustancialmente continua y formatea los datos adquiridos como datos por paquetes de red y transmite los datos a la red cuando dichos datos adquiridos cumplen al menos un criterio predeterminado.

30 27. Aparato de red según la reivindicación 24, en el que el conjunto de circuitos de interfaz de sensor incluye una entrada de petición de servicio de interrupción y en el que dicho subsistema de procesamiento de señal digital adquiere datos desde dicho circuito de interfaz de sensor en respuesta a una señal en dicha entrada de petición de servicio de interrupción, formatea los datos adquiridos como datos por paquetes de red y transmite los datos a un destino predeterminado en la red.

35 28. Aparato de red según la reivindicación 24, en el que el programa informático incluye una característica de reenvío de llamada, habilitándose dicha característica selectivamente en respuesta a una señal aplicada a dicho circuito de interfaz de sensor.

29. Aparato de red según la reivindicación 28, que comprende además un sensor para detectar la presencia de un ser humano acoplado a dicho circuito de interfaz de sensor y para proporcionar la señal para habilitar selectivamente la característica de reenvío de llamada.

40 30. Aparato de red según una cualquiera de las reivindicaciones 1 a 29, en el que una dirección de red está asociada con el aparato y permite comunicaciones de datos de punto a punto.

31. Sistema de red de datos por paquetes que comprende:

45 al menos un aparato de red de datos autónomo de usuario final para recibir y generar datos por paquetes, comprendiendo dicho aparato de red de datos:

un subsistema de controlador de red acoplado a dicha red de datos por paquetes;

50 un subsistema de procesamiento de señal digital acoplado a dicho subsistema de controlador de red, comprendiendo además el subsistema de procesamiento de señal digital un programa informático para detectar llamadas entrantes e iniciar sesiones de llamada;

un subsistema de conversión de señal acoplado a dicho subsistema de procesamiento de señal digital; y

55 un subsistema de interfaz de usuario acoplado tanto al subsistema de conversión de señal como a dicho subsistema de procesamiento de señal digital;

60 una red de área local acoplada a dichos aparatos de red de datos para conectar en red dichos aparatos de red de datos; y

una pasarela acoplada a dicha red de área local para recibir datos de voz desde una red de telefonía convencional y para proporcionar y recibir datos de voz por paquetes desde y a dichos aparatos de red de datos.

