(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 266 456**

(51) Int. Cl.:
H04L 29/06 (2006.01)
H04N 7/173 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Número de solicitud europea: **02706093 .8**
(86) Fecha de presentación : **01.02.2002**
(87) Número de publicación de la solicitud: **1364511**
(87) Fecha de publicación de la solicitud: **26.11.2003**

(54) Título: **Protocolo de aplicación de televisión digital para televisión interactiva.**

(30) Prioridad: **02.02.2001 US 265986 P**
02.02.2001 US 266210 P
09.02.2001 US 267876 P
15.02.2001 US 269261 P
28.03.2001 US 279543 P
16.05.2001 US 858379

(45) Fecha de publicación de la mención BOPI:
01.03.2007

(45) Fecha de la publicación del folleto de la patente:
01.03.2007

(73) Titular/es: **Opentv, Inc.**
275 Sacramento Street
San Francisco, California 94111, US

(72) Inventor/es: **Alao, Rachad;**
Delpuch, Alain;
Dureau, Vincent;
Henrard, Jose;
Huntington, Matthew y
Lam, Waiman

(74) Agente: **Ponti Sales, Adelaida**

DESCRIPCIÓN

Protocolo de aplicación de televisión digital para televisión interactiva.

Antecedentes de la invención

Campo de la invención

La presente invención se refiere al campo de las comunicaciones en el medio de la televisión interactiva, y específicamente se refiere a un método y a un equipo que proporcionan un metalenguaje genérico y un protocolo de aplicación de televisión digital para la televisión interactiva.

Resumen del estado de la técnica relacionado

Se pueden utilizar los sistemas de televisión interactiva para suministrar una gran variedad de servicios a los telespectadores. Los sistemas de televisión interactiva son capaces de producir transmisiones de programas de vídeo normales, aplicaciones de televisión interactiva, texto e imágenes gráficas, páginas web y otros tipos de información. Los sistemas de televisión interactiva también son capaces de registrar las acciones o respuestas de los telespectadores y pueden utilizarse en campos tales como la mercadotecnia, el entretenimiento y la educación. Los usuarios o telespectadores pueden interactuar con estos sistemas encargando productos o servicios anunciados, tomando parte en concursos, solicitando información especializada sobre programas específicos o navegando a través de páginas informativas.

Normalmente, un proveedor de servicios de difusión u operador de red genera una señal de televisión interactiva para su transmisión al televisor del espectador. Las señales de televisión interactiva pueden incluir una parte interactiva que comprende un código de aplicación o información de control, y una parte de audio/vídeo que comprende un programa de televisión u otro tipo de visualización de información. El proveedor de servicios de difusión combina la parte de audio/vídeo (A/V) y la parte interactiva en una señal única para la transmisión a un receptor conectado al televisor del usuario. La señal normalmente es comprimida, con anterioridad a la transmisión, y se transmite a través de canales de difusión normales, como por ejemplo líneas de televisión por cable (CA TV) o sistemas de transmisión directa por satélite.

Normalmente, se controla la funcionalidad interactiva de la televisión mediante un STB (STB, *set top box*) conectado a la televisión. El STB recibe una señal de difusión transmitida por el proveedor de servicios de difusión, separa la parte interactiva de la señal de la parte de A/V y descomprime las partes respectivas de la señal. El STB se sirve de la información interactiva, por ejemplo, para ejecutar una aplicación, mientras que la información de A/V se transmite al televisor. El STB puede combinar la información de A/V con gráficos o audio interactivos generados por la aplicación interactiva con anterioridad a la transmisión de información al televisor. Los gráficos y audio interactivos pueden presentar información adicional al telespectador o pueden solicitarle que introduzca datos. El STB puede proporcionar datos de entrada del telespectador u otra información al proveedor de servicios de difusión mediante una conexión o cable de módem.

De acuerdo con su naturaleza de conjunto, los sistemas de televisión interactiva proporcionan contenido en diferentes formatos y protocolos de comunicación que deben ser comprendidos y visual izados por el STB/cliente que recibe la información del proveedor de servicios/operador de red de difusión. Normalmente, el cliente consiste en un STB dotado de un procesador con un poder de procesamiento limitado. La traducción de los diferentes contenidos y protocolos supera la limitada capacidad de procesamiento disponible en un procesador típico de STB. Por consiguiente, se necesita un protocolo de comunicación sencillo que pueda ser comprendido fácilmente por el procesador del cliente/STB y pueda comunicarse con los diferentes protocolos utilizados por los proveedores de servicios.

Resumen de la invención

De acuerdo con un primer aspecto de la invención, se suministra un método para proporcionar una comunicación en un sistema de televisión interactiva que comprende: la recepción en una plataforma de servicios de un primer mensaje dirigido a un servidor de aplicaciones; este primer mensaje se recibe desde un dispositivo cliente e incluye un identificador de usuario. El método se caracteriza por las siguientes acciones: la asignación del identificador de usuario recibido a un identificador de sesión y la transmisión al servidor de aplicaciones de un segundo mensaje que se corresponde con, el primer mensaje; este segundo mensaje contiene el identificador de sesión y no contiene el identificador de usuario.

De acuerdo con un segundo aspecto de la invención, se proporciona un equipo para facilitar la comunicación en un sistema de televisión interactiva. Dicho equipo comprende: medios configurados para comunicarse con un dispositivo cliente por medio de un primer enlace de comunicación; medios configurados para comunicarse con un servidor de aplicaciones por medio de un segundo enlace de comunicación; y una plataforma de servicios configurada para recibir un primer mensaje, incluido un identificador de usuario del dispositivo cliente. La plataforma de servicios se caracteriza por estar configurada para: asignar el identificador de usuario recibido a un identificador de sesión y transmitir un segundo mensaje al servidor de aplicaciones. Dicho segundo mensaje se corresponde con el primer mensaje, y el segundo mensaje contiene el identificador de sesión y no contiene el identificador de usuario.

De acuerdo con un tercer aspecto de la invención, se proporciona un medio legible por ordenador con instrucciones que al ser ejecutadas por un sistema informático distribuido hacen que dicho sistema informático distribuido: transmita un primer mensaje dirigido a un servidor de aplicaciones desde un dispositivo cliente; dicho primer mensaje incluye un identificador de usuario. El medio se caracteriza porque las instrucciones, cuando se ejecutan, hacen que el sistema informático distribuido: a) reciba el primer mensaje en una plataforma de servicios. Dicha plataforma de servicios asigna el identificador de usuario recibido a un identificador de sesión y envía un segundo mensaje que se corresponde con el primer mensaje; este segundo mensaje contiene el identificador de sesión y no contiene el identificador de usuario; b) reciba el segundo mensaje en un servidor de aplicaciones.

La presente invención aborda las necesidades del medio de televisión interactiva mencionadas anteriormente. La presente invención satisface la necesidad, existente desde hace mucho tiempo, de proporcionar un protocolo de comunicación y contenido sencillo que pueda ser controlado fácilmente por un procesador de STB y permita comunicaciones complejas con la plataforma de servicios y los proveedores de servicios. Aunque a continuación se utilizará el ejemplo de un cliente/STB, la presente invención es aplicable a todos los dispositivos cliente, incluidos las agendas digitales, los teléfonos móviles, los ordenadores personales de bolsillo (Pocket PC) o cualquier otro tipo de dispositivos electrónicos capaces de recibir una señal electrónica. La presente invención reside en una plataforma de servicios (PS) o servidor. La PS permite a un operador de red, que proporciona señales de televisión a sus clientes abonados, crear y proporcionar funciones comerciales, de transporte y comunicación que permitan la comunicación entre proveedores de servicios y el cliente o telespectador del STB.

El medio de televisión interactiva debe abordar y solucionar problemas que son únicos a la televisión interactiva, como por ejemplo una ruta de retorno intermitente desde el cliente a la PS. En otras palabras, el dispositivo cliente no siempre está conectado al enlace de comunicación, como por ejemplo cuando el STB se encuentra apagado. Por consiguiente, no siempre existe una ruta activa de retorno desde el cliente. La presente invención proporciona una función de almacenamiento y reenvío para atenuar este problema de ruta de retorno intermitente.

Las limitaciones de ancho de banda y capacidad de procesamiento, así como las complejidades de comunicación, también resultan problemáticas en el medio de televisión interactiva. Por una parte, el operador de red normalmente proporciona un canal de difusión con una capacidad de transmisión de datos relativamente grande (normalmente un satélite y antena parabólica) para enviar datos y programación al cliente. Por otra parte, la ruta de retorno del cliente posee una capacidad de transmisión de datos relativamente baja (normalmente un satélite y antena parabólica) para enviar datos y programación al cliente [sic]. y por otra parte, la ruta de retorno del cliente posee una capacidad de transmisión de datos relativamente baja, normalmente en el caso del STB; una línea de teléfono es la ruta de retorno. Aun cuando la ruta de retorno posee un ancho de banda más grande, los STB/clientes normalmente poseen un módem de baja velocidad para enviar datos por la ruta de retorno. La presente invención aborda ésta y otras cuestiones.

Breve descripción de los dibujos

Otros objetivos y ventajas de la invención quedarán patentes al leer la siguiente descripción detallada que hace referencia a los dibujos adjuntos, en los que:

la Figura 1 ilustra un diagrama de arquitectura de alto nivel para una modalidad preferida de una plataforma de servicios en la que la presente invención reside;

la Figura 2 ilustra una arquitectura para la plataforma de servicios en la que la presente invención reside;

la Figura 3 ilustra un ejemplo de un marco preferido de servicios de fondo (*backend*) para aplicaciones en la presente invención;

la Figura 4 ilustra un ejemplo de una arquitectura preferida de pila DATP de STB en la presente invención;

la Figura 5 ilustra la Pasarela de Servicios (SGW, Service Gateway) y el Protocolo de Transporte de Aplicaciones de Televisión Digital (DATP, Digital TV Application Transport Protocol) de la presente invención como un subconjunto del Protocolo de Aplicaciones de Televisión Digital (DAP, Digital TV Application Protocol) utilizado para normalizar las comunicaciones de canales de fondo entre los servidores de aplicaciones y la SGW;

la Figura 6 ilustra el DAML y DATP como un subconjunto del DAP;

la Figura 7 ilustra un ejemplo de una arquitectura preferida para la SGW de la presente invención;

la Figura 8 ilustra la ventana deslizante con rechazo de la presente invención;

la Figura 9 ilustra un ejemplo de sesión DATP entre un STB y un servidor de aplicaciones;

las Figuras 10-13 ilustran máquinas de estado para DATP;

la Figura 14 ilustra una arquitectura para la traducción de contenido, H2O; y

las Figuras 15-19 ilustran escenarios de mensajes entre el cliente/STB, la SGW, H2O y los proveedores de servicios de aplicación.

Aunque es posible que la invención adopte formas alternativas o incorporar a la misma diferentes modificaciones, en los dibujos se muestran, a modo de ejemplo, las modalidades específicas de la invención, las cuales se describirán en detalle en este documento. Sin embargo, se sobreentiende que los dibujos y la descripción detallada de los mismos no tienen como objetivo limitar la invención a la forma específica divulgada, sino que, al contrario, la invención cubrirá todas las modificaciones, alternativas y productos equivalentes incluidos dentro del ámbito de la presente invención, tal y como se define en las reivindicaciones incluidas al final de este documento.

Descripción detallada de una modalidad preferida

Introducción general

La presente invención, un protocolo de aplicación de televisión digital DAP/DATP, reside en una plataforma de servicios (PS) e interactúa con el transcodificador de contenido (H2O) y una pasarela de servicios (SGW). En un entorno típico de televisión interactiva existe una multitud de clientes, normalmente STB que deben comunicarse con una multitud de servidores de aplicaciones que proporcionan contenido a través de una multitud de redes que utilizan diferentes protocolos de comunicación. Normalmente el STB posee una capacidad de procesamiento limitada, de forma que no resulta aconsejable colocar una gran cantidad de controladores de protocolos de comunicación en el procesador o en la pila del STB. Por consiguiente, se necesita una interfaz de comunicación común que pueda funcionar con todos los STB y servidores de aplicaciones. El protocolo DATP de la presente invención proporciona una interfaz de programación de aplicaciones (API) de comunicación portable y genérica que no requiere una utilización intensiva de procesador, por lo que resulta idónea para un STB típico que posee una capacidad de procesamiento limitada. El DATP requiere un número relativamente pequeño de ciclos de procesamiento, comparado con los protocolos de comunicación de Internet habituales. El DATP reduce la sobrecarga (*overhead*) del controlador de protocolos de comunicación en el STB y permite que el controlador de protocolos de comunicación sea común para todos los STB. El protocolo preferido de DATP es portable para todos los STB, ya que está escrito en código objeto, un código de byte independiente que interactúa con el sistema operativo del STB.

En la presente invención, una pasarela de servicios o SGW actúa como servidor de DATP. La SGW permite a los clientes de la PS en los STB conectarse a los servidores de aplicaciones mediante un protocolo DATP. Se suministra un proxy de HTML a código nativo, H2O, que puede considerarse en este contexto como un servidor de aplicaciones de PS. H2O lleva a cabo una traducción de contenidos específicos, como por ejemplo de HTML a códigos objeto de PS. Los códigos objeto son el código de byte independiente de STB de la máquina virtual que funciona en la PS. En una modalidad preferida, existe en el cliente (normalmente un STB) una implementación en código objeto de la pila de protocolo DATP. El cliente se comunica utilizando un protocolo DATP con un servidor DATP, SGW. El proxy H2O se encuentra al otro lado de la SGW y realiza una traducción de contenidos, por ejemplo de HTML a código objeto. Una implementación en código objeto de una pila DATP en el cliente/STB envía solicitudes de comunicación y se comunica con la SGW utilizando un protocolo DATP. Los contenidos traducidos por H2O pasan a través de la SGW al cliente, en donde se procede a mostrados.

La SGW es un servidor DATP que crea subprocesos de ejecución para controlar todos los STB individuales y procesar todos los contenidos relacionados. La pila de servidor SGW se comunica con el cliente/STB utilizando el protocolo DATP. La SGW también aplica el protocolo apropiado necesario para permitir las comunicaciones entre el STB y los diferentes servidores de aplicaciones. Las aplicaciones de televisión interactiva normalmente utilizan protocolos basados en Internet bien conocidos (por ejemplo, HTML) para la comunicación entre el cliente/STB y los servidores de aplicaciones. La presente invención proporciona un protocolo apropiado de comunicación, genérico y asimétrico, entre el cliente/STB y los servidores de aplicaciones a través de la SGW. La presente invención se adapta bien a la capacidad mínima de procesamiento y memoria disponibles en el cliente/STB.

La presente invención proporciona una solución asimétrica a la compresión de datos. El ancho de banda de la ruta bidireccional desde el cliente/STB al operador de red es relativamente reducida, normalmente una línea telefónica típica o un canal de retorno por cable, por lo general conectados a un módem de baja velocidad. Por consiguiente, con el fin de incrementar el ancho de banda disponible en el módem de baja velocidad se comprimen los contenidos descargados del servidor al cliente/STB. No obstante, en el cliente/STB es preferible no llevar a cabo una compresión de datos. Los datos del cliente/STB que se devuelven son de tamaño relativamente pequeño y no requieren una compresión de datos por el procesador del STB, el cual normalmente no posee la capacidad de procesamiento necesaria para realizar una compresión de datos. Sin embargo, en una modalidad alternativa se dan casos en los que es preferible la compresión de los datos procedentes del cliente/STB, y en esa situación la compresión de datos se realiza en

la SGW. La compresión de datos, por lo que respecta al cliente/STB, es asimétrica, ya que los datos se comprimen en sentido descendente o de descarga hacia el cliente/STB, pero no se comprimen en sentido ascendente o de carga desde el STB. Por tanto, la arquitectura de la presente invención es asimétrica, a diferencia de los protocolos típicos basados en Internet, en los que se asume que las dos entidades entre las que se establece una comunicación poseen una alimentación simétrica.

La SGW y el cliente/STB se comunican con los servidores de aplicaciones mediante el uso de identificadores de sesión para clientes, y no identificadores de usuario, con el fin de preservar el anonimato de los usuarios del cliente. La presente invención también proporciona multidifusión a los clientes. Se puede enviar un mensaje de multidifusión a múltiples clientes mediante un enlace de difusión, cuando el ancho de banda de la difusión y un sintonizador se encuentran en el STB y los mensajes de difusión están disponibles y se detectan mediante un filtro específico configurado en el STB. En este caso, el DATP solicita que el STB reciba un mensaje desde una entrada específica en la difusión. Si no se dispone de un sintonizador para recibir la difusión en el STB, los fragmentos de mensaje también se envían por cada enlace individual punto a punto a los STB que carecen de sintonizador. Si los STB se encuentran en una LAN o red de área local, los mensajes se envían a los STB en direcciones bien conocidas de la LAN.

La presente invención también proporciona una estructura y un método novedosos para el tratamiento de las cookies de las aplicaciones de Internet y proporciona un protocolo "ligero" de HTTP, LHTTP, que encapsula las solicitudes HTTP dentro de mensajes DATP. LHTTP es una versión simplificada de http que funciona encima de DATP. El nuevo protocolo LHTTP no implementa ninguna parte de la especificación de TCP/IP.

La SGW establece un enlace o una conexión de socket con un STB. Sin embargo, para implementar el Protocolo de Datagramas de Usuario (UDP, *User Datagram Protocol*), el UDP no se lleva a cabo directamente. Para que un STB pueda generar el UDP, la presente invención encapsula DATP encima del UDP. Se envía el UDP con DATP encapsulado a la SGW. En el caso del UDP, un socket en la SGW y un socket en el STB son ligados entre sí en una conexión simulada encima del UDP. A través de esta conexión simulada se envían los paquetes DATP desde el STB al servidor de la SGW y desde el servidor de la SGW al STB.

Un gran número de módems de STB no proporcionan un servicio de compresión de datos, poseen una capacidad de procesamiento mínima y no pueden permitirse el coste de procesamiento necesario para llevar a cabo una compresión de datos en el STB. Por consiguiente, en una modalidad preferida la compresión de datos asimétrica se realiza en el STB. El STB recibe los datos comprimidos y los descomprime, aunque no lleva a cabo la compresión de datos. La descompresión de datos resulta menos intensiva para los recursos informáticos que la compresión de datos, por lo cual se prefiere que el STB realice la descompresión. El STB no lleva a cabo una compresión de datos. Los datos comprimidos se envían a la pila DATP en el STB, mientras que los datos no comprimidos se envían desde el STB a la SGW. La SGW realiza una compresión de datos en los datos no comprimidos enviados desde el STB y devuelve los datos comprimidos a los servidores de aplicaciones. Por lo tanto, la compresión asimétrica preferida de DATP/SGW incrementa el ancho de banda de la ruta de retorno desde el STB que atraviesa la SGW y llega a los servidores de aplicaciones.

La presente invención proporciona un enrutamiento asimétrico por parte de la SGW. En el enrutamiento asimétrico, una parte del ancho de banda se asigna a la SGW para enviar datos al *stream* de contenidos programados para su difusión. La SGW tiene la capacidad de decidir si envía datos a uno o más STB a través del *stream* de contenidos programados o por una conexión punto a punto (PTP, *point-to-point*) entre la SGW y el o los STB. La SGW enruta los datos en modo difusión o PTP, basándose para ello en la cantidad de los datos, la velocidad del enlace punto a punto al STB único o a varios de ellos y las condiciones actuales de carga de los enlaces de comunicación. De esta forma, la SGW puede decidir no enviar un conjunto de datos a través del enlace punto a punto porque el conjunto es demasiado grande, y en vez de ello enviarlo a través del *stream* de contenidos programados para su difusión. La SGW puede comprimir los datos antes de enviarlos al *stream* de la transmisión o al enlace punto a punto con el fin de incrementar el ancho de banda del enlace entre la SGW y el enlace o *stream* y tener en cuenta así la limitación de memoria en el STB.

El DATP es ligero, desde un punto de vista computacional, ya que está diseñado para que todas las operaciones de la pila del STB requieran una cantidad mínima de capacidad de procesamiento. Por ejemplo, en el esquema de cifrado de DATP, cuando se utiliza el cifrado de clave pública de Rivest, Shamir y Alderman (RSA), se elige la clave que procede del servidor para que su exponente sea pequeño (3 o mayor), de forma que la fase de exponenciación requiera una cantidad mínima de tiempo y capacidad de procesamiento. Por lo tanto, se reservan las actividades de computación intensiva para el servidor de la SGW, y el procesador cliente o del STB sólo requiere una capacidad de procesamiento mínima. Asimismo, la capa de LHTTP encima de DATP en el STB no tiene que realizar ningún análisis intensivo ni otras operaciones intensivas de procesamiento. En su lugar, LHTTP encapsula los datos de HTTP en los mensajes DATP y la SGW controla las funciones intensivas de computación de HTTP, como por ejemplo la conversión a protocolo HTTP.

DATP no sólo lleva a cabo transacciones. DATP es un protocolo basado en mensajes, no simplemente un protocolo orientado hacia transacciones. Por lo tanto, cuando un usuario envía un mensaje desde un STB a un servidor de aplicaciones, el servidor de aplicaciones no tiene que responder. Es decir, no existe una correspondencia de uno a uno entre los mensajes del STB y del proveedor de servicios. Todos los mensajes DATP, excepto la clase de mensajes DATP no fiables, se procesan a través de una capa de fiabilidad de DATP. Todos los mensajes DATP poseen identificadores únicos que pueden ser utilizados como la base de una transacción.

En una transacción que se sirve de DATP, por ejemplo una solicitud HTTP, el STB envía un mensaje DATP a la SGW solicitando una página web. La SGW convierte LHTTP a HTTP y lo envía a Internet a través de H2O. Una vez que la respuesta, que contiene la página web, vuelve de Internet a la SGW a través de H2O, el cual se encarga de convertir el contenido, la SGW envía un mensaje DATP LHTTP al STB en el que se devuelve el contenido de la página web solicitada a dicho STH. Otro ejemplo de una transacción es una solicitud de Fetchmail enviada desde un STB. La solicitud de Fetchmail es encapsulada en un mensaje DATP. Se utiliza DAML encima del mensaje DATP. DAML es una extensión de XML orientada a dominios específicos.

Por consiguiente, el STB envía a Fetchmail un mensaje DATP que contiene una solicitud DAML (XML). Fetchmaillee el mensaje DATP y extrae el contenido del mensaje, pasa el contenido al servidor de aplicaciones, el cual procesa la transacción, y devuelve un mensaje a Fetchmail. A continuación, Fetchmail envía un mensaje DATP con el contenido solicitado al STB.

DATP es flexible, debido a su asignación al modelo de Interconexión de Sistemas Abiertos (OSI, *Open Systems Interconnection*). El modelo OSI comprende una arquitectura jerárquica de siete niveles para la comunicación entre ordenadores. Cada uno de los siete niveles se agrega al nivel inmediatamente por debajo de sí mismo. Los siete niveles del modelo OSI son, de abajo a arriba: nivel físico, nivel de enlace de datos, red, transporte, sesión, presentación y aplicación. DATP es flexible, ya que abarca cuatro de los siete niveles del modelo OSI. DATP abarca los niveles de enlace de datos, red, transporte y sesión del modelo OSI. Este modelo asume una capacidad de procesamiento simétrica en cada servidor y host que se comunican mediante el modelo OSI. Es decir, el modelo OSI proporciona un modelo simétrico de comunicación. Este modelo simétrico no resulta adecuado para la limitada capacidad de procesamiento del STB. DATP proporciona un protocolo de comunicación asimétrico que se basa en el paradigma de servidor "gordo" (ancho de banda y capacidad de procesamiento grandes)/cliente "delgado" (ancho de banda y capacidad de procesamiento reducidos), diseñado para adaptarse específicamente al medio de la televisión interactiva.

DATP reduce a un mínimo la sobrecarga por byte transmitido. El protocolo DATP se implementa en un formato binario y tiene su propio formato de paquete DATP, de forma que la sobrecarga de paquete es aproximadamente veinte bytes, la mitad de la que se requiere en las tramas de formato TCP/IP. DATP proporciona una capa de fiabilidad. DATP también proporciona "paquetes DATP no fiables" para enviar a mensajes dirigidos a los STB, cuya entrega no se reconocerá y que no se harán fiables a través de la capa de fiabilidad. Los paquetes DATP no fiables resultan útiles para la multidifusión.

La SGW también proporciona una función de almacenamiento y reenvío que sirve para gestionar los periodos de actividad máxima en el número de pedidos enviados desde usuarios múltiples, a la vez que permite reaccionar rápidamente a la solicitud de pedidos de usuarios. La SGW envía rápidamente un "reconocimiento de pedido" al usuario como respuesta a su pedido y almacena el pedido para su transmisión posterior al servidor de aplicaciones, el cual procesará la transacción del pedido. Al enviar el pedido más tarde se puede distribuir un gran número de pedidos durante un periodo de tiempo y no enviarlos todos al mismo tiempo al servidor de aplicaciones. De esta forma se utiliza más eficazmente el ancho de banda. DATP también proporciona una ventana deslizante con rechazo basada en números de secuencia &ente a periodos de tiempo. Más adelante se analizarán en detalle DATP y SGW.

La plataforma de servicios

En la Figura 1 se presenta la plataforma de servicios (PS) en la que reside la presente invención. La PS (50) comprende un grupo de aplicaciones que se pueden dividir aproximadamente en tres categorías: Conversión de contenido (204), Control de transacciones/Funciones comerciales (106) y Conversión de transporte (108). La PS permite a los servicios (200) interactuar con un cliente (212). Los servicios (200) se comunican a través de un enlace de comunicación (102) con la PS (50). La PS (50), a su vez, se comunica con un cliente (212). El cliente (212) puede ser un STB, un asistente digital, un teléfono móvil o cualquier otro dispositivo de comunicación capaz de comunicarse con la PS a través del enlace de comunicación (210). Los servicios de conversión de contenido (204) y de conversión de transporte (108) proporcionan la función de transporte y comunicación, y los servicios de función comercial proporcionan las funciones de control comercial.

En la Figura 2 se ilustra un ejemplo de una implementación preferida de la plataforma de servicios (50). Los servicios (200) proporcionan servicios de compras, charla y otros, ya sea a través de Internet o de otra red o canal de comunicación accesible al operador de red. El operador de red accede a estos servicios mediante el uso de la PS. Las funciones comerciales (206), que comprenden un administrador de servicios (238), interactúan con un administrador de carrusel (254) para obtener el contenido de un servicio (200). El carrusel comprende un *stream* que se repite de difusión de datos de audio/vídeo/interactivos a los clientes desde la PS (50). El administrador del carrusel (254), el administrador de transacciones (242) y el administrador de servicios (238) controlan la inserción y borrado de contenido del carrusel de difusión. H2O (248) se encarga de obtener el contenido de servicios y convertirlo a un formato adecuado para la PS. H2O (248) es una implementación posible de una conversión de contenido (204). H2O convierte el contenido HTML en contenido que puede ser leído por la PS/el cliente. El contenido convertido es formateado a un carrusel de datos y multiplexado por Open Streamer (256) para su difusión al cliente (212). El cliente (212) interactúa con los servicios y, si es necesario, se comunica con la PS y los servicios (200). La comunicación de tipo PTP pasa a través de la SGW (246). La SGW (246) lleva a cabo una conversión de transporte para convertir el protocolo DATP de STB a un formato que resulte aceptable y comprensible para los Agentes Comerciales de Plataforma (226) y H2O (248). El equilibrador de carga (236) interactúa con las funciones comerciales (206), el administrador de carrusel (254)

y la SGW (246) para determinar la carga óptima entre el enlace de difusión (241) y el enlace de comunicación PTP (210). Las funciones comerciales (206) interactúan con los agentes comerciales de plataforma (226) para controlar el acceso y el intercambio de información entre los servicios (200) y el cliente (212).

La PS esconde la valiosa base de datos de perfiles de abonados del operador al exigir que el operador de red proporcione exclusivamente la información de telespectadores a un servicio bajo el control del operador de red. A fin de proteger la identidad de los abonados se transmite un identificador de usuario abstracto (es decir, un identificador de sesión) al servicio durante la sesión en la que el servicio transmite los detalles de la transacción a la PS. El identificador de usuario es específico para cada sesión. Puede existir más de un identificador de usuario asociado con un cliente, como por ejemplo cuando diferentes miembros de una familia utilizan un mismo STB. Se puede asignar individualmente a cada miembro de la familia, así como al STB de la casa, un identificador de telespectador y una categoría, vinculados a transacciones relacionadas con compras, solicitudes de películas, hábitos de visionado de programas, etc., y sobre los que el administrador de telespectadores de la PS realiza un perfil. El servicio únicamente tiene conocimiento del cliente o del identificador de STB a través de un identificador de sesión. Sólo el operador de red, mediante la SGW, puede descomponer un identificador de sesión para obtener los detalles del telespectador (nombre, dirección, información de envío, etc.) necesarios para satisfacer un pedido. Se pueden hacer excepciones para números de tarjetas de crédito u otro tipo de información en aquellos casos en los que el operador no desea realizar cobros de tarjetas de crédito u otro tipo de transacciones.