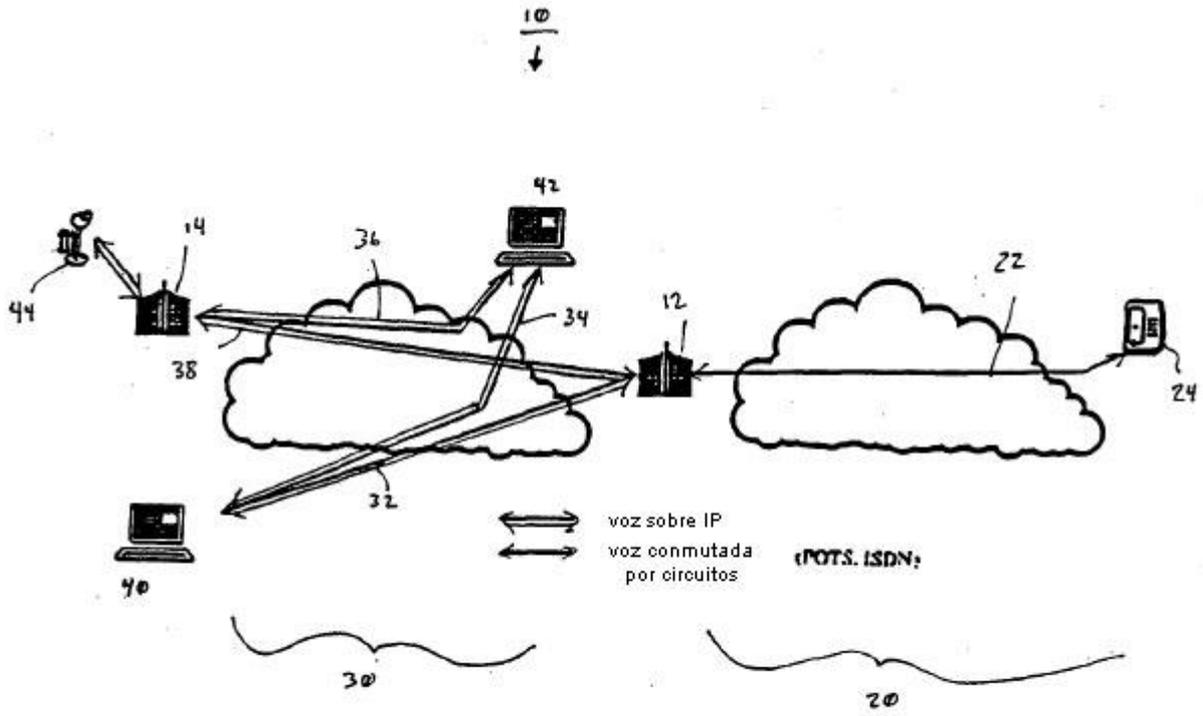


FIG. 1

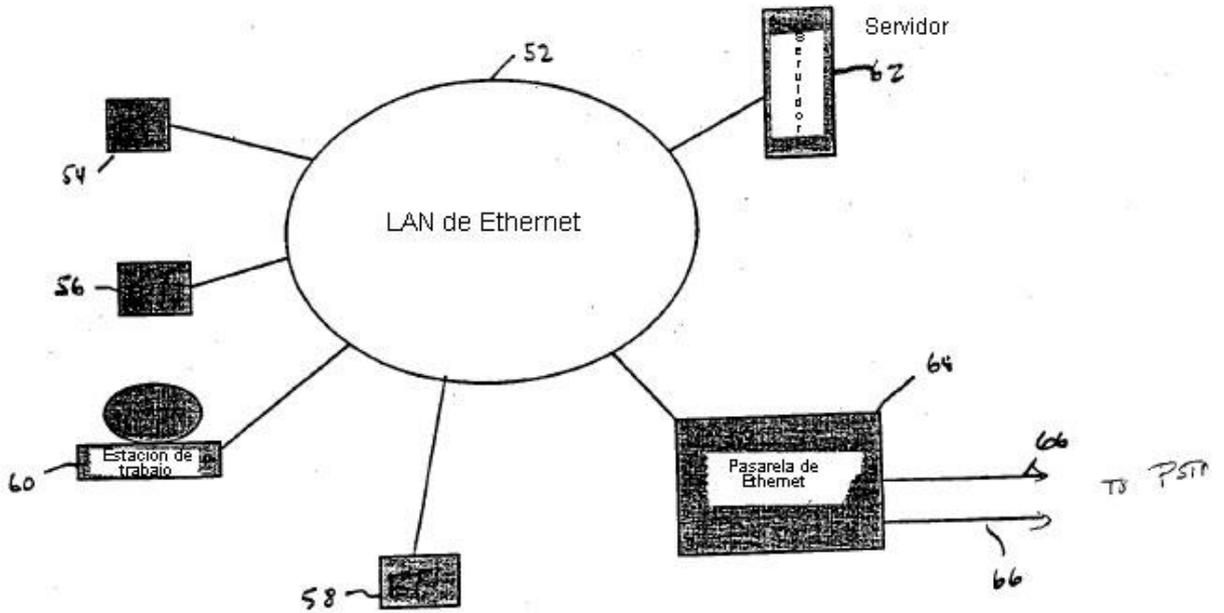


FIG. 2

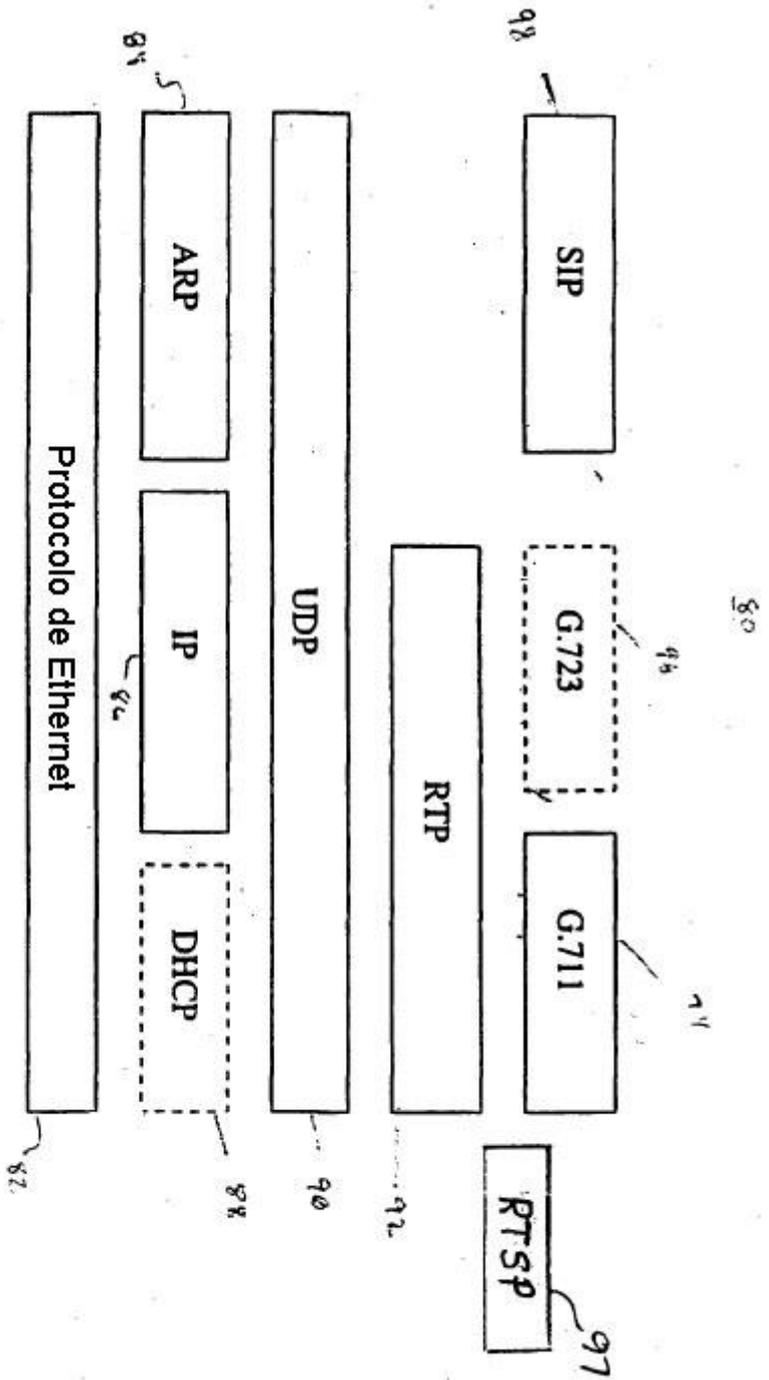


FIG. 3

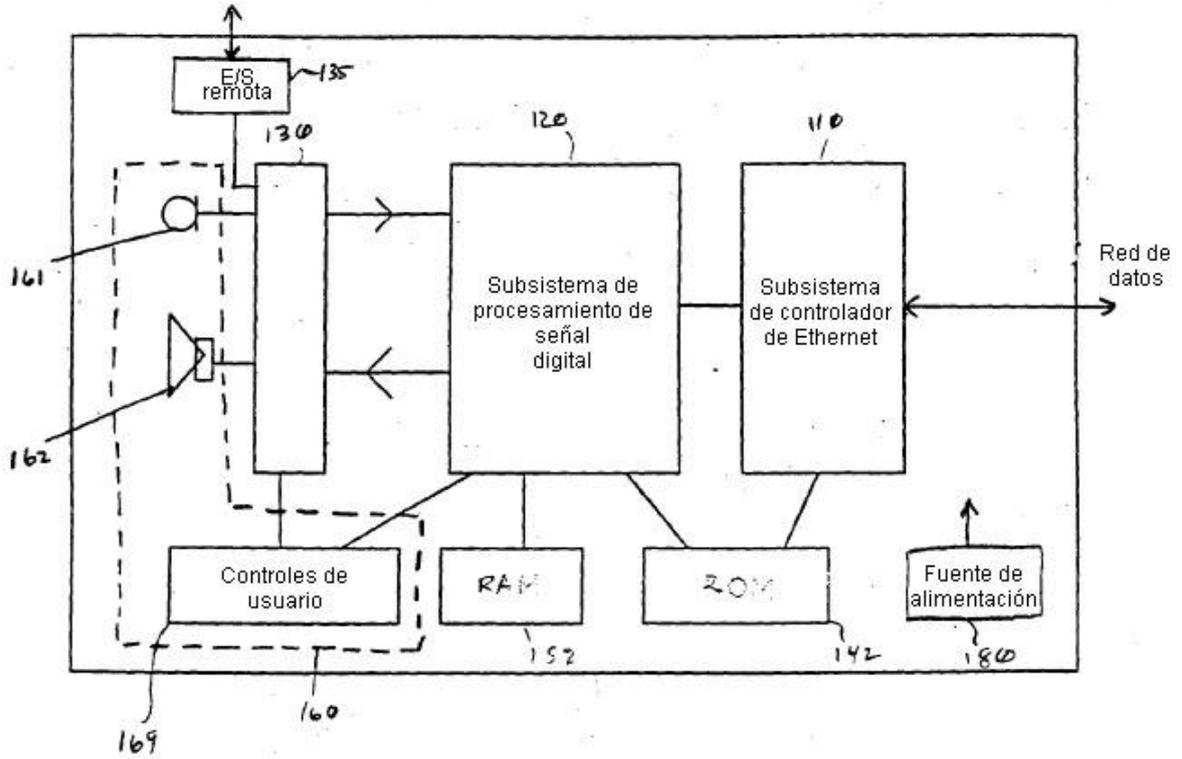


FIG. 4

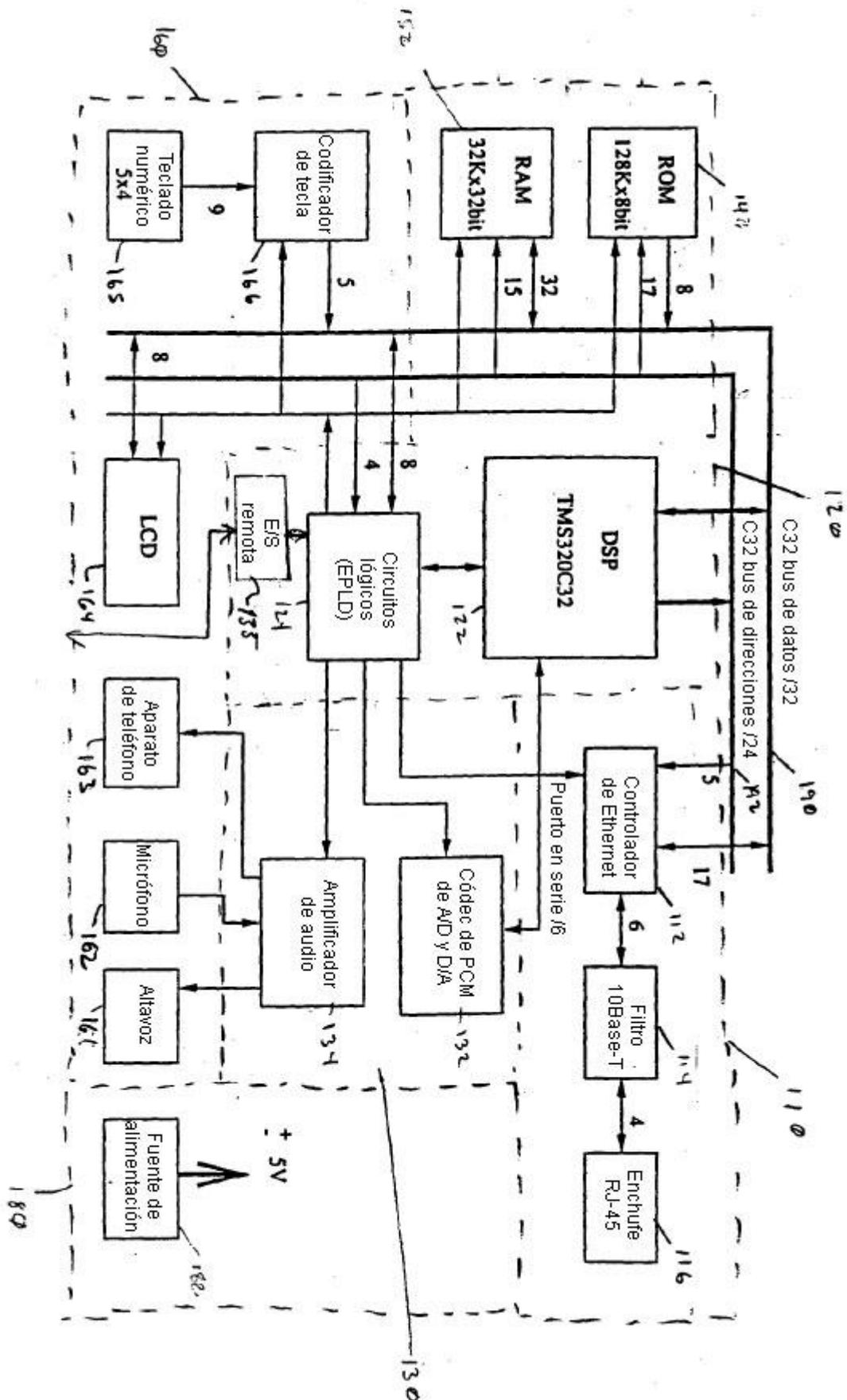


FIG. 5

Dirección de DSP	Longitud (hex)	(dec.) Uso
0x1000	0xB000	128K ROM
0x20000	0x10	16 LCD
0x810000	0x20	32 Controlador de Ethernet
0x820000	0xf	16 lectura de teclado numérico, amplificador de audio, control, control de cuelgue, reinicio de software de sistema
0x87fe00	0x200	512 RAM interna
0x900000	0x8000	32K Ram externa

Fig.6 Mapa de memoria de DSP

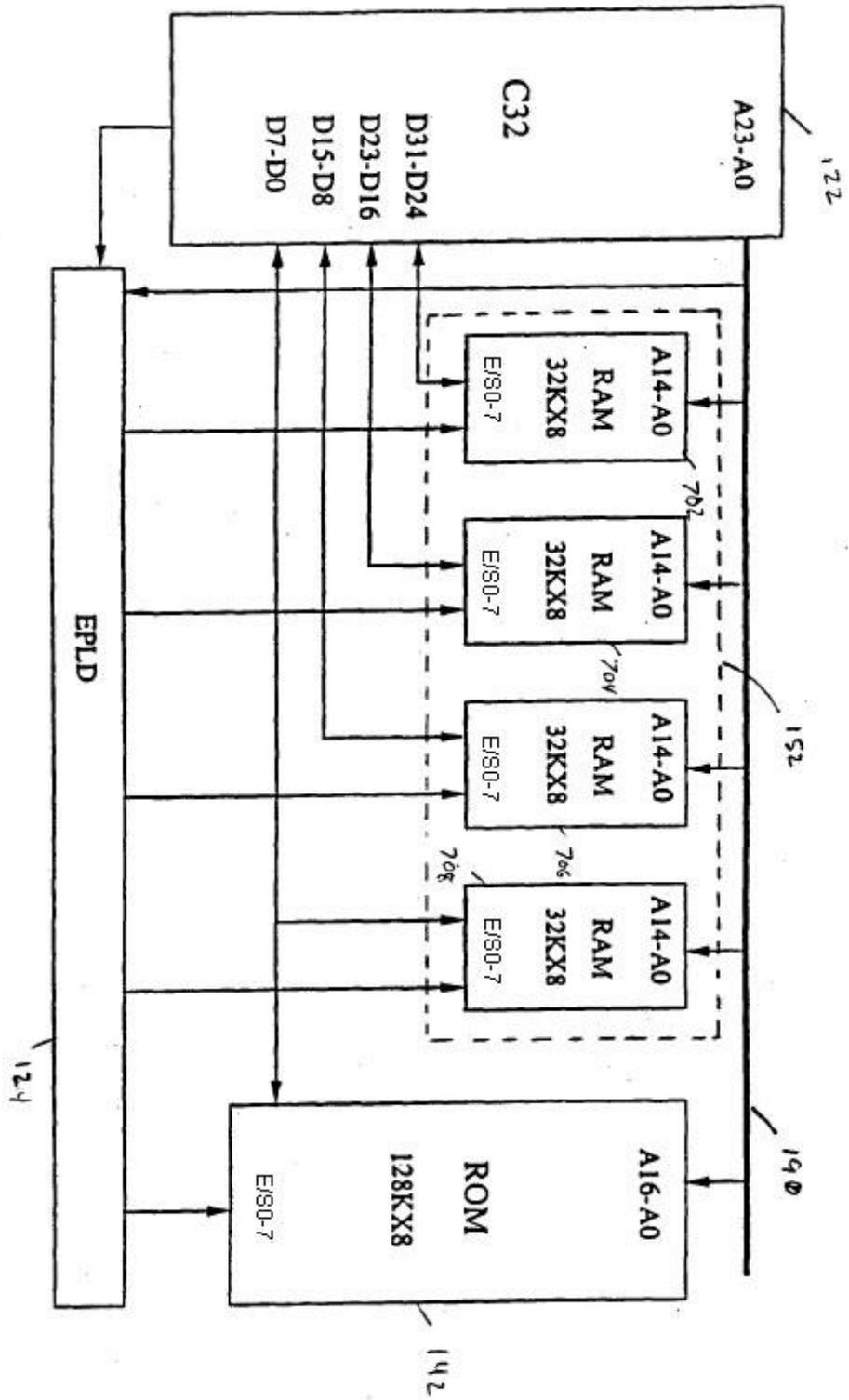


FIG. 7

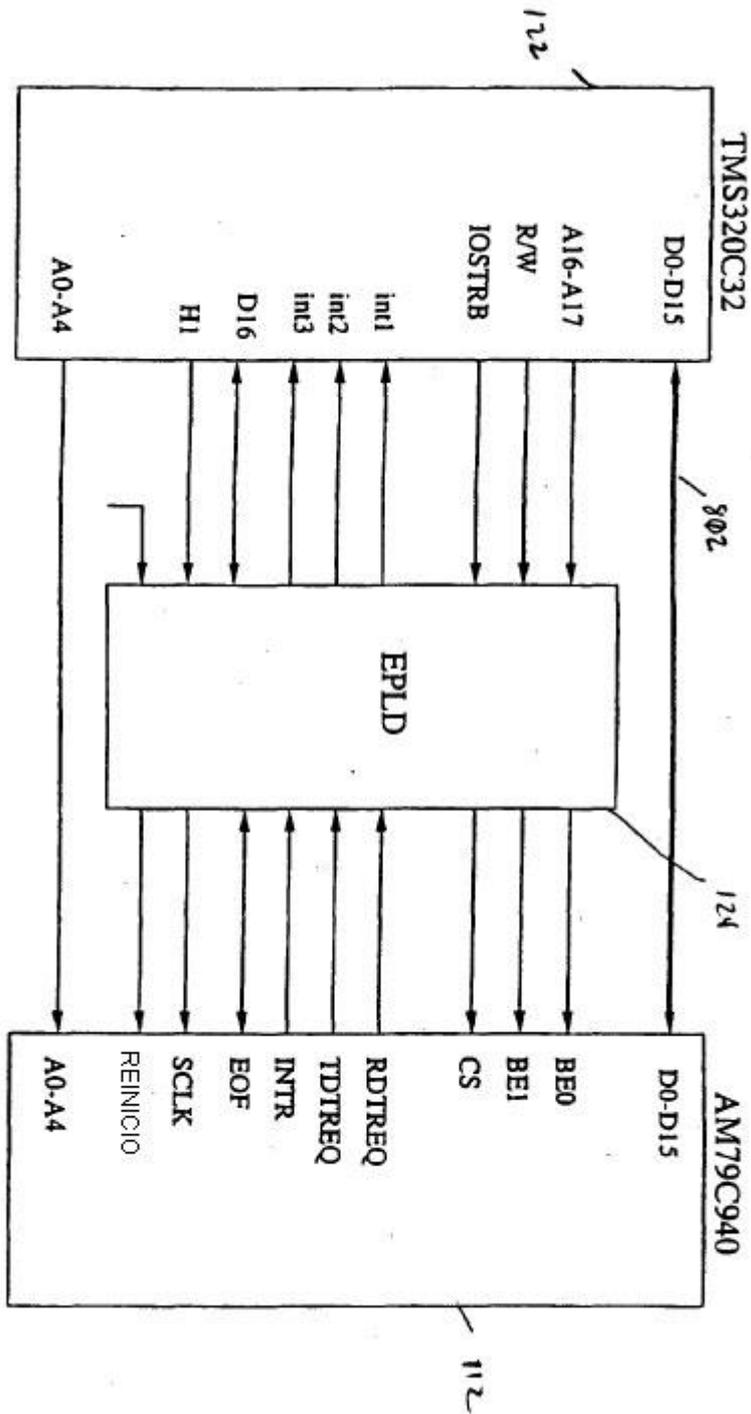


FIG. 8

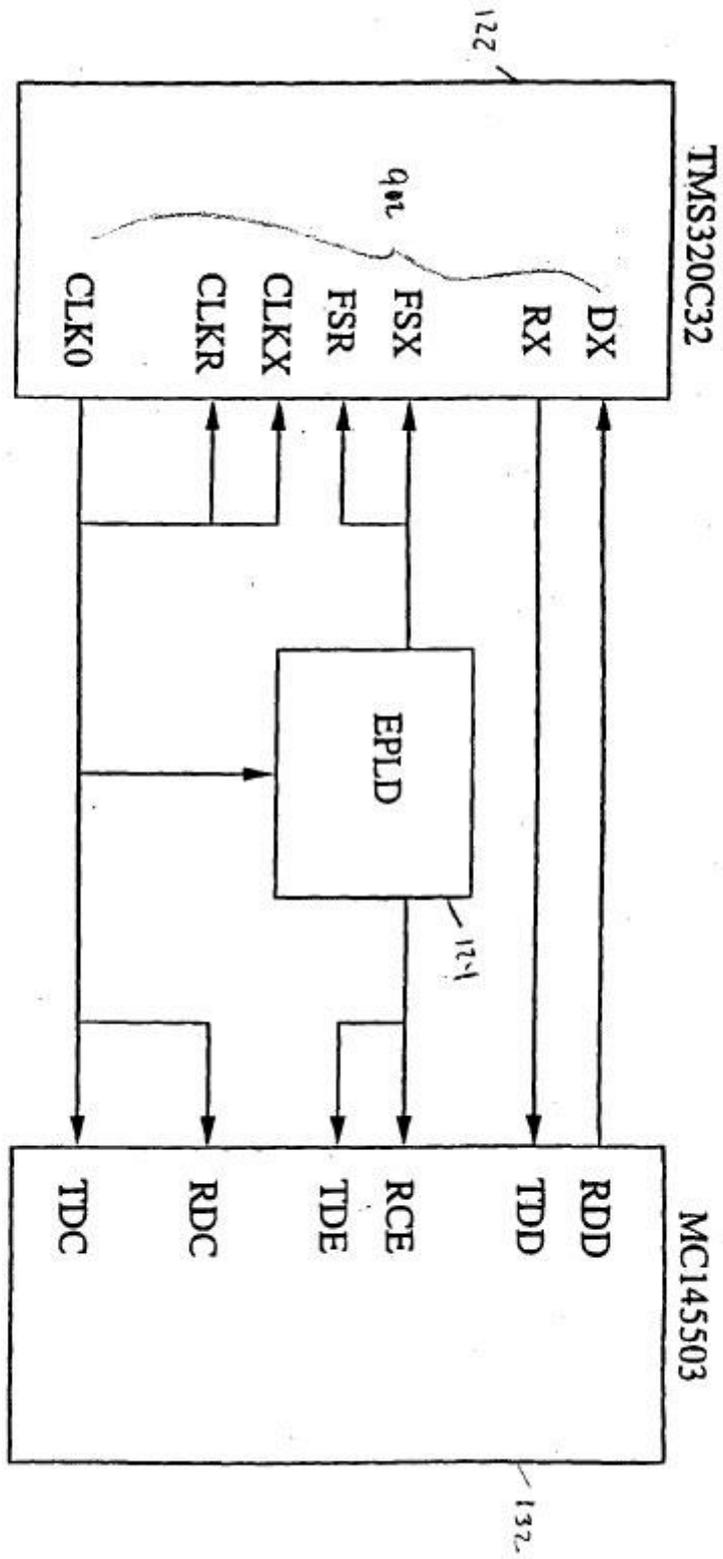


FIG. 9

Dirección de DSP	Uso
0x2000	Puerto de órdenes para mitad izquierda de LCD
0x2001	Puerto de datos para mitad izquierda de LCD
0x2002	Puerto de órdenes para mitad derecha de LCD
0x2003	Puerto de datos para mitad derecha de LCD