La presente invención permite a los operadores de red controlar el acceso a la base de datos con la información de telespectadores y permite únicamente a aquellos proveedores de servicios que poseen un acuerdo con el operador de red acceder a información privilegiada (por ejemplo, números de tarjetas de crédito, nombres reales de los telespectadores, direcciones, números de teléfono, números de la seguridad social, etc.). El administrador de telespectadores (252) permite el acceso a la información personal y de perfil almacenada en los dispositivos cliente y permite a estos dispositivos o a la PS seleccionar el contenido y los hábitos de compra preferidos por los usuarios, basándose para ello en el historial de programas vistos almacenado en el perfil del telespectador. Los clientes o la PS seleccionan el contenido preferido por los usuarios, basándose para ello en los perfiles de telespectador, mediante filtros comerciales activados en los dispositivos cliente por el cliente, la SGW u otro componente de la PS.

El administrador de telespectadores (252) proporciona identificación y autenticación de la casa/abonado/STB (o de otro dispositivo cliente) en apoyo de las funciones de SGW y de control parental. El administrador de telespectadores (252) admite la identificación y autenticación del registro de múltiples telespectadores en un único STB mediante el uso de sobrenombres y/o números de identificación personales (PIN), además del identificador de telespectador que se deriva del número o números de identificador del dispositivo cliente, el historial de transacciones, los perfiles de telespectadores, los sobrenombres y los números de identificación personal. El administrador de telespectadores (252) recopila perfiles de telespectadores individuales o de la casa a través del registro, la generación y el emparejamiento vinculados a los hábitos de visionado de programas y de compras por televisión observados. El administrador de telespectadores admite la captura y el almacenamiento de datos distribuidos entre la PS y el STB, así como la sincronización bidireccional.

El administrador de telespectadores (252) permite el uso distribuido de perfiles entre todas las aplicaciones de la PS y proporciona una sincronización con SMS/CRM externos. El administrador de telespectadores (252) permite el registro de telespectadores múltiples para un único STB o dispositivo cliente mediante identificadores abstractos de telespectadores que comprenden el almacenamiento de pseudónimos o sobrenombres, nombres completos y números PIN en el STB o en otro dispositivo cliente. Los agentes comerciales (226) hacen respetar las reglas comerciales de transacciones para la interacción entre los proveedores de servicios y los telespectadores. Basándose en las reglas comerciales, definidas por los operadores de red y que se basan en acuerdos con los proveedores de servicios, los agentes comerciales (226) controlan las transacciones y el acceso del proveedor de servicios a la información sobre usuarios. Los agentes comerciales (226) complementan, agregan, reemplazan y eliminan la información de telespectadores durante una transacción, basándose para ello en los acuerdos con los proveedores de servicios y en un identificador de sesión abstracto.

Los agentes comerciales (226) crean sesiones entre los abonados de los clientes y los proveedores de servicios. Los agentes comerciales (226) controlan el acceso a la información de telespectadores y manipulan esta información mediante la inclusión, sustitución y eliminación de la información de telespectadores presentada a los proveedores de servicios. Los agentes comerciales (226) proporcionan valores predeterminados y controlan el acceso a la información de usuario. Los agentes comerciales (226) también llevan a cabo registros de transacciones, registros de mensajes y supervisión de carga/transacción.

El administrador de publicidad (244) proporciona una interfaz con los enlaces de difusión y PTP que permite una interacción de publicidad gratuita entre los dos canales de entrega. Por ejemplo, un anuncio (difundido cuando la emisora lo desea) puede provocar una conexión PTP al servicio de publicidad a través de la PS, de forma que el usuario puede comprar el producto u obtener más información relacionada con dicho producto. También se puede colocar un anuncio en el contenido PTP para informar a un usuario de la disponibilidad de servicios de difusión (por ejemplo, un publirreportaje).

En algunos casos, se emiten diferentes productos o segmentos de publicidad sin que el cliente solicite dicha información. Se utilizan filtros comerciales asociados con el cliente, ubicados preferentemente en un STB, para seleccionar

el mejor anuncio para el telespectador en función de su perfil de usuario. Por ejemplo, durante un programa de cocina la PS puede programar un grupo de anuncios de cocina para su difusión a los telespectadores. Este grupo de anuncios puede contener anuncios sobre cocina italiana, francesa, india y alemana. El filtro comercial asociado con el STB/cliente o ubicado en el mismo seleccionará qué tipo de anuncio de cocina se presentará al cliente. Un telespectador verá un anuncio de cocina francesa, mientras que otro podrá ver el anuncio de cocina india, dependiendo del filtro de STB fijado por el cliente o la PS, basado en las preferencias de usuario y el perfil de cliente.

La PS permite la reutilización de la infraestructura de comercio Web. La PS reemplaza las plantillas de HTML "normales" con un formato compatible con la PS. Los agentes comerciales reciben las solicitudes de pedido del STB o del cliente a través de la SGW. La SGW pone en cola mensajes (para gestionar periodos de máximo tráfico) y algunos pedidos son recibidos por los agentes comerciales con retraso (los pedidos que no requieren ninguna forma de confirmación son los que preferentemente utilizarían este sistema). Los agentes comerciales agregan información de telespectador a los pedidos. La cantidad y el tipo de información de telespectador proporcionada en un pedido/mensaje se rige por las reglas comerciales establecidas en el acuerdo de servicios/venta al por menor.

En su calidad de comunicaciones entre los servicios y los telespectadores/clientes, la información se envía a carruseles independientes con un único carrusel por *stream* de transporte o se fusionan en los carruseles existentes de las aplicaciones. Los pedidos tienen lugar, si así se desea, a través de una función de "autorización de tarjetas de crédito" suministrada por la PS. A medida que los minoristas envían las confirmaciones, los pedidos se devuelven en tiempo real al usuario mediante correo electrónico o se ofrecen previa solicitud mediante alguna forma de aplicación de atención al cliente.

La PS también proporciona una identificación de telespectador fuera de línea (OVI, *offline viewer identification*), la cual permite a un telespectador ser identificado o autenticado sin necesidad de que establezca una conexión en línea. De esta forma se garantiza que el retraso de conexión (por ejemplo, entre 10 y 40 segundos) quede ubicado en el momento más apropiado durante el proceso de compra. Ello también permite la identificación de telespectador junto con la función de almacenamiento y reenvío. La OVI permite comunicaciones y la realización de pedidos/operaciones con un dispositivo cliente que se enciende y apaga intermitentemente.

Se proporciona una función de formulario de pedido fuera de línea que permite a la PS suministrar servicios de tele-comercio para que un telespectador pueda agregar artículos a un formulario de pedido (cesta de la compra) sin necesidad de estar conectado. La función de almacenamiento y reenvío es importante para lograr un mayor grado de escalabilidad. El almacenamiento y el reenvío pueden realizarse fuera de las horas punta o simplemente se distribuye la carga durante un periodo de tiempo preestablecido después de que se ha iniciado una transacción. La solución íntegra de almacenamiento y reenvío está integrada con el [texto original incompleto] de forma que las respuestas se pueden reenviar desde cualquier canal en cualquier momento. El almacenamiento y el reenvío se pueden utilizar para mejorar las transacciones de comercio electrónico y tele-comercio. La autenticación de telespectador fuera de línea posibilita la selección de pago fuera de línea. La PS proporciona la selección de pago fuera de línea para mejorar el proceso de compra y permitir el uso de la función de almacenamiento y reenvío con el comercio electrónico y el tele-comercio.

La PS utiliza transporte Web estándar cuando es aplicable, es decir, utiliza H1TP para solicitudes en tiempo real y SMTP para una comunicación asíncrona cuando sea aplicable (por ejemplo, en informes de compras y almacenamiento y reenvío). Incluso cuando se conecta en línea, la PS proporciona la capacidad de conectarse durante un breve período de tiempo para acceder a los datos (por ejemplo, por correo electrónico) y a continuación utiliza los datos localmente. La PS proporciona identificadores basados en sesión, en lugar de las típicas cookies de Internet, para proteger la base de datos de telespectadores del operador. En vez de las cookies de Internet, la PS proporciona un identificador basado en sesión que el servicio no puede utilizar para identificar al usuario, sólo a la sesión. El servicio debe solicitar la información de telespectador a la SGW (y el operador de red le cobrará por este servicio).

La PS informa opcionalmente al telespectador cuando se produce una conexión, y también puede pedir opcionalmente la aprobación del telespectador para mantener la conexión. Asimismo, la PS muestra un estado de "Conexión activa" (*Connection ON*) en la pantalla. La PS utiliza un ancho de banda de difusión para la comunicación PTP cuando éste resulta más eficaz. Se suministra un equilibrador de carga que determina qué información se transmite mediante difusión y qué información se transmite mediante la conexión PTP. Las decisiones de equilibrio de carga se basan en la urgencia de los datos, la latencia de entrega de la difusión frente a los enlaces de transmisión PTP, la carga comparativa en la difusión y en las rutas PTP y el número de telespectadores que reciben los datos. En general, los datos dirigidos a un gran número de telespectadores se envían por difusión, mientras que las pequeñas cantidades de datos que necesitan enviarse inmediatamente se envían por un enlace PTP. Los STB que carezcan de sintonizador de banda ancha recibirán los mensajes PTP enviados junto al contenido de banda ancha.

La PS proporciona a los STB y/o clientes filtros que reciben en la ruta de difusión, de forma selectiva, información basada en los perfiles de los telespectadores, de manera que únicamente los telespectadores seleccionados con un filtro específico configurado en sus STB capturan el contenido (publicidad, información, programas de A/V, etc.) en el *stream* de difusión transmitido. Estos filtros mejoran los aspectos de entrega adaptable y selectiva de la PS. El administrador de carrusel proporciona un carrusel de datos para Open Streamer. El administrador de carrusel gestiona un carrusel de datos en tiempo real. El administrador de carrusel complementa Open Streamer. El administrador de carrusel proporciona un componente de servidor y una biblioteca OCOD de cliente STB. El servidor de carrusel recibe solicitudes de las aplicaciones para agregar, eliminar o modificar los contenidos de los carruseles. Cuando un

administrador de carrusel recibe una solicitud, la trata como una transacción única y obtiene todos los datos necesarios (normalmente a través de HITP). El administrador de carrusel genera un índice de carrusel o un archivo de directorio de carrusel nuevos, según se requiera. El administrador de carrusel publica el directorio de carrusel actualizado a Open Streamer, controlando así las prioridades y pistas de la difusión de Open Streamer.

Open Streamer es un producto de software/hardware que permite a los operadores de red difundir aplicaciones y datos de la PS en su difusión de red. Open Streamer permite la transmisión de los datos y aplicaciones de la PS simultáneamente con los programas de A/V del operador de red. Open Streamer posibilita la actualización de un *stream* de datos en tiempo real para que coincida con el contenido de A/V. Por ejemplo, un operador de red puede difundir una aplicación interactiva de deporte junto con la difusión en directo de un acontecimiento deportivo. Open Streamer consta de dos componentes, un DLL de servidor común y un *streamer* de difusión. Un servidor de aplicaciones (por ejemplo, un servidor de aplicaciones meteorológicas) o el generador de carruseles en la PS invocan el archivo DLL de servidor común para que envíe los datos de carrusel al *streamer* de difusión. A continuación, el *streamer* de difusión lleva a cabo un multiplexado (de acuerdo con la relación código/velocidad y los requisitos de velocidad de bits) de las aplicaciones y los datos de A/V y envía los datos multiplexados al equipo de difusión para que se lleve a cabo su difusión.

Introducción general al esquema de protocolo DAP/DATP

La presente invención permite la comunicación entre los STB y los proveedores de servicios asociados con una PS. El protocolo DATP es un protocolo basado en mensajes en el que una entidad envía un mensaje a otra entidad con una garantía de entrega. Cuando el STB envía un mensaje a la SGW, el STB recibe un mensaje de confirmación en el momento en que el mensaje ha alcanzado su destino final (la SGW o un servidor de aplicaciones). Cuando un servidor de aplicaciones ha procesado el mensaje, se puede enviar un mensaje de respuesta al STB, siempre y cuando la sesión del STB con la SGW se encuentre aún abierta. La fase de transmisión del mensaje DATP estará precedida de una fase de inicio de sesión con DATP Y seguida de una fase de cierre de sesión con DATP, necesarias para establecer una sesión DATP. DATP en un protocolo orientado hacia sesiones. En la Figura 9 se ilustra un ejemplo sencillo de sesión DATP.

DATP admite sesiones múltiples encima de la conexión del mismo nivel de transporte del STB. Los clientes de STB pueden enviar, durante el desarrollo de una sesión abierta con la SGW, paquetes de inicio de sesión para iniciar una nueva sesión en el mismo enlace de transporte del STB utilizado para la primera sesión. Los dos módulos de administración de sesión DATP en el cliente STB yen la SGW se encargan de multiplexar los diferentes mensajes de sesión en el mismo enlace.

Introducción general al contenido de los paquetes DATP

Los paquetes de protocolo DATP comprenden un encabezado de tamaño fijo, una carga de datos de tamaño variable (mensajes DAML) y un finalizador. El encabezado se compone de los siguientes elementos: el número de versión del protocolo; el tipo de paquete (protocolo de enlace de inicio de sesión/cierre de sesión, ping, datos, confirmación, etc.); la información de transporte real (sin formato, TCP/IP, UDP, etc.); el número de secuencia de mensaje (número de mensaje DATP generado por STB o SG); el identificador de servicio (identificador del servicio que recibirá los datos). El identificador de servicio es un identificador de 8 bits que se define en el protocolo DATP. El identificador de sesión (proporcionado por SGW en el protocolo de enlace); los indicadores de cifrado para las sesiones cifradas; y el tamaño del área de datos [sic].

La información del área de datos puede contener los siguientes elementos, dependiendo del tipo de paquete: información de inicio de sesión/cierre de sesión para los paquetes de protocolo de enlace; información de confirmación para el paquete de confirmación; área de datos para el paquete de datos. El finalizador contendrá, por lo menos, la suma de comprobación CRC de 32 bits del paquete DATP. El ordenamiento de bytes del protocolo DATP es Big-Endian.