FIG. 10

1100

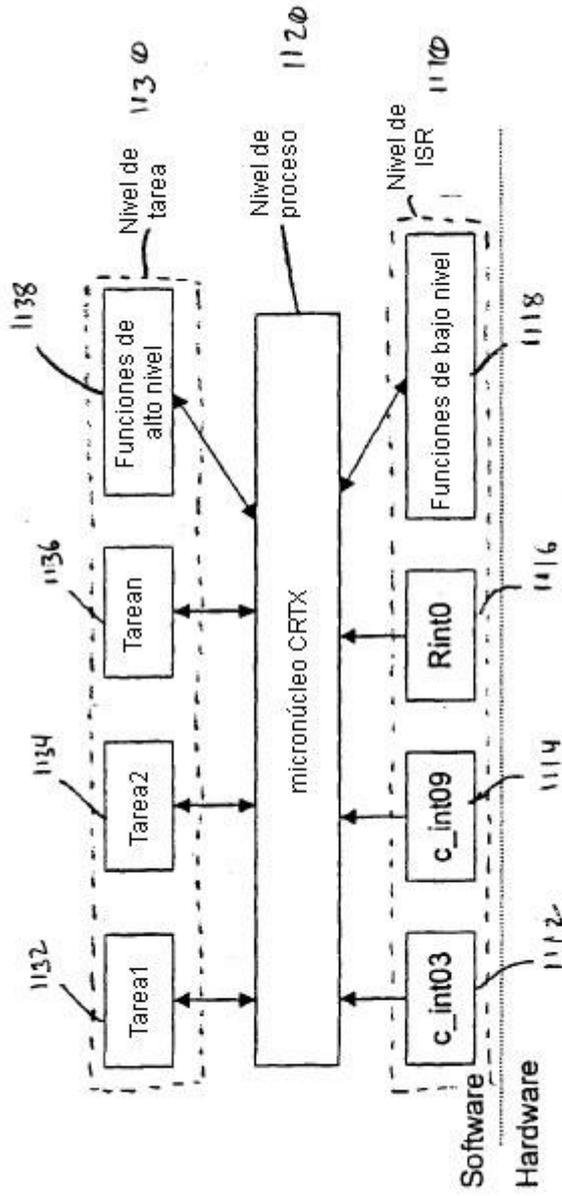


FIG. 11

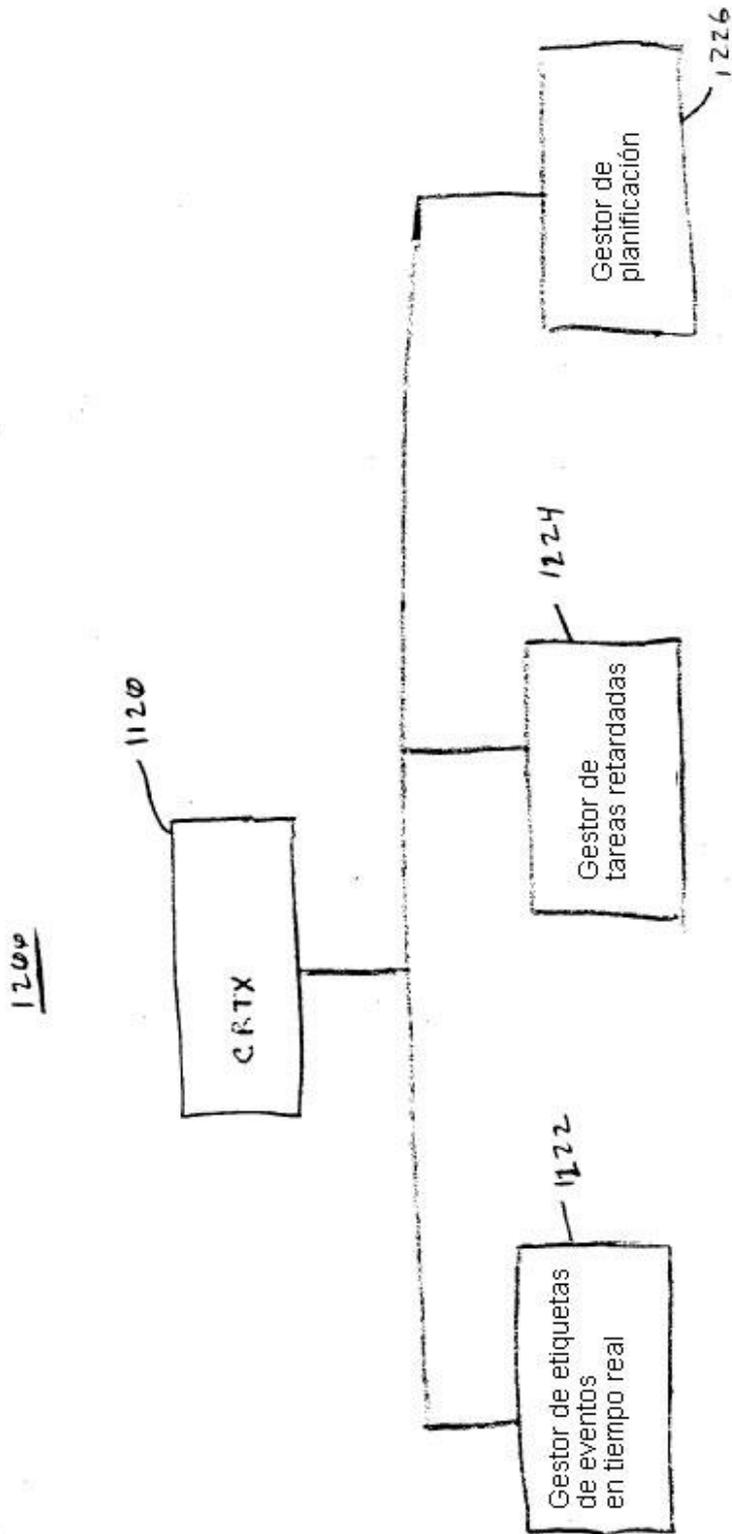


FIG. 12

FIG. 13A

Software de nivel de tarea

Nombre de función	Función
<i>ARPinit()</i>	(Función de inicialización) Inicialización de tabla de ARP
<i>c_int00()</i>	(Función de inicialización) Programa principal, inicializar el puntero de pila, interfaz de bus externa y vector de interrupción para TMS320C32
<i>DMA_initialize()</i>	(Función de inicialización) Inicializar los canales DMA0 y DMA1
<i>ENET_initialize()</i>	(Función de inicialización) Inicializar el controlador de Ethernet
<i>InitHardWare()</i>	(Función de inicialización) Inicializar el Temporizador0, Temporizador1 y puerto serie
<i>NameInit()</i>	(Función de inicialización) Inicializar algunas cabeceras de SIP y cuerpo de SDP
<i>SerialPortInit()</i>	(Función de inicialización) Inicializar el puerto serie
<i>ARP_In_task()</i>	Analizar sintácticamente paquetes de entrada de ARP
<i>ARPTimer_task()</i>	Temporizador de ARP, mantener la tabla de ARP
<i>Call_task()</i>	Procesamiento de llamada
<i>Clock_task()</i>	Un reloj genera la hora, minuto y segundo
<i>Codec_task()</i>	Una tarea para codificación, decodificación de llamada, generación de timbre, generación de tono o bucle de memoria
<i>CreateSipCall()</i>	Crear un paquete de petición de SIP para una llamada
<i>Ercv_task()</i>	Receptor de paquete de Ethernet y demultiplexación de IP

Fig. 13B

Nombre de función	Función
<i>IP_Send_task()</i>	Multiplexación de IP y envío de paquete de Ethernet
<i>Key_task()</i>	Monitor y entrada de teclado numérico
<i>RTP_In_task()</i>	Procesamiento de RTP
<i>Sendto()</i>	Enviar paquetes de UDP a una dirección IP dada
<i>Setting_task()</i>	Establecer los parámetros de teléfono E*
<i>SIP_In_task()</i>	Aceptar paquetes de SIP y actualizar la llamada y estado de SIP
<i>SIP_task()</i>	Tarea de transición de estados de SIP
<i>Tone_task()</i>	Contar la duración activa y de detención para un tono o timbre
<i>UDP_In_task()</i>	Aceptar paquetes de UDP
<i>ARP_Out()</i>	(Función de alto nivel) Programa de petición de ARP
<i>ClearScreen()</i>	(Función de alto nivel) Borrar todas las líneas en la LCD
<i>CodecConfig()</i>	(Función de alto nivel) Planificar una tarea de códec según el parámetro de modo de ejecución
<i>Disp()</i>	(Función de alto nivel) Presentar visualmente una cadena en la pantalla LCD
<i>LCD()</i>	(Función de alto nivel) Presentar visualmente un carácter en la pantalla LCD
<i>LCDClear()</i>	(Función de alto nivel) Borrar una línea en la pantalla LCD
<i>LinearToUlaw()</i>	(Función de alto nivel) Datos lineales para conversión de datos de ley u

Fig. 13C

Nombre de función	Función
<i>Initialization()</i>	(Función de alto nivel) Función de inicialización de llamada y tareas de planificación previa
<i>RTP_para_init()</i>	(Función de alto nivel) Generar el sello de fecha y hora aleatorio y SSRC para una sesión de RTP
<i>ScreenScroll()</i>	(Función de alto nivel) Desplazar la pantalla LCD para una línea hacia arriba o hacia abajo
<i>SDPParse()</i>	(Función de alto nivel) Analizar sintácticamente paquetes de SDP
<i>SIPParse()</i>	(Función de alto nivel) Analizar sintácticamente paquetes de SIP
<i>SIP_Request()</i>	(Función de alto nivel) Crear mensajes de petición de SIP
<i>SIP_Response()</i>	(Función de alto nivel) Crear mensajes de respuesta de SIP
<i>SpeechDecode()</i>	(Función de alto nivel) Decodificación de habla
<i>SpeechEncode()</i>	(Función de alto nivel) Codificación de habla
<i>ToneGenerate()</i>	(Función de alto nivel) Genera un tono de marcación, tono de llamada devuelta, tono de ocupado o tono de alerta