Especificación de los campos de los paquetes

El campo de versión del protocolo es la versión del protocolo DATP utilizado por la entidad transmisora. Es el primer byte de un paquete DATP. El formato de paquete DATP puede variar, dependiendo del número de versión del protocolo DATP. Cuando se especifican nuevas versiones de protocolo DATP, este número de versión se incrementa para reflejar los cambios. Las comunicaciones DATP entre dos entidades utilizarán la versión más alta de DATP disponible en ambas entidades. La negociación de versiones formará parte del proceso de inicio de sesión.

El campo de información de tipo de paquete constituye el segundo byte de un paquete DATP. En éste se indica el tipo de paquete DATP que se está enviando. El campo de, información de transporte del STB constituye el tercer byte de un paquete DATP. En el mismo se proporciona información sobre el transporte utilizado en el STB y se divide en tres subcampos: STB_transport_info [7..4]: los cuatro bits de MSB del campo representan el tipo de protocolo de transporte nativo del STB; STB_transport_info [3]: este bit indica si el transporte subyacente es fiable. Obsérvese que este bit se fija al valor correcto, aun cuando el valor del tipo de protocolo de transporte nativo puede proporcionar una buena indicación de la fiabilidad del protocolo; STB_transport_info [2..1]: este bit indica la clase de velocidad del transporte del STB nativo.

El identificador de servicio es el cuarto byte en un paquete DATP e indica el identificador del destino (paquetes desde el STB a la SGW) o del host transmisor (paquetes desde la SGW al STB) de un paquete DATP. El identificador de sesión es el segundo quadlet (doble palabra) de un paquete DATP. Éste indica el identificador de sesión del paquete DATP.

La SGW genera los valores de identificador de sesión durante el proceso de inicio de sesión. Se establece en O el campo de identificador de sesión de los paquetes de inicio de sesión. En DATP, un número de secuencia es la primera palabra del tercer quadlet de un paquete DATP. Ésta indica el número de secuencia del mensaje DATP. Dicho número identifica una "transacción" DATP desde un paquete enviado hasta su confirmación correspondiente. La entidad transmisora genera los números de secuencia del mensaje, los cuales son únicos solamente en los mensajes enviados en una fase de una conexión DATP. Ello quiere decir que un mensaje DATP enviado desde el cliente STB a la SGW y un mensaje enviado desde la SGW al cliente STB pueden tener el mismo número de secuencia pero pueden corresponder a dos "transacciones" independientes.

En DATP, el tamaño de los datos es la segunda doble palabra del tercer quadlet de un paquete. Éste indica el tamaño del área de datos del paquete en bytes. De acuerdo con su diseño, este tamaño está limitado a 64 KB para dar cabida a diversos factores comunes en los STB de gama baja, como por ejemplo enlaces por módem de baja velocidad, canales de comunicación sumamente ruidosos, recursos de memoria RAM limitados, etc. En DATP, los indicadores de cifrado constituyen el primer byte del cuarto quadlet de un paquete DATP. El área de datos DATP empieza desde el primer byte, después del encabezado de tamaño fijo de 16 bytes, hasta llegar al tamaño del área de datos, tal y como se indica en el campo de tamaño de datos de encabezado. En DATP, CRC es el primer quadlet después del área de datos y contiene el valor de la suma de comprobación CRC de 32 bits de la totalidad del paquete DATP (incluido el encabezado).

El cliente STB envía el paquete de inicio de sesión para iniciar una sesión DATP con la SGW. Éste representa la primera fase de la negociación del proceso de inicio de sesión, en la que el STB se presenta a sí mismo a la SGW. Si la solicitud es correcta, la SGW responde a una solicitud de inicio de sesión con un paquete de confirmación. La SGW tomará una decisión sobre los atributos negociables de la conexión DATP Y asignará un identificador de sesión a la sesión recién creada.

En caso de producirse un error, la SGW responderá a una solicitud de inicio de sesión con un paquete de confirmación negativa. El STB envía este paquete para cerrar una sesión DATP con la SGW. En caso de que la solicitud de cierre sea correcta, la SGW responderá a una solicitud de cierre de sesión con un paquete de confirmación de cierre de sesión.

En caso de que se produzca un error, la SGW responderá a una solicitud de cierre de sesión con un paquete de confirmación negativa de cierre de sesión. Entre los casos de error figuran identificador de sesión desconocido, CRC erróneo, etc. Cualquier entidad de una conexión DATP puede enviar un paquete de datos. Una aplicación cliente STB puede enviar paquetes de datos DATP a servidores de aplicaciones, los cuales pueden responder a un STB, forzando la transmisión de un paquete de datos desde la SGW al STB cliente. Una entidad que ha recibido un paquete de datos responderá, en caso de que la recepción sea correcta, con un paquete de confirmación de datos. Una entidad que ha recibido un paquete de datos responderá, en caso de que la recepción no sea correcta, con un paquete de confirmación negativa de datos. Si no se ha recibido paquete alguno desde una entidad remota DATP durante un periodo configurable de tiempo, la otra entidad remota podría someter a prueba el enlace DATP al enviar un paquete ping DATP Y esperar una respuesta. Una entidad remota que ha recibido un paquete ping debe enviar un paquete de confirmación ping a la otra entidad semejante remota, en caso de haberse recibido con éxito el paquete ping. Una entidad remota que ha recibido un paquete ping debe enviar un paquete de confirmación negativa ping a la otra entidad semejante remota, en caso de haberse producido una recepción errónea de un paquete ping. Entre los casos de error figuran identificador de sesión desconocido, CRC erróneo, etc.

Si nos fijamos en la Figura 3, a continuación presentamos de forma resumida la arquitectura de DATP/SGW mostrada en dicha figura. Un gran número de PS y aplicaciones de cliente STB poseen necesidades comunes que son más específicas del transporte que de la aplicación, y que la arquitectura de DATP satisface. DATP lleva a cabo operaciones de cifrado, compresión de datos, enrutamiento de HTTP y otras muchas funciones que se analizarán más adelante. La arquitectura para el marco de servicios de fondo para aplicaciones DATP se ilustra en la Figura 3. DATP proporciona HTTP ligero (LHTTP) en el nivel de aplicación de código objeto, función de almacenamiento y reenvío, identificación del STB (mediante el uso del Registro Central de OpenTV [OCR]) y muchas otras funciones. Estas funciones forman parte del protocolo DATP o están ubicadas encima del mismo.

Como se muestra en la Figura 3, el servidor de la SGW (1018) proporciona un enlace de comunicación robusto entre el STB (1008) Y una variedad de servidores de aplicaciones (1026, 1028, 1030 Y 1032), incluido el servidor de Fetchmail (1026). La SGW (1018) enruta las solicitudes transmitidas en ambas direcciones entre el STB y los servicios de aplicaciones. La SGW recibe paquetes DATP del cliente/STB (1018), contacta el servidor de aplicaciones apropiado y envía/recibe datos al servidor de aplicaciones a través de una conexión TCP/IP. La SGW permite a un servidor de un tercero, o a servidores específicos de la PS, como por ejemplo el servidor Fetchmail (1026), enviar mensajes al STB.

Como se muestra en la Figura 4, la arquitectura de pila del STB/cliente presenta una multiplicidad de módulos, así como una capa adicional, el administrador de mensajes (1104), entre la aplicación y el transporte de STB/cliente nativo. Se suministran API a las aplicaciones STB, como por ejemplo una API LHTTP (1106) y una API de almacenamiento y reenvío (1108). El servidor utiliza una versión asíncrona de la capa PAL, implementa grupos de subprocesos y técnicas de aislamiento de procesos.

En una modalidad preferida, DATP proporciona tamaños de mensaje incrementados, a la vez que garantiza la fiabilidad de las entregas y soluciona complejas cuesti0nes de memoria causadas por los entornos incrustados limitados en el STB. Con el fin de aumentar el tamaño del mensaje DATP, los mensajes de gran tamaño se dividen en fragmentos más pequeños y se transmiten, reorganizan y entregan en un mensaje DATP reconstruido. En un enlace no fiable con una frecuencia de errores binarios (BER) de 10^{-64}, la probabilidad de que un mensaje de 64 KB tenga un error es aproximadamente del 7% (1 mensaje de cada 14). Sabiendo que se tarda poco más de cinco minutos en transferir 64 KB por un módem de 2400 bits/s, DATP evita retransmitir el mismo mensaje durante otros cinco minutos simplemente porque uno de sus bits está dañado. Para evitar la retransmisión, a continuación se exponen las directrices de implementación preferidas para DATP.

En una modalidad preferida, los mensajes de gran tamaño, es decir, los mensajes superiores a 64 KB, son fragmentados en paquetes DATP más pequeños. Se pueden utilizar umbrales de fragmento inferiores a 64 KB. Cada fragmento DATP es confirmado independientemente. Como se muestra en la Figura 8, DATP realiza un seguimiento de los números de secuencia del mensaje y el momento en el que se utilizó por última vez el número de secuencia. Se rechazan los mensajes DATP que posean un número de secuencia usado "recientemente", puesto que se consideran "ya recibidos". Para implementar esta directiva, los hosts DATP mantienen una ventana deslizante de números de secuencia usados recientemente dotados de una marca de fecha y hora en cada número de secuencia. Los números de secuencia más antiguos son eliminados de la ventana del host remoto si son anteriores a (host_max_retry+l)*host_timeout. En una modalidad preferida el valor de tiempo de espera (*time out*) es programable y puede ser fijado a cualquier valor deseado.

La ventana con rechazo realiza un seguimiento de los números de secuencia de los paquetes recibidos en un margen de tiempo determinado, comenzando en el momento actual. Cuando la capa núcleo de DATP recibe un paquete, se busca en la ventana con rechazo su número de secuencia. Si se encuentra el número de secuencia en la ventana, éste es descartado, es decir, se omite el paquete o fragmento asociado con ese número de secuencia. Si no se encuentra el número de secuencia del paquete en la ventana, se añade el nuevo número de secuencia a la ventana. La ventana o "ventana con rechazo" se borra periódicamente para eliminar los números de paquete que sean anteriores a una fecha determinada, dependiendo del tiempo utilizado en el enlace de comunicación. El algoritmo de la ventana con rechazo de paquetes proporciona una protección eficaz contra las recepciones múltiples de paquetes idénticos que pueden producirse habitualmente con los protocolos de transporte orientados hacia mensajes fiables basados en retransmisión/tiempo de espera.

Los mensajes DATP se envían basándose en las condiciones de memoria del host remoto. Cada paquete confirmado de un mensaje DATP contiene un campo de memoria disponible que indica la condición de memoria actual de la entidad receptora. Una entidad remota que se sirve de DATP para enviar un mensaje a otra entidad semejante comprueba en primer lugar si el tamaño del mensaje DATP es más pequeño que la memoria disponible en la entidad receptora. Se envían los fragmentos del mensaje DATP al host receptor si existe suficiente memoria en la entidad receptora para recibir el mensaje. El host receptor confirma la recepción del mensaje en el momento en que lo recibe. De lo contrario, el host transmisor envía paquetes de control al host receptor para consultar sobre la disponibilidad de memoria del host remoto o receptor. También se puede implementar, si así se desea, una entrega parcial basada en la memoria disponible, que contiene únicamente una parte del mensaje. En este caso, se almacenan en la caché los mensajes parciales hasta que sean completados. Se envían los paquetes de control hasta que haya suficiente memoria disponible en la entidad remota o hasta que se haya superado el número máximo de intentos de envío. Si se sobrepasa el número máximo de intentos de envío y, aun así, todavía no hay memoria suficiente en el host receptor para completar la transmisión del mensaje, entonces se produce un error en la transmisión del mensaje (a menos que se haya autorizado la entrega parcial de mensajes).

El protocolo DATP es un protocolo basado en mensajes en el que una entidad envía un mensaje a otra entidad con una garantía de entrega. Cuando el STB envía un mensaje a la pasarela de servicios y éste alcanza su destino final (la propia pasarela de servicios o un servidor de aplicaciones), el STB recibe un mensaje de confirmación. Cuando un servidor de aplicaciones ha procesado el mensaje, es posible que se envíe un mensaje de respuesta al STB, siempre y cuando aún se encuentre abierta la sesión del STB con la pasarela de servicios. La fase de transmisión de un mensaje DATP estará precedida de una fase de inicio de sesión DATP y seguida de una fase de cierre de sesión DATP, necesarias para establecer una sesión DATP. Cabe destacar que los mensajes enviados a través de DATP se fragmentan en paquetes DATP de, como máximo, MTU (unidad media de transmisión) bytes que se transmiten y confirman independientemente. Ello permite a los mensajes DATP alcanzar el mayor tamaño posible que puede ser administrado físicamente por las entidades DATP. En la Figura 9 se ilustra un ejemplo sencillo de una sesión DATP.

DATP admite sesiones múltiples encima de la misma conexión de capa de transporte de STB. Los clientes STB pueden enviar, en medio de una sesión abierta con la pasarela de servicios, paquetes de inicio de sesión para comenzar una nueva sesión en el mismo enlace de transporte exacto de STB que están utilizando para la primera sesión. Los dos

módulos de gestión de sesión DATP en el cliente STB y en la pasarela de servicios se encargarán de multiplexar los diferentes mensajes de sesión en el mismo enlace.

Para apoyar la transmisión de mensajes DATP de gran tamaño, DATP se sirve de un sistema de fragmentación/reconstrucción de paquetes. Los mensajes de gran tamaño se fragmentan en paquetes DATP más pequeños, con un tamaño máximo de MTU. Cada host tiene un tamaño de MTU y cada entidad DATP puede tener uno diferente. Cada fragmento (paquete DATP) de un mensaje DATP es confirmado independientemente.

Se rechazarán todos los mensajes DATP con un número de secuencia usado "recientemente" para evitar condiciones de competencia del tipo "recepciones múltiples de fragmentos idénticos". Para implementar esta directiva, los hosts DATP mantienen una ventana deslizante de elementos usados recientemente (número de secuencia, identificador de fragmento) con una marca de hora y fecha en cada entrada de la ventana. Se eliminarán las entradas antiguas (número de secuencia, identificador de fragmento) de la ventana de un host DATP si son anteriores a (host_max_retry+l)*host_timeout.

Un tamaño de fragmento DATP predeterminado (es decir, tamaño de MTU) se limita a 4 KB para adaptarse a un entorno limitado de STH en el que la fragmentación de memoria constituye un problema. Se puede incrementar el tamaño de los fragmentos a un máximo de 64 KB a discreción de la aplicación.

DATP admite hasta 65.536 fragmentos por mensaje DATP. Ello proporciona un tamaño máximo de mensaje teorético de 4 GB. Un primer fragmento de un mensaje DATP proporciona un marcador que indica que este fragmento es un primer fragmento de un mensaje nuevo y su campo de identificación (id) de fragmento se establece como el número de fragmentos que componen este mensaje DATP.

Las entidades remotas deberían descartar los mensajes DATP incompletos después de (host_ max _ retry+ 1)*host_ timeout.

DATP proporciona cifrado para permitir a las aplicaciones devolver datos confidenciales a sus servidores de aplicaciones respectivos. Al proporcionar cifrado en el nivel de transporte se trata de resolver el reto de suministrar cifrado en el entorno de baja capacidad de procesamiento del STB o cliente. Por consiguiente, la cuestión de cifrado se aborda mediante un sistema de cifrado diseñado minuciosamente y una API segura preferida de DATP. Se proporciona seguridad/cifrado en el nivel de sesión. Las aplicaciones abren una sesión segura utilizando una API segura de DATP. En el inicio de sesión se negocian los parámetros de cifrado de DATP. Se suministra una sesión segura en por lo menos dos fases: durante una fase estándar de inicio de sesión DATP Y durante una fase de negociación de clave.

A continuación se presenta una breve descripción de los pasos principales de la fase de negociación de clave. El servidor DATP envía su clave pública server_epk a un cliente o STB. DATP preferentemente utiliza RSA de Rivest, Shamir y Adleman (tecnología de cifrado de clave pública), aunque es posible utilizar otros algoritmos. DATP elige el exponente RSA server_epk = (e, n), de forma que e=3 o mayor, mientras que se mantiene un alto nivel de seguridad (la seguridad depende únicamente de n). Con el fin de cifrar un mensaje con RSA el STB necesita calcular (m^e) mod n. Un valor pequeño para la "e" quiere decir que la fase de exponenciación será reducida, lo que producirá un cálculo más rápido del mensaje cifrado. El STB o cliente inicializa su generador de números aleatorios con la fecha y hora del sistema, más cualquier fuente aleatoria disponible para la capa de código objeto (por ejemplo: el fotograma de vídeo actual, etc.). El STB/cliente elige una clave secreta de STB/cliente (stb_sk). El STB cifra la clave secreta, stb_sk, con server_epk usando RSA. El STB envía una clave secreta cifrada, stb_sk, al servidor DATP. El servidor DATP descifra la clave cifrada stb_sk con su clave privada secreta server_dpk.

El servidor DATP (por ejemplo, SGW) inicializa su generador de números aleatorios y elige una clave de servidor secreta server_sk. El servidor DATP (por ejemplo, SGW) cifra server sk con stb_sk utilizando un sistema de cifrado de clave secreta. El servidor DATP envía server sk cifrado al servidor DATP. El STB descifra la clave cifrada server sk con su clave secreta stb_sk. Una vez que se han intercambiado correctamente las claves, las dos entidades pueden intercambiar datos cifrados secretos a través de DATP mediante el uso de las claves secretas mutuas. En una modalidad preferida, se añade un paso de autenticación de servidor DATP al protocolo para mejorar el sistema de intercambio de claves y fortalecerlo contra los ataques *Man-in-the-middle* ("hombre en el medio"). Por consiguiente, el protocolo DATP permite la firma de pilas DATP Y la administración de certificados de autenticación.

Para reducir al mínimo el tiempo de comunicación con la SGW, la clave pública del servidor se incrusta preferentemente en. la pila, de forma que el cifrado de la clave privada del STB se pueda realizar fuera de línea. Ello introduce una nueva cuestión de administración de claves, ya que el servidor DATP debería conocer la clave pública de servidor Utilizada por el STB o cliente. Los mensajes enviados mediante una sesión segura serán cifrados preferentemente en el nivel de fragmento. Esto quiere decir que cada fragmento individual de un mensaje DATP se cifrará independientemente.

Se proporciona una API segura de DATP con la capacidad de enviar mensajes no cifrados a través de una sesión segura de DATP, lo que proporciona a las aplicaciones de la PS la opción de ahorrarse ciclos de CPU al no tener que cifrar los datos no confidenciales enviados a través de una sesión segura. Ello resulta útil para los clientes o STB con una capacidad de procesamiento limitada, como por ejemplo el Motorola DCT 2000.

Una vez que se establece una sesión segura entre un servidor DATP Y un cliente DATP o STB, se descifran en primer lugar los mensajes enviados por el cliente/STB a cualquier servidor de aplicaciones en el servidor DATP (por ejemplo, SGW) y después se reenvían a los servidores de aplicaciones utilizando una conexión de *Secure Sockpt Layer* (SSL, Nivel de socket seguro). La capa de cifrado se basa en una biblioteca criptográfica disponible para los programadores de código objeto y para los programadores de servidores de aplicaciones. Las aplicaciones pueden utilizar esta biblioteca para administrar el cifrado en el nivel de aplicaciones. Esta capacidad puede resultar útil para administrar el cifrado de extremo a extremo que garantiza la seguridad de aplicaciones críticas, como por ejemplo las aplicaciones bancarias.

La compresión de datos en enlaces lentos, como por ejemplo los disponibles en la mayoría de STB y clientes (2400 a 33600 bps), es aconsejable para enviar datos comprimidos para incrementar el rendimiento total de la línea [sic]. En algunos casos, la compresión de datos por módem está disponible en el nivel de enlace OSI. Los protocolos de nivel más alto no obtienen beneficios al comprimir su área de datos. Un gran número de módems de clientes/STB no ofrecen una compresión en el nivel de enlace, por lo que los protocolos de nivel más alto proporcionan la compresión. La presente invención suministra compresión de datos en el nivel de servidor DATP.

El desafío consiste en que los STB o procesadores cliente carecen de la capacidad necesaria para llevar a cabo las búsquedas eficaces de patrones (u otras operaciones que implican el uso intensivo de CPU) requeridas por la mayoría de los algoritmos de compresión. Sin embargo, la descompresión es una tarea relativamente sencilla y se proporcionan API de descompresión al cliente/STB en el nivel de código objeto. Basándose en estas consideraciones, el apoyo de DATP para la compresión es asimétrico, es decir, preferentemente se comprime sólo el enlace descendente desde el servidor DATP al STB o cliente utilizando las herramientas habituales de compresión de la PS.

Los paquetes comprimidos DATP poseen un indicador de "datos comprimidos" en el encabezado del paquete que indica que las áreas de datos están comprimidas. Los encabezados de los paquetes no se comprimen. La compresión y descompresión utilizarán herramientas y API estándar de compresión y descompresión proporcionadas por la PS. El tamaño de los paquetes DATP indica el tamaño del área de datos comprimido. En el encabezado de compresión del área de datos se indicará el tamaño descomprimido del área. La compresión de mensajes DATP se realiza en el nivel de fragmento. Cada paquete DATP individual de un mensaje DATP es comprimido independientemente. Ello es preferible, ya que los fragmentos de mensaje DATP no se almacenan necesariamente de forma contigua cuando son recibidos; por lo tanto, se prefiere que DATP descomprima cada fragmento independientemente. La descompresión independiente es posible, ya que cada fragmento DATP se comprime por separado. La pila de STB DATP y la API del servidor de aplicaciones DATP pueden deshabilitar o habilitar la compresión de datos en el enlace descendente. Esta función proporciona a los servidores de aplicaciones al menos dos capacidades importantes: la capacidad de transferir una gran cantidad de datos a clientes o STB utilizando el canal de difusión de alta velocidad y la capacidad de enviar datos de multidifusión a una colección de clientes o STB a través del canal de difusión, permitiendo así un ahorro general de ancho de banda en la PS.

El servidor DATP proporciona un módulo de servidor de aplicaciones Open Streamer que administra un número configurable de *streams* de difusión. Estos *streams* se utilizan para enviar grandes segmentos de datos, así como datos de multidifusión a clientes y/o STB. La multidifusión se suministra como una característica tan importante como el enrutamiento en la difusión, ya que permite a los servidores de aplicaciones enviar datos a un grupo de STB sin dirigirse a cada STB individualmente. El apoyo de la multidifusión en DATP proporciona paquetes DATP no fiables. La PS mantiene una lista de los identificadores de sesión del grupo de multidifusión y administra los casos en que un STB o cliente sin un sintonizador de difusión disponible es miembro de un grupo de multidifusión.

El Servicio de Nombres (DNS) de DATP proporciona una asignación entre los nombres del servidor de aplicaciones y los identificadores de servicio. Aunque servicios bien conocidos han reservado identificadores de servicio, también hay disponible VD gran número de identificadores de servicio definidos por el usuario que diferentes aplicaciones pueden utilizar. Con el fin de evitar una "codificación dura" de los identificadores de servicio en las aplicaciones de STB o código objeto, las aplicaciones cuentan con la capacidad de hacer referencia a los servicios por nombre después de una fase de resolución de nombres. De esta forma la aplicación depende menos del archivo de configuración de la SGW.

A continuación se describe cómo se facilitan características de DNS a los clientes DATP. Desde un punto de vista de protocolo DATP, se considera que DNS es simplemente otro servicio más. Se reserva un identificador específico de servicio para el servicio DNS. El servicio DNS está alojado dentro de la SGW o puede estar alojado en otra ubicación en la PS o en un STB u otro cliente. El cliente DATP proporciona una API sencilla para resolver los nombres de servidores de aplicaciones. Preferentemente, la invocación principal (datp_get_asid_by_name (as_name)) devuelve un número de solicitud de forma sincrónica. Una notificación asíncrona devuelve el estado de la resolución de nombre, incluido el identificador de servidor de aplicaciones, en los casos con éxito. Es posible realizar resoluciones de nombres simultáneas sin consecuencias negativas significativas para el rendimiento. Los usuarios pueden enviar notificaciones de servidor de nombres basadas en un identificador de solicitud asociado con cada solicitud. El parámetro de nombre de los servidores de aplicaciones se añade al archivo de configuración actual de DNS. El mismo nombre no se utiliza para diferentes identificadores de servicio. Para obtener redundancia o satisfacer cuestiones de escalabilidad se admite el registro de varias máquinas por identificador de servicio.

En la modalidad preferida, se considera que DNS es un ejemplo de un servicio de directorio que aún no se ha definido. El formato de paquete de solicitud DNS comprende los siguientes campos: Tipo de consulta (en el que se indica el tipo de consulta (0 para consulta DNS, por ejemplo)); Etiqueta de consulta (etiqueta proporcionada por el usuario que se compara con las respuestas del servicio de directorio); Datos de consulta (datos utilizados para realizar la operación de consulta (normalmente el nombre del servicio para DNS)). El formato del paquete de respuesta DNS comprende los siguientes campos: tipo de respuesta (en el que se indica el tipo de respuesta (0 para resolución correcta de DNS)), etiqueta de respuesta (la misma que la etiqueta de consulta que generó la respuesta) y datos de respuesta (datos de la respuesta de la consulta (normalmente, el identificador del servicio para DNS)).

En una modalidad alternativa de DATP, se asume que todos los clientes DATP se encuentran detrás de un *modem rack* y, por cada cliente conectado, el servidor del terminal de *modem rack* abre una conexión dedicada de TCP/IP con la SGW y reenvía todo lo que recibe desde un STB determinado a esta conexión TCP. Con la posible implementación de receptores de televisión por cable de generación anterior que no admiten TCP/IP, pero sí el protocolo de datagramas de usuarios (UDP), el servidor DATP (por ejemplo, la SGW) proporciona la capacidad de escuchar en un puerto UDP. UDP se admite como se especifica a continuación. En el servidor, se crea una nueva clase datp_socket_listener para administrar las conexiones UDP. Se crea un capa de abstracción de tipo socket que incorpora los sockets UDP (PAL_udp_socket).

Las conexiones UDP se procesan como se describe a continuación. UDP_listener lee el nuevo datagrama de solicitud de conexión y crea un nuevo AL_udp_socket. UDP_listener contesta a la conexión, enviando el datagrama mediante el uso del recién creado PAL_udp_socket. UDP_listener crea un nuevo subproceso de administrador de sesión, pasando el recién creado PAL_udp_socket como atributo. El nuevo administrador de sesión contesta directamente al cliente DATP utilizando pal_udp_socket_send con el AL_udp_socket proporcionado. Obsérvese que no es necesario especificar la dirección remota del datagrama. UDP_listener establece la misma mientras responde a la solicitud de conexión.

En el lado del cliente se crea un módulo UDP, stb_transport, que implementa la API stb_transport ya especificada encima de cualquier API de UDP disponible en el STB o cliente destinatario. Este módulo UDP, stb_transport, preferentemente envía un datagrama de solicitud de conexión al puerto de escucha UDP en la SGW y espera hasta que recibe una respuesta de la SGW antes de notificar al núcleo de DATP que el enlace de transporte está activado. Se envían datagramas posteriores que utilizan el puerto especificado en la respuesta a la solicitud de conexión de la SGW.

Se suministra un enrutamiento HTTP para proporcionar una interfaz a la SGW con los servidores de aplicaciones estándares que utilizan servidores Web como su servidor de aplicaciones para usuarios (*front-end servers*). En este caso, DATP preferentemente no utiliza la API estándar del servidor de aplicaciones DATP que se suministra a los programadores de servidores de aplicaciones, sino que se comunica directamente con estos servidores de aplicaciones reenviando los mensajes DATP a su servidor Web de aplicaciones para usuarios utilizando el mecanismo HTTP POST (HTTPP). En este sistema, las aplicaciones del cliente y/o STB utilizan la API de DATP sin ser conscientes de que están comunicándose con un servidor HTTP.

Con el fin de ser compatible con HTTPP, se suministra un tipo de servidor de aplicaciones DATP. Todos los servidores de este tipo se suministran con una entrada adicional en el archivo de configuración del servidor de nombres para especificar su dirección URL de notificación. El módulo de comunicación del servidor de aplicaciones proporciona la capacidad de enviar mensajes DATP a servidores HTTP en función del tipo de servidor destinatario. Preferentemente, este módulo se divide en un administrador de comunicación de un servidor de aplicaciones (SA) y dos emisores de datos SA. Un emisor de datos SA envía datos a los servidores de aplicaciones compatibles con la API DATP de SA y otro emisor envía datos a los servidores de aplicaciones basados en HTTP. Las cookies HTTP recibidas del servidor HTTP se almacenan en la SGW y se vuelven a enviar al servidor HTTP según se vayan necesitando. Los mensajes DATP recibidos en una sesión DATP segura se reenvían a los servidores HITP mediante HTTPS. Los inicios y cierres de sesión DATP preferentemente no son anónimos, permitiendo de esta manera a la SGW controlar el acceso a los servicios interactivos de PS y ofrecer a los servidores de aplicaciones una forma de acceder a la identidad de un cliente conectado.