FIG. 13D

Software de nivel de proceso

Función
Prototipos de función
Archivo de cabecera
Supervisor (núcleo)
Archivo de fuente de gestor de captura

FIG. 136

Software de nivel de ISR

Nombre de función	Función
<i>c_int03()</i>	ISR de controlador de Ethernet. Activado en INT3 de TMS320C32 por interrupción externa desde AM79C940
<i>c_int09()</i>	ISR de temporizador de sistema. Activado en TINT1 por temporizador interno 1 de TMS320C32
<i>Rint0()</i>	ISR de A/D y D/A. Activado en RINT0 por interrupción de puerto serie interno de TMS320C32
<i>AmpControl()</i>	(Función de bajo nivel) Control de volumen de altavoz
<i>DMA1()</i>	(Función de bajo nivel) Inicio de canal DMA1
<i>DMA0_Release()</i>	(Función de bajo nivel) Inicio de canal DMA0
<i>DMA_int_set()</i>	(Función de bajo nivel) Habilitar INT1 e INT2 para DMA0 y DMA1
<i>ENET_reset()</i>	(Función de bajo nivel) Reiniciar el controlador de Ethernet
<i>ENET_disable()</i>	(Función de bajo nivel) Deshabilitar el controlador de Ethernet
<i>HandSet()</i>	(Función de bajo nivel) Control de conmutación de aparato de teléfono y manos libres
<i>HookState()</i>	(Función de bajo nivel) Comprobar el estado de cuelgue
<i>Key()</i>	(Función de bajo nivel) Comprobación y lectura de teclado numérico
<i>KeyMap()</i>	(Función de bajo nivel) Mapear la entrada binaria de tecla en formato ASCII
<i>LCDCmd()</i>	(Función de bajo nivel) Orden de control de LCD

FIG 13 F

Nombre de función	Función
<i>LCDWrite()</i>	(Función de bajo nivel) Escribir datos de presentación visual en LCD
<i>RintEnable()</i>	(Función de bajo nivel) Habilitar el RINT0 para ISR de Rint0
<i>RintDisable()</i>	(Función de bajo nivel) Deshabilitar el RINT0
<i>SerialPortRst()</i>	(Función de bajo nivel) Reiniciar el puerto serie
<i>TimerEnable()</i>	(Función de bajo nivel) Habilitar el temporizador de sistema TCLK1
<i>TimerDisable()</i>	(Función de bajo nivel) Deshabilitar el temporizador de sistema TCLK1