A continuación se describe en mayor detalle la identificación del STB o cliente como parte de DATP. Las pilas de DATP contienen un identificador de hardware único (HID) que depende del STB o cliente. En el caso de un STB, se obtiene este identificador de hardware del nivel de transporte del STB dependiente del STB/la red. El formato de HID es una cadena de caracteres de longitud variable. Los HID para una red determinada se almacenan en una lista de HID. El operador de red, a través de PS, actualiza la lista de HID desde su base de datos de clientes utilizando las API. En caso de que no se pueda conectar directamente con la base de datos de abonados del operador de red, la PS importa la información de abonado (incluido su HID) de un archivo plano.

Para establecer sesiones DATP, las pilas DATP del STB o cliente incluyen su HID dentro del paquete de inicio de sesión DATP. La SGW comprueba la validez del HID mediante un repositorio central. Una vez que el repositorio central borra el HJD, se concede acceso a la pila del STB. El IDD permite a la SGW determinar la identidad de un STB o cliente conectado. De forma similar a las cookies de HTTP, un HID no autentica de forma “exhaustiva” un STB o cliente remoto. Por consiguiente, la autenticación oficial de usuarios remotos se llevará a cabo, preferentemente, mediante aplicaciones que requieren una autenticación más robusta de otras aplicaciones similares remotas.

DATP proporciona LHTTP de funciones HTTP para programadores de aplicaciones en código objeto que les permiten interactuar con servidores HTTP remotos. Se proporciona LHTTP para permitir el desarrollo de aplicaciones basadas en HTTP similares a la Web. LHTTP completa la estrategia actual de H2O al ofrecer una interfaz de HTTP simplificada, independiente del sistema operativo, para las comunicaciones de canales de fondo entre el cliente, el operador de red y los servicios. La interfaz LHTTP se basa en la pila de DATP, encapsulando las solicitudes de HTTP en los mensajes DATP. Se asigna un identificador de servicio de DATP especial a la capa de LHTTP y a los mensajes DATP recibidos en este identificador de servicio, que son enrutados al servidor HTTP de destino utilizando un módulo específico de emisor de datos LHTTP en la SGW.

Preferentemente, se admite un conjunto limitado de comandos HTTP, que comprenden comandos GET y POST. Se pueden añadir comandos adicionales de HTTP a LHTTP. Las solicitudes de LHTTP se transforman en solicitudes reales de HTTP en la SGW. Las solicitudes de HTTP se realizan desde la SGW en nombre de los clientes de LHTTP. Las cookies se reenvían a los clientes de LHTTP. La SGW almacena en caché las cookies y mantiene una tabla de traducción de cookies a identificadores de sesión. La DNS responde a las solicitudes de resolución de HID de los servidores HTTP utilizando esta tabla de traducción. Los servidores HTTP preferentemente utilizan el HID para extraer información de usuario del servidor de registro central. LHTTP también proporciona una API segura, LHTTPS. Esta API se basa en la capa de cifrado de DATP. Las solicitudes de LHTTPS se traducen automáticamente a solicitudes de HTTPS en la SGW.

Se proporciona a la interfaz entre la SGW y los servidores de aplicaciones un enrutamiento por Protocolo Simple de Transferencia de Correo (SMTP) o simplemente un reenvío de mensajes por correo electrónico. Puede utilizarse esta interfaz para transacciones que no tienen lugar en tiempo real, en las que una aplicación envía mensajes DATP a servidores de aplicaciones basados en SMTP y estos mensajes se reenvían por correo electrónico a los servidores de aplicaciones destinatarios.

Con el fin de soportar el enrutamiento por SMTP se crea un tipo de servidor de aplicaciones DATP para los servidores de aplicaciones SMTP. Los servidores de este tipo poseen entradas adicionales en el archivo de configuración del servidor de nombres para especificar su dirección de correo, así como el asunto del correo electrónico de los mensajes reenviados. El módulo de comunicación del servidor de aplicaciones envía los mensajes DATP a los servidores de aplicaciones basados en SMTP, dependiendo del tipo de servidor destinatario. Se proporciona un módulo de emisor de datos de servidores de aplicaciones SMTP para soportar este tipo de transacción. Se adjuntan los mensajes DATP enviados a los servidores de aplicaciones de SMTP a correos electrónicos codificados MIME (Extensiones Multipropósito de Correo Internet) que contienen varias partes. La primera parte del mensaje contiene los identificadores de hardware de los emisores, así como el identificador de mensaje DATP de los mensajes que se reenvían. La segunda parte del mensaje contiene mensajes DATP codificados MIME.

Los mensajes DATP enviados a un servidor de aplicaciones SMTP son confirmados una vez que el administrador de sesiones descodifica el mensaje y este está listo para ser enviado por correo electrónico al servidor de aplicaciones destinatario. Se podrían producir errores relacionados con SMTP una vez que la SGW intenta realizar una entrega de correo electrónico del mensaje DATP al servidor de aplicaciones destinatario. Los mensajes que se envían utilizando la capa de cifrado DATP se reenviarían no cifrados al host final. También se admite el cifrado PGP para enrutar de forma segura los mensajes DATP a través de SMTP.

El servicio de almacenamiento y reenvío de DATP proporciona la funcionalidad necesaria para que las aplicaciones envíen mensajes que no sean en tiempo real a un servidor de aplicaciones específico. Se proporciona una biblioteca de almacenamiento y reenvío encima del DATP. La aplicación utiliza el módulo de almacenamiento y reenvío para enviar mensajes con diferentes limitaciones temporales, dependiendo de sus necesidades. Las limitaciones temporales van desde “tan pronto como sea posible”, “un momento específico”, “una circunstancia, suceso o mensaje específicos” hasta “cuando estemos conectados”, pasando por “después de un periodo aleatorio de tiempo”.

El módulo de almacenamiento y reenvío almacena los mensajes DATP no entregados en el sistema de archivos, junto con algunos atributos específicos (registro de hora, limitaciones temporales, identificador de SA destinatario, etc.). La ruta de almacenamiento del sistema de archivos es configurable, por lo menos en el momento de compilación, para albergar una red específica. Los mensajes no reenviados mientras que una aplicación determinada habilitada para almacenamiento y reenvío DATP se encuentra activa no se reenvían hasta que otra aplicación habilitada de almacenamiento y reenvío se activa. El módulo de almacenamiento y reenvío no modifica el contenido del mensaje DATP reenviado. El mensaje se reenvía sin modificación al servidor de aplicaciones destinatario.

Si nos fijamos a continuación en la Figura 4, en la misma se ilustra la arquitectura DATP de la pila del cliente, la cual comprende una multiplicidad de módulos. Los módulos situados por debajo de la línea 1121 están escritos en código nativo del cliente, mientras que los módulos situados por encima de la línea 1121 están escritos en código objeto. El módulo de HTTP ligero (1106) proporciona capacidades de HTTP ligero a las aplicaciones de código objeto y se implementa encima de la API DATP. El módulo de almacenamiento y reenvío (1108) proporciona capacidades de almacenamiento y reenvío a las aplicaciones de código objeto y se implementa encima de la API DATP. El módulo DNS (1110) utiliza el módulo de administrador de mensajes (1104) para proporcionar servicios de resolución de nombres DATP. El módulo de administrador de mensajes DATP (1104) proporciona el “frontal” (*front end*) de DATP. Todas las invocaciones de API relacionadas con mensajes pasan a través del módulo del administrador de mensajes DATP. Este módulo divide los mensajes en paquetes DATP Y reconstruye los mensajes a partir de los paquetes DATP. El

módulo núcleo de transporte DATP (1102) administra las sesiones DATP, envía y recibe paquetes DATP Y administra la recepción de módulo DATP de la difusión. El módulo de extensión de transporte seguro DATP (1120) administra las sesiones seguras DATP. La biblioteca de paquetes DATP (1134) proporciona la funcionalidad para leer (analizar) y escribir (componer) paquetes DATP al módulo de transporte del STB DATP (1132), basándose en la especificación de formato de paquetes DATP. Este módulo, cuando lee un paquete completo DATP, notifica al núcleo de transporte DATP con el paquete analizado DATP.

La biblioteca de difusión DATP (1126) se dedica a escuchar en *streams* seleccionados de la PS basados en las especificaciones de núcleo de transporte DATP (1102), esperando a los módulos destinados a un STB o cliente determinado y notificando al núcleo de transporte DATP (1102) con los módulos DATP analizados. El módulo de transporte de STB DATP (1132) proporciona una interfaz de paquete en el nivel de enlace encima de cualquier transporte nativo o enlace de datos que esté disponible en el host DATP. El stub de bucle de suceso (*event-loop stub*) (1116) proporciona una versión de stub de la API del mensaje especificada en la capa de portabilidad de DATP. Este stub se basa en el bucle de suceso de la biblioteca común. El papel de la capa de portabilidad (1114) es abstraer la pila de DATP de las cuestiones dependientes de la aplicación, como por ejemplo el mecanismo de distribución de mensajes, las API de cifrado, etc. El stub de la biblioteca cifrada (1118) es una versión de stub de la API cifrada especificada en la capa de portabilidad de DATP. Este stub se basa en el paquete de cifrado de la biblioteca común. El stub de la biblioteca de módulo (1124) es una versión de stub de la API de descarga de módulo multipista especificada en la capa de portabilidad de DATP. Este stub se basa en el paquete de descarga del módulo de multipista de la biblioteca común.

Si nos fijamos a continuación en la Figura 6, DATP es un subconjunto del Protocolo de Aplicación de Televisión Digital (DAP). En la Figura 6 se ilustra DAP/DATP. DAP es utilizado para normalizar las comunicaciones de canales de fondo entre las aplicaciones de la PS y la SGW. DATP y la SGW proporcionan un mecanismo de transporte virtual genérico a las aplicaciones de la PS, ya que no todos los STB habilitados en la PS proporcionan una extensión de pila de TCP/IP. Además, algunos STB poseen su propia pila privativa o no proporcionan ninguna pila de comunicación.

DAP es un conjunto de protocolos sencillo y ligero para aplicaciones. El objetivo principal de DAP es proporcionar una forma sencilla y eficaz de potenciar los protocolos de aplicación existentes - como por ejemplo POP3, SMTP, el protocolo de acceso a mensajes de Internet (IMAP) y otros - en los STB de gama baja. Los STB a menudo poseen una capacidad baja de procesamiento y/o protocolos de comunicación privativos. DAP está concebido para que abstraiga la complejidad de las comunicaciones de los proveedores de aplicaciones, de forma que se consiga una mejor utilización de la infraestructura existente de redes para los estándares de aplicaciones actuales.

Como se muestra en la Figura 6, DAP se divide en dos partes: DAML (1610), un metalenguaje de aplicaciones para televisión digital y DATP (1620), un protocolo de transporte para aplicaciones de televisión digital. DAML (1610) es un metalenguaje que abarca muchas aplicaciones de PS. Cada aplicación de PS posee su propio dominio de DAML. La aplicación cliente responde y solicita mensajes encapsulados en un dominio de DAML. Los servidores de aplicaciones traducen estos mensajes de solicitud al protocolo apropiado para las aplicaciones existentes, como por ejemplo SMTP o IMAP.

DATP (1620) es un protocolo de transporte ligero y sencillo diseñado para aplicaciones con un ancho de banda reducido en las que TCP/IP u otro protocolo conocido no está disponible. DATP ha sido diseñado para interactuar con los protocolos de comunicación existentes en los STB actuales. DAP comprende: DATP, DAML-Mail (dominio XML para correo), DAML-Regi (dominio XML para registro de cuentas) y DAML-Acct (dominio XML para acceder al sistema VMS/AMS de la PS).

Los STB típicos se basan en una arquitectura de cliente delgado, es decir, poseen una capacidad de procesamiento mínima. Los servicios proporcionados por los STB actuales son a menudo de gama baja, aplicaciones “bobas” sin mucha capacidad de procesamiento. Las aplicaciones actuales que utilizan una gran cantidad de recursos, como por ejemplo las de correo electrónico, charla y navegadores de Internet, requieren dispositivos con una mayor capacidad de procesamiento. Los STB actuales no pueden proporcionar esta capacidad de procesamiento, de ahí la necesidad de un protocolo de aplicación ligero y de gama baja. DAP es lo suficientemente sencillo como para esconder o abstraer la complejidad de red de cliente/servidor del programador de aplicaciones.

DAP es modular, flexible y adaptable a las arquitecturas emergentes de software actuales. Éstas pueden ser un modelo basado en CORBA (Arquitectura Común de Intermediarios de Peticiones de Objetos) (*Object Management Group*) o un modelo COM (Módulo de Objeto Común)/DCOM (Módulo de Objeto Componente Distribuido). DAP es lo suficientemente flexible como para dar cabida a sistemas antiguos de otros fabricantes y poderse integrar con los mismos. DAP proporciona una interfaz para diferentes protocolos abiertos y privativos. Estos protocolos existen para los sistemas de servicios en los que el PC es el cliente principal, por ejemplo servicios IMAP o POP3. DAP utiliza la tecnología de software intermedio (*middleware*) de PS. El software del servidor DAP traduce el protocolo DAP a los protocolos específicos de las aplicaciones existentes.

DAP y su subconjunto DAML (1610) están diseñados para ser ligeros y capaces de soportar STB sensibles a ancho de banda de gama baja. Las etiquetas DAML preferentemente tienen un tamaño no superior a 4 caracteres y, cuando es posible, se limitan a 2 ó 3 caracteres. DAML incorpora XML binario para facilitar etiquetas DAML. Se utiliza DAP como un protocolo de comunicación entre las aplicaciones que funcionan en el STB y los subsistemas

de servicios. DATP (1620) controla el protocolo de enlace de la comunicación, el enrutamiento y la autenticación especifica del transporte, en los que DAML administra los requisitos específicos de la aplicación. Las solicitudes y respuestas de DAML se comunican entre un cliente STB y un proveedor de servicios a través de un protocolo existente de comunicación, por ejemplo TCP, UDP, DATP o un protocolo de comunicaciones privativo.