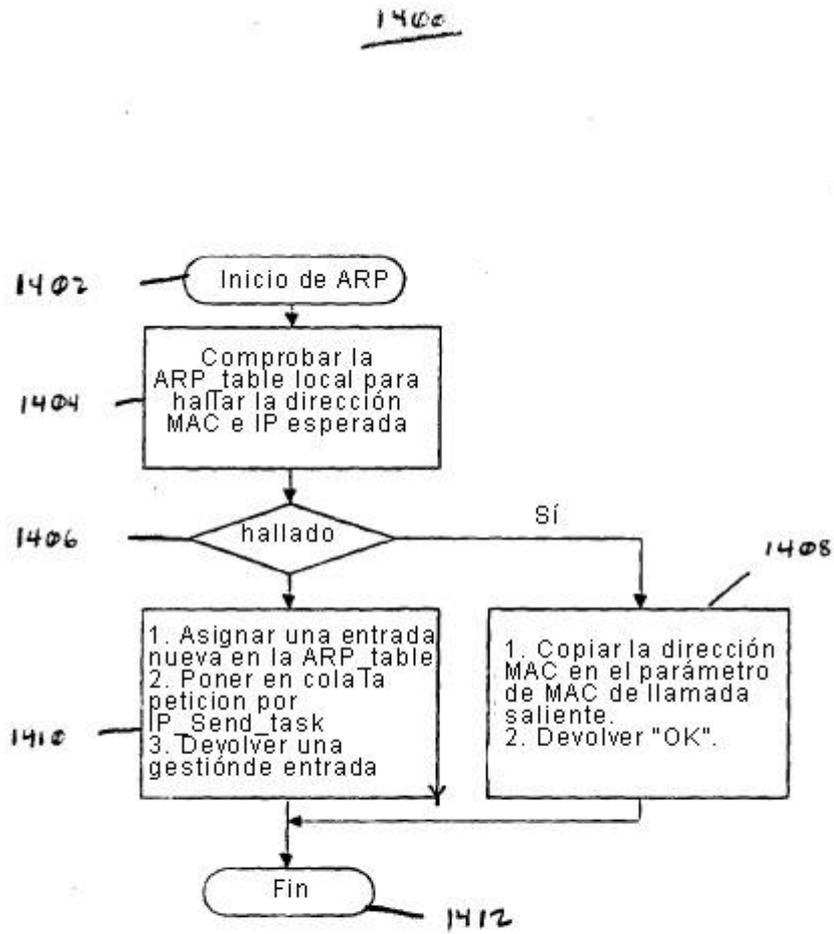


FIG.14

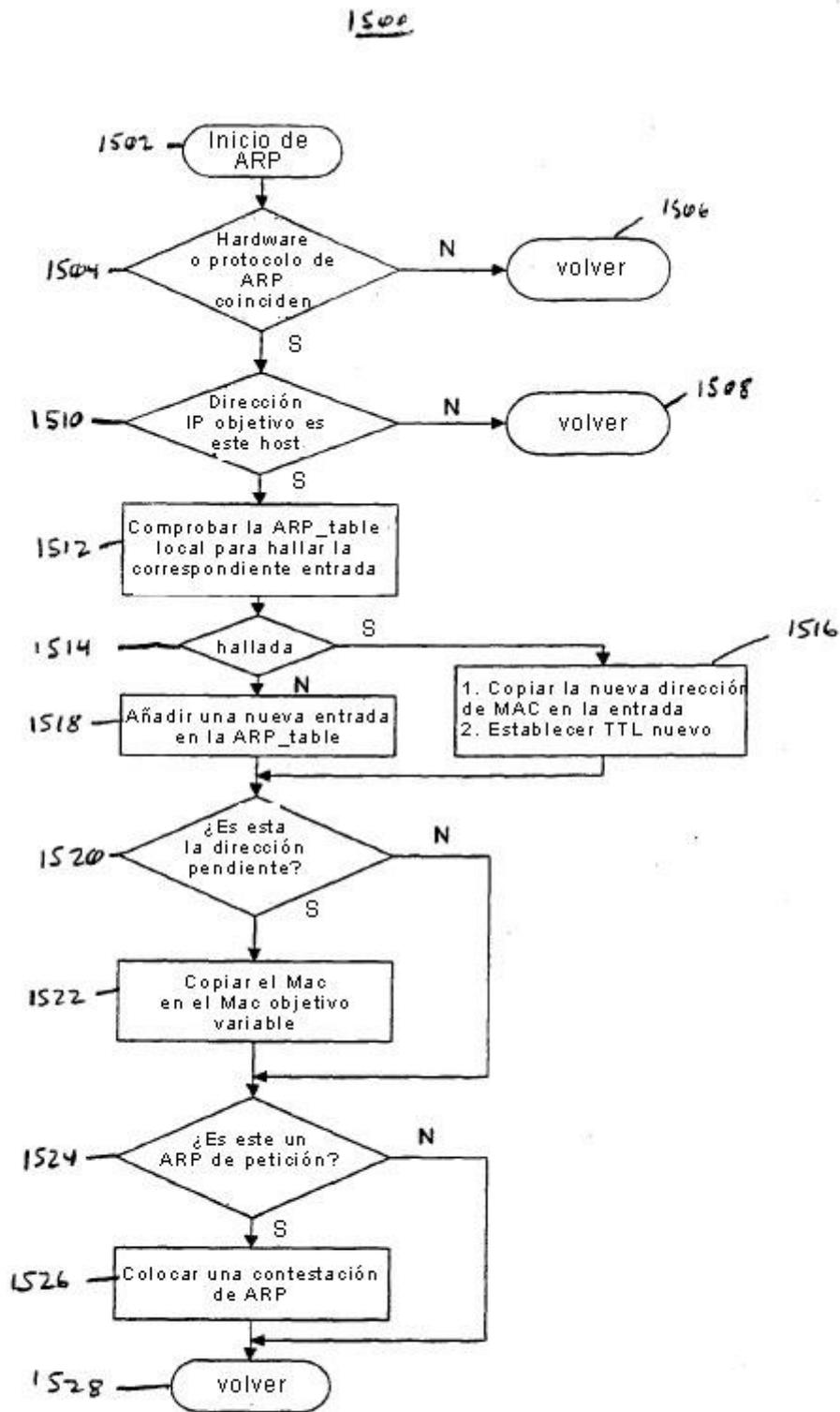


FIG.15

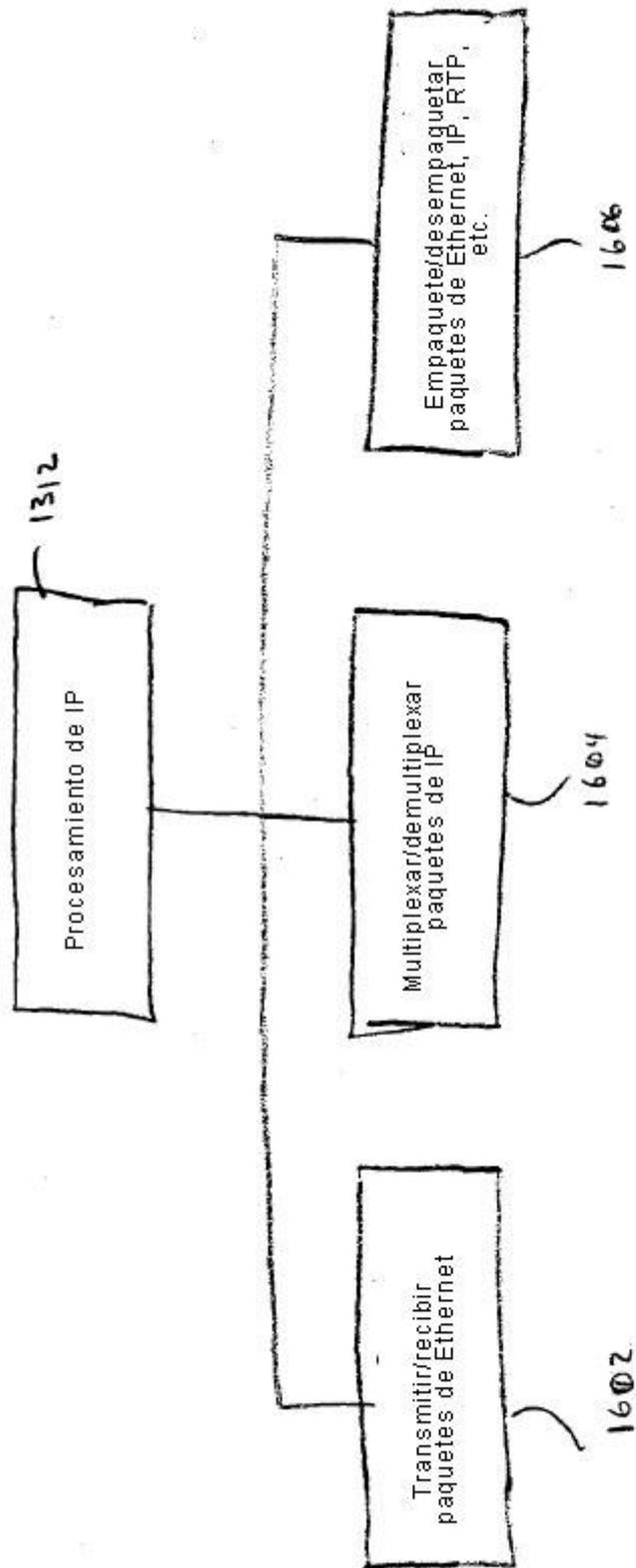


FIG. 16

```

struct ENetHeader {          /* estructura de cabecera de Ethernet */
    ETA Dest;                /* dirección de MAC de destino de Ethernet */
    ETA Source;              /* Dirección de MAC de origen*/
    int Type;                /* tipo de paquete de Ethernet */
};

struct IPHeader {           /* estructura de cabecera de IP */
    int VI_ToS;              /* versión de IP, longitud de cabecera y tipo de servicio */
    int Length;              /* longitud total */
    int Identify;            /* identificador del paquete de IP */
    int FragDff;             /* etiquetas y framento desplazados */
    int TTL_Protocol;        /* periodo de vida y protocolos */
    int ChkSum;               /* suma de comprobación */
    IPA Source;              /* dirección IP de origen*/
    IPA Dest;                /* dirección IP de destino */
};

struct UDPHeader {          /* estructura de cabecera de UDP */
    int SPort;               /* puerto de origen*/
    int DPort;               /* puerto de destino */
    int Length;              /* longitud de mensaje de UDP */
    int ChkSum;              /* suma de comprobación de UDP */
};

struct EPACKET {            /* estructura de paquete de recepción de Ethernet */
    struct ENetHeader Enh;    /* cabecera de Ethernet */
    struct IPHeader Iph;      /* cabecera de IP */
    struct UDPHeader Uh;      /* cabecera de UDP */
    int data[MaxUDPLength];   /* campo de datos */
};

```

FIG. 17

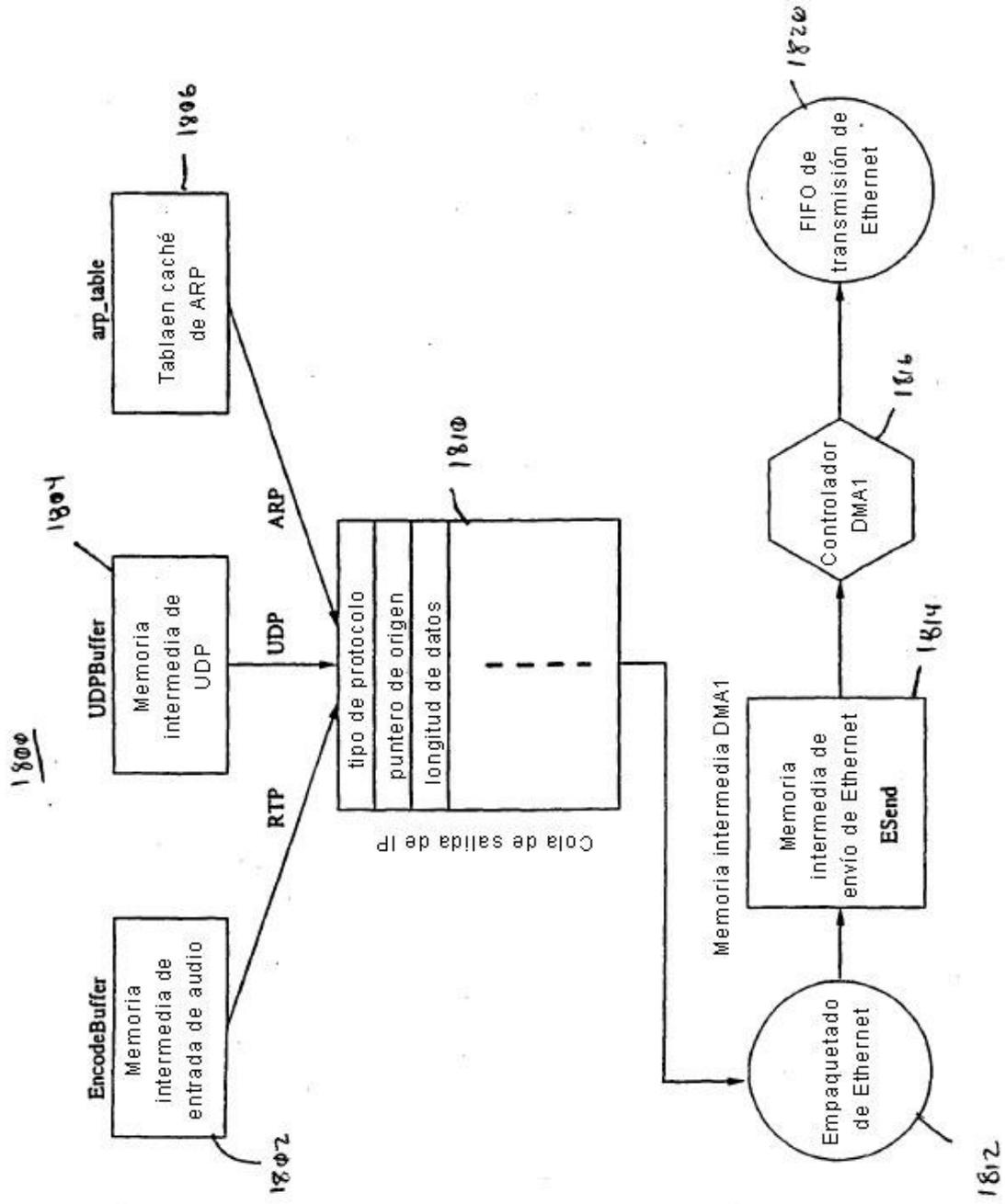


FIG. 18

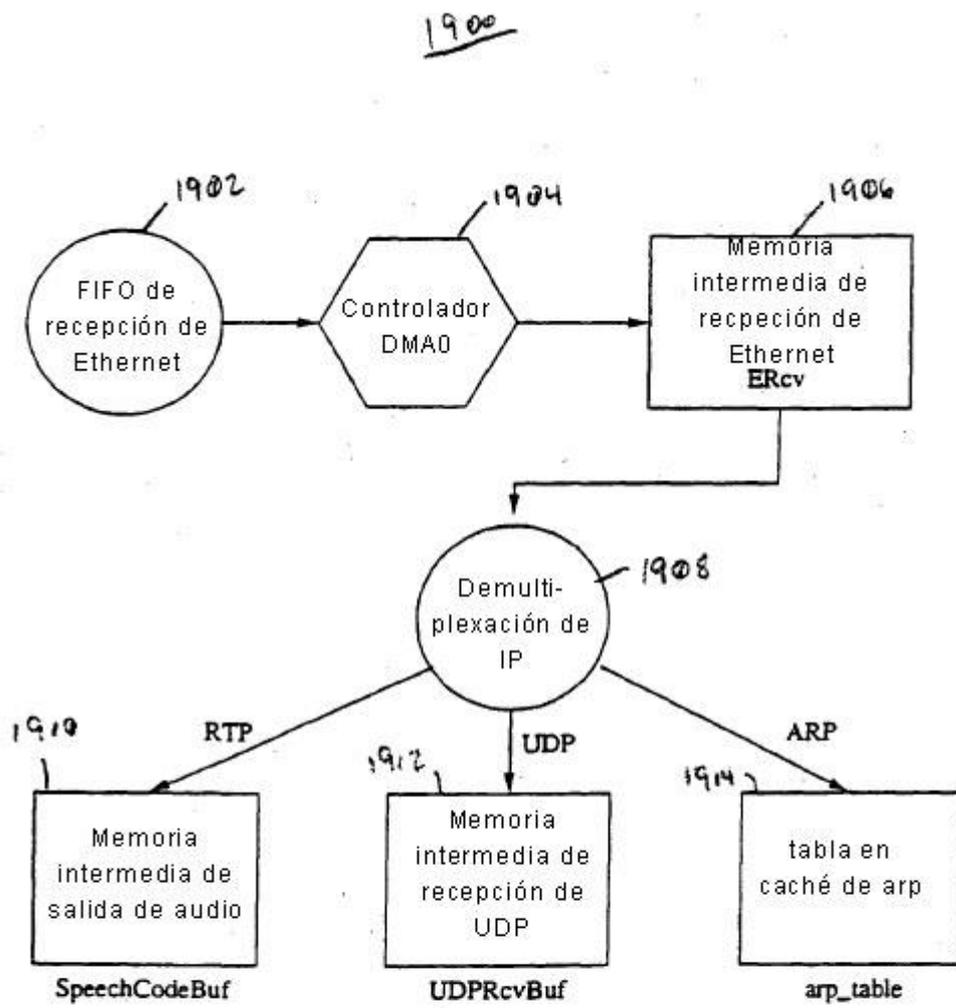
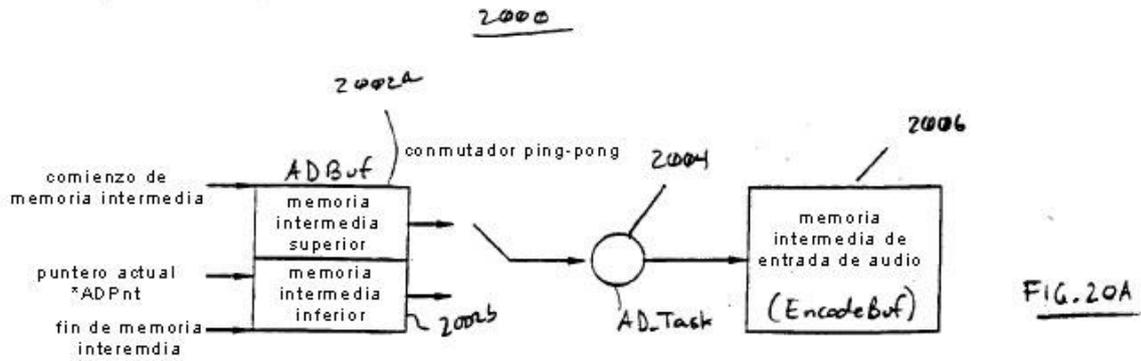
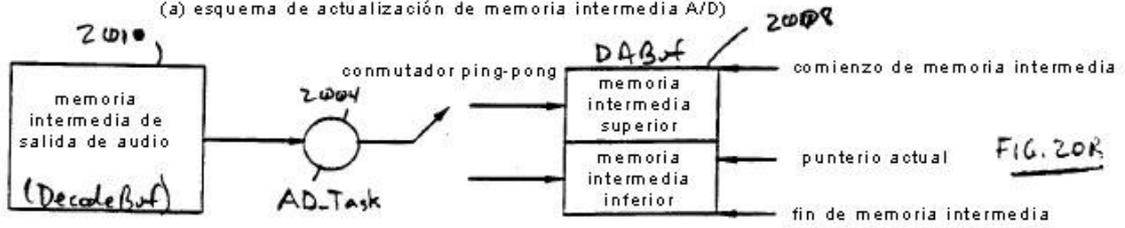


FIG. 19



(a) esquema de actualización de memoria intermedia A/D



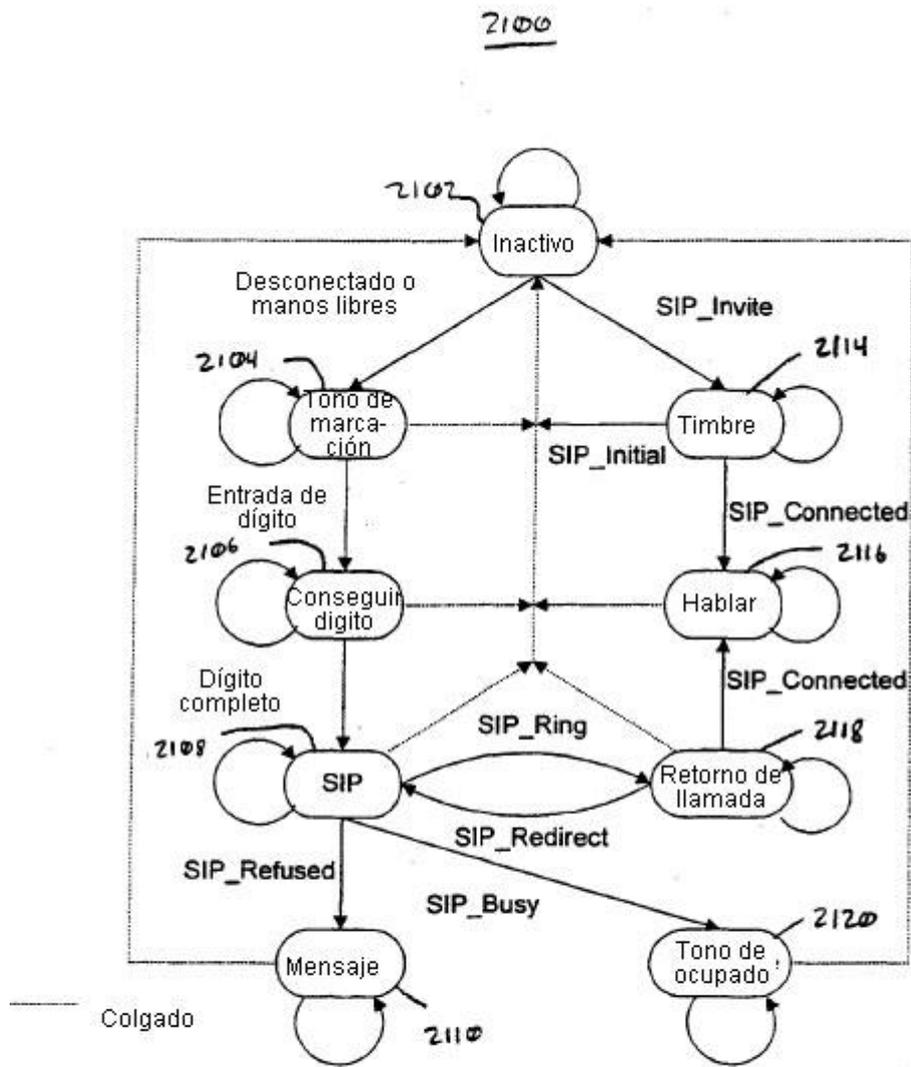


Fig. 22

Teclas	Valor de retorno
Teclas de dígitos	'0' ... '9'
Teclas especiales	'*' y '#'
Tecla introducir	'E'
Manos libres	'H'
Tecla marcar de nuevo	'R'
Hacia arriba	'U'
Hacia abajo	'D'

Fig. 23

```

struct FuncKey {
    PALABRA Introducir; tecla Introducir
    PALABRA Marcar de nuevo; tecla Marcar de nuevo
    PALABRA Arriba; tecla de flecha hacia arriba
    PALABRA Abajo; tecla flecha hacia abajo
    PALABRA Dígito; teclas de dígitos o tecla especial
    PALABRA Llena; tecla memoria intermedia llena
    PALABRA Habilitar; cuando se establece, indica que está habilitada la entrada de tecla
    PALABRA Tocar; se presionó cualquier tecla
    PALABRA Alf; se presionó una tecla alfabética
};

```

FIG. 24

Dirección de puerto	Leer Escribir	Valor de bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x820001	W	0	x ⁵	x	x	x	Reiniciar software	Bloqueo de volumen	Liberar ENET	Manos libres
0x820001	W	1	x	x	x	x	x	Desbloqueo	Reiniciar ENET	Aparato de teléfono
0x820001	R	0	x	x	x	x	x	Sin entrada de tecla	x	Descolgado
0x820001	R	1	x	x	x	x	x	Tecla tocada	x	Colgado

FIG. 25

```

struct Message {
    int ENetXmtST; /* una estrucutra para todos los mensajes en el teléfono de SIP */
    int ENetRcvST0; /* estado de paquete de transmisión de Ethernet */
    int ENetRcvST1; /* estado de paquete de recepción de Ethernet */
    int ENetRcvST2; /* estado de paquete de recepción de Ethernet */
    int ENetRcvST3; /* estado de paquete de recepción de Ethernet */
    int RcvFlag; /* Los datos de habla de recepción están disponibles cuando se ESTABLECEN */
    int ARPST; /* reservado */
};
    
```

FIG. 26

```

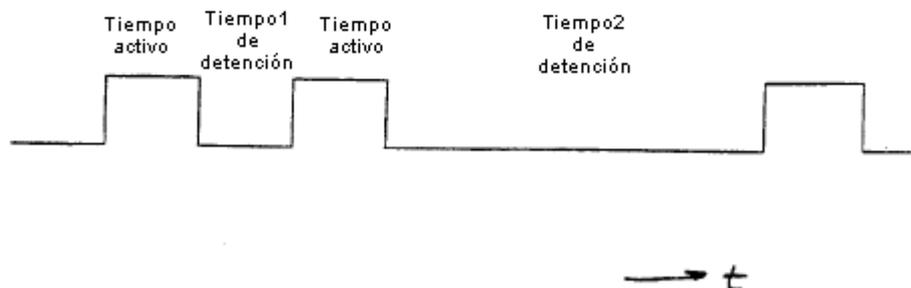
typedef struct {
  int pt:7          /* tipo de carga útil */
  int m:1          /* bit de marcador */
  int cc:4          /* contador de CSRC */
  int x:1          /* etiqueta de extensión de cabecera */
  int p:1          /* etiqueta de relleno */
  int version:2    /* version de protocolo */
  int seq          /* número de secuencia */
  int ts1          /* sello de fecha y hora menos significativo de 16 bits */
  int ts2          /* sello de fecha y hora más significativo de 16 bits */
  int ssrc1        /* fuente de sincronización menos significativa de 16 bits */
  int ssrc2        /* fuente de sincronización más significativa de 16 bits */
  int csrc[1]      /* dirección de lista de CSRC opcional */
} RTPHeader;
    
```