El protocolo DAP y su subconjunto DAML pueden constituir un conjunto de protocolos orientados hacia sesión o un conjunto de protocolos "sin sesión". Los dominios DAML son dependientes de aplicaciones. Se pueden utilizar nuevos dominios de protocolo DAP para nuevos tipos de aplicaciones. La adición de nuevos dominios DAP apenas tiene efecto en los dominios DAP existentes. Por consiguiente, DAP proporciona una PS única y simplista para que los operadores de red añadan servicios adicionales sin que ello tenga un impacto en los servicios existentes. Cada dominio DAML puede basarse en una etiqueta simplista legible por humanos o en una etiqueta abreviada cifrada que incrementa el rendimiento del protocolo, ya que disminuye el tamaño de paquete cuando el rendimiento es un factor crítico.

A continuación proporcionaremos una idea general de la función de DAML en la arquitectura DAP. DAML es un protocolo de comunicación del nivel de aplicación que se utiliza para especificar la conducta de comunicación y los datos de comunicación para los servicios de televisión interactiva. El protocolo de comunicación del nivel de servicio se encuentra por encima del protocolo de nivel de transporte y define cómo el contenido específico de la aplicación queda encapsulado entre las comunicaciones de cliente/servidor.

DAML es una colección de protocolos específicos de dominio que permite un diseño modular de la PS. Por ejemplo, DAML-Mail es un subconjunto de DAP. DAML-Mail es un protocolo de correo específico de dominio de correo. Se pueden añadir nuevos protocolos específicos de dominio como un subconjunto de DAP simplemente mediante la creación de nuevos DTD. DAP especifica las conductas de comunicación a través del envío y recepción de mensajes DAP. Los datos específicos de aplicación se encapsulan en un formato XML. La sintaxis de cada dominio de aplicación XML especifica las acciones que deben realizar los servidores de aplicaciones. Ello permite el diseño de protocolos simplistas muy ligeros que los STH actuales pueden utilizar para interactuar con la infraestructura existente, como por ejemplo los servicios de SMTP e IMAP.

DATP es un protocolo de nivel de transporte/servicios que proporciona una plataforma de comunicación entre la SGW y múltiples STB o clientes. DAML está encapsulado en un paquete DATP. En general, los protocolos de nivel de servicios se encuentran por encima de los protocolos de transporte, pero DATP es único porque puede residir en un modelo de red típico, ya sea en el nivel de servicios, de enlace de datos o de transporte. Ello aporta a DATP una gran flexibilidad. DATP interactúa con los protocolos de transporte subyacentes, como por ejemplo TCP, UDP, X.25, *raw socket* u otros protocolos.

SGW proporciona enrutamiento y tecnología SGW que permite a STB de gama baja conectarse a una infraestructura de red de servicios de fondo. La SGW admite protocolos de nivel de transporte entre los STB/clientes y la SGW, por ejemplo un protocolo de transmisiones secuenciales a través de *raw sockets*. DAML hace uso de esta característica.

DAML-Mail es un subconjunto de protocolo de DAP. DAML-Mail es un protocolo de correo específico de dominio. Este protocolo se utiliza para enlazar STH con servicios IMAP, POP3 y SMTP. DAML-Regi es un protocolo de dominio de servicios DAP que especifica un método genérico para el registro de cuentas para servicios múltiples. DAML-Regi es un protocolo sencillo entre un STB y el servidor de registro. DAML-Regi permite la interacción compleja entre un STB y una variedad de diferentes sistemas de aplicación con un único punto de integración, el servidor de registro.

DAML-Acct es un protocolo de dominio de servicio DAP que se comunica con la base de datos VMS/AMS de la PS. DAML-Acct permite al STB/cliente consultar y devolver datos específicos de usuario del sistema VMSI AMS. Todos los dominios DAML se definen utilizando sintaxis de definición de tipo de documento (DTD) de XML. DTD describe la sintaxis del mensaje pero no la lógica para los intercambios de solicitudes y respuestas. XML resulta útil a la hora de definir el formato de un bloque de texto. Las solicitudes y respuestas específicas DAML son interacciones relacionadas entre sí. Las reglas para su interacción están modularizadas en los componentes del STB y del servidor de aplicaciones.

El administrador de mensajes proporciona diferentes tipos de comunicaciones mediante mensajes entre los usuarios y también con las personas ajenas al sistema (aquellos que no están abonados a los servicios de la red). Por ejemplo, permite a los usuarios enviar y recibir correo electrónico, charlar con otras personas no abonadas y recibir mensajes instantáneos de éstas. La parte de correo electrónico del administrador de mensajes contiene un componente de Fetchmail conectado a un servidor de correo electrónico basado en Internet, como por ejemplo IMAP, POP3 y otros mensajes Webmail para el servidor apropiado que aloja correo.

Fetchmail administra toda la gestión de correo en la parte del servidor de la PS. Fetchmail traduce los mensajes DAP a mensajes IMAP, POP3 o Webmail para el servidor apropiado que aloja correo. SGW enruta mensajes de correo DAP a "Fetchmail" para su procesamiento. Fetchmail responde con la respuesta apropiada a la solicitud. Fetchmail se comunica con los servidores IMAP. La PS proporciona una aplicación de correo electrónico. Todas las aplicaciones de la PS pueden "enviar" correo electrónico a través del servicio de correo electrónico ofrecido por la SGW.

El servicio de la PS de charla interactúa con un servidor de charla u otro alternativo que incluye un servidor de charla. El servicio de charla es accesible a través de una aplicación de charla dedicada y también desde una aplicación de la PS que enlace con el DLL del cliente de charla de la PS. Al proporcionar una interfaz entre una charla y un listado de programas, es posible crear dinámicamente un salón de charla con un programa de difusión. Las aplicaciones y otros servicios pueden utilizar el servicio de "alerta" de PS para activar miniaplicaciones residentes en el STB. La alerta utiliza la extensión OMM de la PS y la funcionalidad de Open Streamer. El servicio de correo electrónico utiliza avisos de alerta para informar al telespectador de un mensaje entrante.

En la Figura 10 se describe la máquina de estado de conexión DATP. Esta máquina de estado describe la conducta de una máquina de estado de conexión de sesión DATP desde el lado del cliente del STB. Una sesión debe encontrarse en un estado de conexión antes de enviar cualquier paquete de datos a la SGW y debe estar desconectada después de una solicitud de cierre de sesión por parte del cliente. En la Figura 11 se describe el estado de envío del mensaje. Esta máquina de estado describe cómo una entidad DATP envía un paquete de datos DATP a otra entidad DATP. En la Figura 12 se describe la máquina de estado de recepción de mensaje. Esta máquina de estado describe cómo una entidad DATP recibe un paquete de datos DATP de otra entidad DATP. En la Figura 13 se describe la máquina de estado de mantenimiento de conexión. Esta máquina de estado describe cómo una entidad DATP debería utilizar los paquetes ping para garantizar que un enlace DATP con otra entidad aún se encuentra activo. Si el enlace no se encuentra activo se inicia un cierre de sesión.

SGW

Por lo que respecta a la Figura 5, la SGW incorpora una multiplicidad de módulos que son compatibles con características DATP. La arquitectura de la SGW es una arquitectura basada en multiprocesos que proporciona grupos de subprocesos. El servidor entero funciona sobre una versión asíncrona de una capa de abstracción de plataforma (PAL, *Platform Abstraction Layer*). La PAL implementa un proceso de cola de mensajes. La PAL se comunica utilizando técnicas de transmisión de mensajes. La SGW utiliza tres tipos de procesos, como se muestra en la Figura 5.

Como se puede observar en la Figura 5, los servidores de aplicaciones o los servicios se comunican con múltiples clientes/STB a través de la SGW utilizando un protocolo DAP específico del dominio. En determinados casos, los clientes/STB pueden conectarse directamente con los servicios de aplicaciones. Por ejemplo, si el protocolo de transporte entre el STB y la red es TCP/IP, el STB está habilitado para TCP/IP y no existe ningún requisito para llevar a cabo servicios comunes complejos proporcionados por la SGW, es posible mejorar el rendimiento de red más rápido a través de una comunicación directa del cliente/STB a un servicio a través de TCP/IP.

Por lo que respecta a la Figura 7 (el servidor DATP), el proceso principal de la SGW es el proceso principal del servidor DATP descrito anteriormente. La SGW aloja varios módulos clave. El módulo de socket de escucha TCP (1204) es un sencillo subproceso de socket de escucha TCP que se dedica a esperar conexiones en el puerto de escucha TCP DATP, las acepta y solicita la creación de administradores de nueva sesión para gestionar las nuevas conexiones. El socket de escucha UDP (1202) espera en un puerto conocido a la creación de conexiones UDP. Cuando se recibe una solicitud de conexión, el socket de escucha UDP (1202) crea un nuevo socket y envía una confirmación de solicitud de conexión al host remoto. El socket de escucha UDP (1202) solicita a continuación la creación de un administrador de nueva sesión para gestionar la conexión.

El módulo de monitor de administrador de sesión (1206) forma parte del subproceso principal. El papel principal de este componente es realizar un seguimiento de la población de los procesadores del administrador de sesión (SM, *session manager*) (1214) (la creación y eliminación de los procesadores SM basándose en la carga) y reenviar las solicitudes de creación del administrador de sesión al procesador de SM menos ocupado (1215). Cada procesador SM (0-n) (1215) comprende un módulo de comunicación con el servidor de aplicaciones DATP (ASCM) (1217) y un emisor de datos de servidor de aplicaciones independiente (ASDS) para DATP, HTTPP, LHTTP Y SMTP.

El subproceso del servidor de nombres DNS (1212) mantiene una tabla de correspondencias entre los identificadores de servidor de aplicaciones y sus atributos (nombre de host, puerto, tipo, etc.), así como una tabla de correspondencias entre los identificadores de sesión y los identificadores de cola de mensajes del administrador de sesión. El módulo del servidor de nombres, DNS, responde a las consultas de resolución de nombres enviadas a su cola de mensajes. El subproceso de socket de escucha del servidor de aplicaciones (1208) es responsable de esperar solicitudes de envío de mensajes procedentes de los servidores de aplicaciones. El servidor de nombres (1212) reenvía a continuación las solicitudes de envío a los administradores de sesión destinatarios, basándose en el identificador de sesión de solicitud de envío.

El proceso del procesador de administrador de sesión (1214 y 1216) alberga un grupo de subprocesos de administrador de sesión (1215). Se crean nuevos subprocesos de administrador de sesión basados en las solicitudes del monitor del administrador de sesión (1206) al subproceso del procesador del administrador de sesión. El subproceso del procesador del administrador de sesión (1214 y 1216) acepta las solicitudes de los procesadores del administrador de sesión (1214 y 1216) Y crea o elimina administradores de sesión, basándose en las solicitudes del monitor SM, notificando al procesador de administrador de sesión el resultado de sus solicitudes. Los subprocesos del administrador de sesión (1215) administran las sesiones DATP Y reenvían los mensajes DATP desde los STB o clientes a los servidores de aplicaciones y viceversa. Habrá un subproceso por cada STB o cliente. Estos subprocesos utilizan diversos módulos clave para controlar las sesiones DATP: la biblioteca de paquetes; el módulo de comunicación con el servidor

de aplicaciones; el emisor de datos al servidor de aplicaciones DATP; el emisor de datos al servidor de aplicaciones HTTPP; el emisor de datos al servidor de aplicaciones LHTTP; y el emisor de datos al servidor de aplicaciones SMTP.

El proceso de administrador de difusión (1210) es el componente principal del enrutamiento DATP a través de la difusión. Este proceso es un servidor de aplicaciones Open streamer que administra los carruseles del servidor DATP. El proceso de administrador de difusión actualiza estos carruseles de forma dinámica, dependiendo de las solicitudes que recibe de otros componentes del servidor DATP.

La PS y la SGW admiten preferentemente un sistema de procesamiento de datos Sun Solaris 7 dotado de memoria, monitor, GUI, mouse, teclado y procesador, bien conocido en este campo y que comercializa Sun Microsystems. La SGW funciona en un daemon UNIX, se configura gracias a un archivo de configuración y se inicia a partir de la línea de comandos. Una vez que se ha establecido una conexión entre la SGW y el STB/cliente en una red, TCP/IP controla todas las comunicaciones entre los otros servicios. Además de controlar-diferentes protocolos de transporte, la SGW también enruta mensajes a diferentes subsistemas de servicios, dependiendo de la configuración de la SGW.

La SGW lleva a cabo sus funciones en el punto de entrada de los servidores de aplicaciones. Ello permite que las características se configuren y/o añadan fácilmente, ya que la funcionalidad de los mensajes y la red están aisladas en la SGW. De esta forma se liberan los subsistemas de servicios para que puedan funcionar en funcionalidades de aplicaciones núcleo y se dejan las cuestiones de conectividad de la red a la SGW. Ello también permite una mayor esca1abilidad al aislar la funcionalidad específica en hosts independientes: [aísla] la entrega y recepción de mensajes de correo electrónico (usando el servidor Fetchmail) del enrutamiento y seguridad de la red utilizando la SGW.

La SGW posee un tamaño adecuado para admitir cientos de conexiones simultáneas en un solo servidor. La SGW es configurable para gestionar más conexiones, dependiendo de la capacidad de procesamiento del procesador que aloja la SGW. Este límite se basa en el número de módems (normalmente varios centenares) por POP (Punto de Presencia) para los ISP principales. En el caso de una arquitectura WAN, en la que la SGW está ubicada en un punto central, se suministra un dispositivo de equilibrio de carga basado en la traducción de direcciones de red (NAT) de hardware con el fin de conectar diversas SGW en paralelo y distribuir la carga.

Traducción de contenido - H2O

A continuación se presenta una introducción general del entorno de proxy H2O mediante una vista lógica de la arquitectura de H2O y varios ejemplos de transacciones. Las solicitudes de direcciones URL pueden proceder de diferentes componentes de H2O, por ejemplo STB/SGW y Carrusel. La siguiente introducción del contexto centra su atención en el STB/SGW que realiza las solicitudes, aunque el flujo general de información permanece igual.

Un telespectador decide interactuar con una página web por televisión, por lo que envía una solicitud desde el STB al sistema H2O y espera su respuesta. Las solicitudes de STB se envían a la SGW utilizando solicitudes de HITP ligero (LHTTP) encapsuladas en mensajes DATP como protocolo de transporte. El objeto solicitado se devuelve a través del mismo canal y protocolo. La SGW convierte el protocolo LHTTP a HTTP estándar a través de TCP/IP y envía la solicitud a una caché web.