FIG. 27

```

struct ToneType {
  int ActiveTime;    El periodo para sonido está activo
  int ActiveCnt;     El contador para el sonido durante el tiempo activo
  int StopTime1;     Primer periodo de detención de sonido
  int StopCnt1;      Primer contador de detención de sonido
  int StopTime2;     Segundo periodo de detención de sonido
  int StopCnt2;      Segundo contador de detención de sonido
}
    
```

FIG. 28



```

typedef struct {
    char *s;
    short len;
} string;
/* tipo de cadena usada en estructura de mensaje_t */
/* inicio de cadena */
/* longitud de cadena */

typedef enum {
    Initial,
    Proceeding,
    Failure,
    Success,
    Confirmed,
    Calling,
    CallProc,
    Completed,
    Bye
} Tstate;
/* estructura de transición de estado */
/* estado inicial de SIP, UAC o UAS */
/* procedimiento de la petición, UAS */
/* fallo, UAS */
/* éxito, UAS */
/* confirmado, UAS */
/* llamada, UAC */
/* llamada en curso, UAC */
/* completado, UAC */
/* estado Adiós, UAC o UAS */

typedef struct {
    method_t method;
    short status;
    string url;
    string via;
    string callid;
    string contact;
    string from;
    string from_display;
    string subject;
    string to;
    string to_display;
    string ts;
    string reason;
    content_t contenttype;
    int contentlength;
    unsigned cseq;
    string body;
    sdp_t sdp;
} message_t;
/* estructura de mensaje de SIP
/* petición: método; respuesta: 0 */
/* respuesta: valor de estado; petición: 0 */
/* URL de petición */
/* a través de cabecera */
/* identificador de llamadas */
/* contactar con cabecera */
/* Desde dirección */
/* Desde nombre de presentación visual */
/* Asunto */
/* Dirigir */
/* Presentar visualmente el nombre */
/* sello de fecha y hora */
/* frase de razón de respuesta */
/* cabecera de tipo de contacto */
/* longitud de contacto */
/* número de secuencia */
/* cuerpo de SDP */
/* descripción de sesión */

typedef struct {
    int flag;
    int ua_state;
    int status;
    message_t m;
    char * udp;
    char * local;
    sockaddr peer;
    sdp_t sdp;
    Tstate state;
    int t1;
    int t2;
} call;
/* estructura de llamada */
/* ESTABLECER para eficaz, REINICIAR para borrar */
/* Sin llamada actual: 0; UAC: 1; UAS: 2 */
/* estado de respuesta actual */
/* mensaje de SIP */
/* recibir puntero de paquetes de SIP */
/* puntero de paquetes de petición de UAC */
/* dirección IP de host del mismo nivel */
/* copia de seguridad de sdp */
/* estado de transición de SIP */
/* T1 Temporizador */
/* T2 Temporizador */

```

FIG. 29

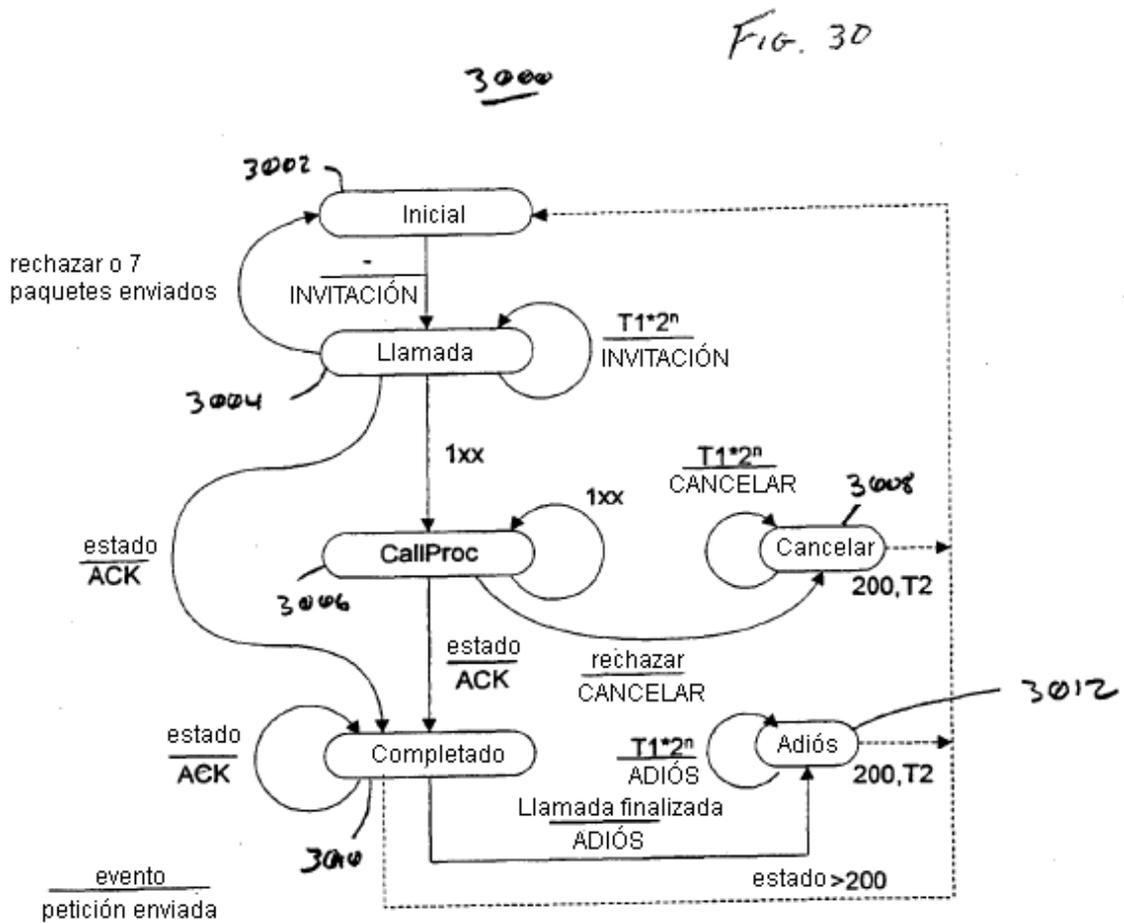


FIG. 31

Mensaje recibido	SIP_Status
100	SIP_Trying
18x	SIP_Ring
200	SIP_Connected
3xx	SIP_Redirect
4xx, 5xx	SIP_Refused
6xx	SIP_Busy

Fig. 32

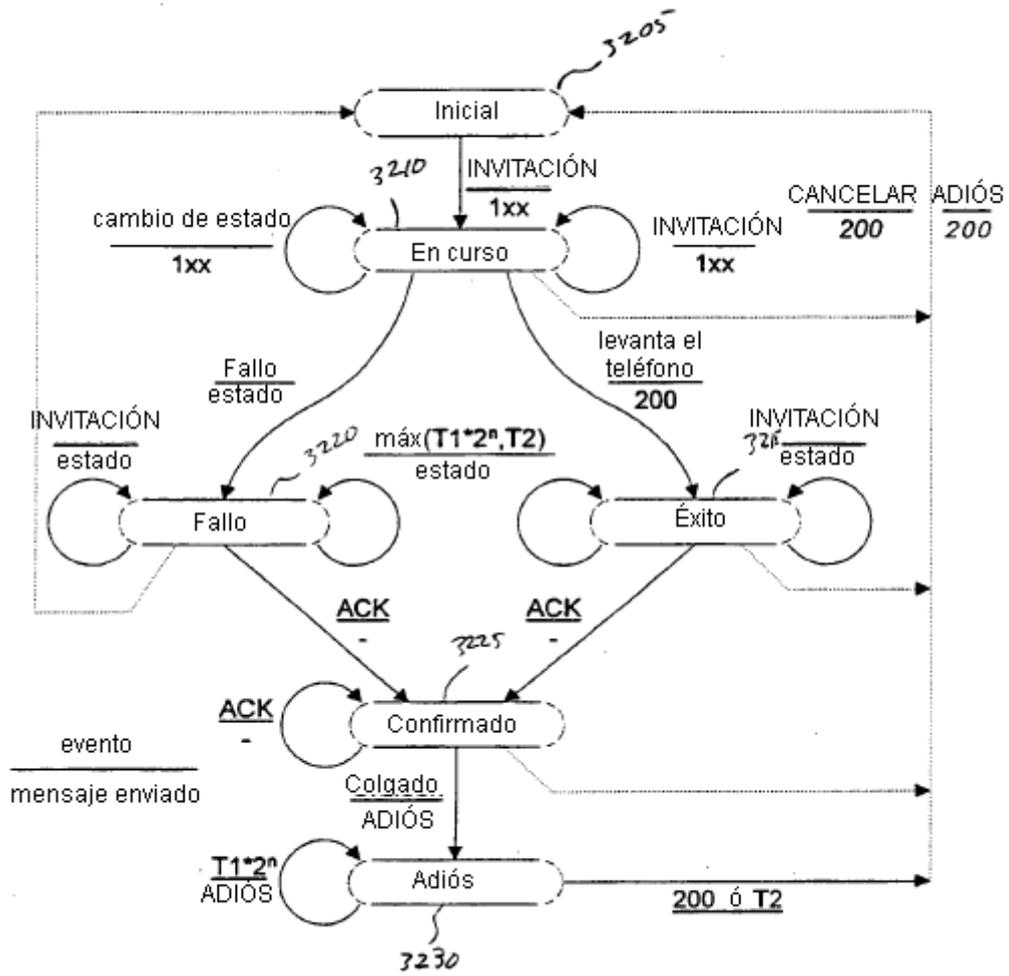


FIG. 33