La caché de objeto compilado (COC) utiliza su espacio de disco interno para ocuparse de cualquier solicitud que pueda atender (en función de una técnica heurística que tiene en cuenta el tiempo de vida de los objetos). Su función es atender a todos los objetos estáticos (direcciones URL estándar sin consultas y sin formulario enviado) sin consultar el proxy H2O, reduciendo así su carga de procesamiento. En esta arquitectura, la COC únicamente almacenará objetos compilados (módulos H2O). La máquina de COC está controlada por E/S.

Si centramos nuestra atención a continuación en la Figura 14, el proxy H2O (248) proporciona un entorno escalable para que puedan funcionar los diferentes compiladores de H2O (o filtros). Procesa solicitudes y respuestas HITP "sobre la marcha" y, por consiguiente, la máquina proxy H2O está controlada por procesos. El compilador HTML H2O (1420) está a cargo de la compilación de recursos HTML a PS. Para permitir el procesamiento del contenido en formato televisivo (1422), este componente realiza solicitudes HTTP por sí mismo, basándose en el tamaño de las imágenes incrustadas. El compilador reorganiza la imagen basada en la Web para ajustarla al dispositivo de visualización del cliente, por ejemplo un televisor.

El compilador MPEG (1426) es responsable de la conversión de un formato de imágenes web normales a recursos MPEG H2O PS. El formato de origen incluye JPEG y GIF Y puede incluir PNG. Argumentos que pasan a través de la dirección URL pueden controlar el proceso de conversión. El compilador PIXMAP es responsable de la conversión de imágenes web normales a recursos H2O PS. El formato de origen comprende GIF y puede incluir PNG.

El revisor de solicitudes (*Request Patcher*) (1424) es responsable de completar o modificar la solicitud o respuestas para incorporar los datos procedentes de otro sistema (por ejemplo, el número de tarjeta de crédito, etc.). Se comunica con un proceso o base de datos externos para buscar información de clientes. El componente PS proporciona un repositorio central de información de usuarios. El revisor de solicitud se comunica con este componente para obtener los datos necesarios e incorporar revisiones a las solicitudes/respuestas.

La caché de objetos no compilados (1430) utilizará su espacio de disco interno para ocuparse de cualquier solicitud que pueda atender (en función de una técnica heurística que tiene en cuenta el tiempo de vida de los objetos). Los objetos en caché comprenden HTML estático, imágenes GIF, imágenes JPEG y todos los archivos de formatos web normales. Su función es atender a todos los objetos estáticos (direcciones URL estándar sin consultas y sin formulario enviado) sin necesidad de realizar consultas en Internet, reduciendo así el periodo de latencia para obtener un objeto y proporcionando un tipo de tolerancia a errores para el sistema. El sitio web del cliente contiene el sitio web que se publica a través del sistema H2O.

En la Figura 15 se ilustra una solicitud de página estática que ya se encuentra en la caché. El usuario del STB envía una solicitud para cargar una página HTML (1520). Esta solicitud se envía a la SGW (248) utilizando LHTTP a través de DATP. La SGW convierte la solicitud a HTTP a través de TCP/IP y la reenvía (1522) a la caché de objetos compilados (1410). La caché de objetos compilados (1410) tiene la página HTML solicitada (compilada a un módulo) almacenada en su espacio de disco duro interno; si el tiempo de vida del objeto no ha expirado la caché de objetos compilados responde a esta solicitud con la página HTML compilada. Transmite la respuesta en HTTP (1424) a la SGW utilizando HTTP a través de TCP/IP. La SGW traduce el protocolo de HTTP por TCP/IP a LHTTP por DATP. El STB carga la página solicitada (1526) (compilada) en su memoria y se la entrega al motor del navegador de H2O para su interpretación. El motor del navegador de H2O solicita (1528) a la SGW que obtenga las imágenes necesarias para procesar la pantalla en televisión, con opciones de conversión (mpeg o pixmap, ancho, altura, etc.) en la dirección URL. La SGW transmite la solicitud de HTTP (1530) a la caché de objetos compilados. La caché de objetos compilados tiene la imagen solicitada (compilada a un módulo) almacenada en su espacio de disco duro interno; el tiempo de vida del objeto no ha expirado y la caché de objetos compilados atiende (1532 y 1534) esta solicitud con la imagen compilada. En este escenario, el proxy H2O no necesita procesar esta solicitud y, por consiguiente, puede dedicarse a procesar otras solicitudes.

Como se muestra en la Figura 16, el usuario del STB (212) envía una solicitud (1610) para cargar una página HTML (home.asp), la información de usuario y de host del encabezado de la solicitud contienen [el modelo + el número de serie del STB] y [el identificador de la tarjeta de acceso] del usuario. Esta solicitud (1610) se envía a la SGW utilizando LHTTP a través de DATP. La SGW convierte la solicitud a HTTP a través de TCP/IP y la reenvía (1612) a la caché de objetos compilados. El objeto solicitado no está disponible en el espacio de disco de la caché web. A continuación la caché web reenvía la solicitud (1614) al proxy H2O. El proxy H2O pide (1616) a la PS que devuelva (1620) el nombre del usuario (para el servicio amazon.com). El proxy H2O incorpora revisiones a la solicitud con el nombre del usuario y envía esta solicitud (1622) a la "caché de objetos no compilados". La "caché de objetos no compilados" no contiene la página HTML solicitada en su espacio de disco y envía la solicitud (1624) al servidor web de destino, en este caso amazon.com. El servidor web de destino procesa la página HTML, en función de la información de usuario, y la devuelve (1626) a la "caché de objetos no compilados". La "caché de objetos no compilados" devuelve la página HTML (1628) al proxy H2O.

El proxy H2O envía la solicitud HTTP (1630) a la "caché de objetos no compilados" con el fin de obtener las imágenes (1632, 1634 Y 1636) necesarias para sus cálculos de diseño (GIF, JPEG, etc.). El proxy H2O compila la página HTML, procesa el diseño, incorpora revisiones a las direcciones URL de las imágenes incrustadas y devuelve a la "caché de objetos compilados" el recurso resultante de OpenTV (1646) (con un tipo MIME de recurso PS). La caché de objetos compilados almacena el objeto en su espacio de disco interno y devuelve la página HTML compilada (1648) a la SGW. La SGW convierte la respuesta a LHTTP a través de DATP Y la devuelve al STB (1650). El STB carga el objeto solicitado en su memoria y lo pasa al motor del navegador H2O para su interpretación.

El motor del navegador H2O envía solicitudes (1652) a la SGW para obtener las imágenes necesarias para el procesamiento (a través de las direcciones URL revisadas: en este caso la URL logo.gif incluye una directiva para un formato de recurso pixmap): pix/logo.gif. La SGW convierte la solicitud (1652) a HTTP a través de TCP/IP y la reenvía al caché de objetos compilados. La "caché de objetos compilados" ya posee la imagen GIF solicitada en el formato de recurso correcto - puesto que un usuario ya solicitó esta imagen previamente - y la imagen se devuelve directamente (1654) a la SGW. La SGW convierte la respuesta a LHTTP a través de DATP y la devuelve (1656) al STB. El motor de navegador H2O envía solicitudes (1658) a la SGW para que obtenga las imágenes necesarias para el procesamiento de la imagen: mpg/banner.jpg. La "caché de objetos compilados" no contiene la imagen solicitada en su espacio de disco y por tanto envía la solicitud (1660) al proxy H2O. El proxy H2O envía la solicitud HTTP (1662) a la "caché de objetos no compilados" para obtener la imagen/banner.jpg.

La "caché de objetos no compilados" contiene la imagen en su caché y la devuelve (1664) inmediatamente al proxy H2O. El proxy H2O convierte la imagen, utilizando para ello los parámetros proporcionados en la dirección URL (formato MPG, ancho, altura, etc.) y devuelve el resultado a la caché de objetos compilados (1668). La caché de objetos compilados almacena el objeto en su espacio de disco interno y devuelve (1668) la imagen MPEG convertida a la SGW. La SGW convierte la respuesta a LHTTP a través de DATP Y la devuelve (1670) al STB. El STB procesa la página HTML y la reproduce en la pantalla.

El componente de proxy H2O proporciona a otros componentes o compiladores H2O una arquitectura robusta y escalable y una interfaz para la configuración de "compiladores". Entre los otros servicios que se proporcionan figuran: la capacidad de registrar errores, la capacidad de alertar a un administrador sobre eventos definidos y la capacidad de realizar una depuración-seguimiento de los "compiladores". Desde el entorno de proxy H2O suministrado y las API, los compiladores "incorporan revisiones" a las solicitudes y respuestas HTTP sobre la marcha, accediendo en última

instancia para ello a una base de datos, un archivo o un proceso externos. Los compiladores incorporan revisiones en las solicitudes de HTTP mediante las siguientes acciones: la eliminación del encabezado específico de HTTP (identificador de STB, identificador de tarjeta de acceso, etc.); la adición de un encabezado de HTTP específico (nombre de usuario, número de tarjeta de crédito, etc.); la adición de campos de formulario HTML a una solicitud entrante POST (número de tarjeta Visa, etc.); y gracias a la sustitución de cadenas (UID -> Identificador de usuario) los compiladores convierten los formatos de objetos web y los tipos MIME "sobre la marcha" en respuestas HTTP y envían solicitudes HTTP por sí mismos, obteniendo un objeto de respuesta a cambio.

Como se muestra en la Figura 17, en una modalidad preferida el proxy H2O se implementa al desarrollar una extensión de software envolvente (proxy web, firewall, servidor web u otro). Este software de host proporciona subprocesamiento y programación H2O de las tareas H2O, así como algunas de las funcionalidades necesarias para implementar "compiladores" H2O y componentes de revisión.

Mediante el uso de la API proporcionada por el software de host proxy se suministra un conjunto de API (las API proxy H2O) para implementar las funcionalidades requeridas por los compiladores H2O que no se encuentran en los Servicios de Software de Host Proxy H2O y proporcionar un mayor nivel de abstracción para los servicios disponibles en el Software de Host Proxy H2O. El componente de revisor de solicitud (1424) lee las solicitudes HTTP entrantes para las páginas HTML y las completa con información de otro proceso o archivo o base de datos. El compilador HTML2RES (1420) compila las páginas devueltas HTML a archivos de recursos PS y cambia el tipo MIME del encabezado de respuesta HTTP para adaptarse a un nuevo formato: tipo MIME: text/otvres.

El compilador GIF2PIX (1422) convierte una imagen GIF devuelta a un archivo de recurso PS y cambia el tipo MIME del encabezado de respuesta HTTP para adaptarse a un nuevo formato: Tipo MIME: image/otvpix. El compilador 2MPEG (1426) convierte una imagen GIF o JPEG devuelta a un archivo de recurso PS y cambia el tipo MIME del encabezado de respuesta HTTP para adaptarse a un nuevo formato: Tipo MIME: image/otvmpg. Si nos fijamos a continuación en la Figura 18, se ilustra en la misma una solicitud dinámica de un diagrama de secuencia de una página HTML. En el diagrama de secuencia no se muestran las cachés de objetos ya que se trata de componentes "pasivos" en esta interacción. El STB de usuario (212) envía una solicitud (1810) para obtener una página (home.asp) a través de la solicitud HTTP. El revisor de solicitud (1424) accede a un proceso/archivo/base de datos/dirección URL externos (1812 y 1814) para obtener el nombre de usuario, incorpora revisiones a la solicitud y la envía (1816) al compilador HTML2RES. El compilador HTML2RES envía la solicitud (1818) al sitio web de destino (amazon.com). El sitio web procesa la solicitud y devuelve (1820) la página HTML resultante al compilador HTML2RES. El compilador HTML2RES analiza el archivo para obtener la dirección URL de los vínculos de imagen y enviar las solicitudes (1822) al sitio web para obtener (1824) los archivos de imágenes (logo.gif, banner.jpg). El compilador HTML2RES procesa el diseño en la televisión para la página, lo compila en un archivo de recursos PS y lo devuelve (1830) al STB. El STB procesa y reproduce la página HTML en la televisión.

Si nos fijamos a continuación en la Figura 19, en la misma se ilustra un diagrama de secuencia de una solicitud de un archivo de imagen. Una página HTML que está cargándose en el STB del usuario necesita una imagen para su reproducción en pantalla. La página envía una solicitud HTTP (1910) de la imagen (opciones de conversión incrustadas en la dirección URL) al compilador 2MPG. El compilador 2MPG solicita la imagen (1912) al sitio de destino (amazon.com). El sitio de destino devuelve el archivo de imagen banner.jpg (1914) al compilador 2MPG. El compilador 2MPG convierte el archivo bannerJpg, utilizando para ello las opciones especificadas en la dirección URL, y devuelve el resultado (1916), con un tipo MIME image/otvmpg, al STB. El STB reproduce la imagen en pantalla.

Los diferentes compiladores H2O identificados heredan de la clase H2OCompiler y proporcionan una implementación para los diferentes puntos de entrada virtuales puros de la clase. Se suministran funciones de memoria a los compiladores para asignar y liberar los búferes de solicitudes/respuestas. Se suministra el tamaño del búfer asignado a una función FreeBuffer, de forma que puedan utilizarse diferentes esquemas para liberar el búfer (para un tamaño determinado se podría implementar el búfer como un archivo temporal asignado a la memoria, mientras que para tamaños más pequeños se prefiere implementar como un búfer de memoria).

El búfer se pasa a una función Ejecutar que contiene la solicitud/respuesta HTTP completa, el compilador analiza los encabezados de solicitud, los tipos MIME y adopta las acciones apropiadas. Es preferible que este búfer sea de sólo lectura. Sin embargo, también es posible que se pueda escribir en el búfer para permitir el aumento por el compilador u otras funciones posteriores. El búfer devuelto por las funciones Ejecutar contiene una solicitud/respuesta HTTP válida, la memoria será liberada por el proxy H2O utilizando la función apropiada FreeBuffer y la función AllocBuffer suministrada ha de asignada. Se proporciona un miembro de depuración para los implementadores de compilador que permite realizar una depuración y seguimiento desde el interior del entorno proxy H2O.

Se utilizan las funciones de parámetro para obtener los nombres de los parámetros, obtener los valores actuales (cadena) de los parámetros, establecer un valor nuevo para un parámetro y validar un conjunto de parámetros. Se proporcionan funciones URL para que el compilador de HTML pueda buscar y obtener imágenes. Estas funciones están disponibles para otros compiladores y proporcionan servicios adicionales a los componentes cuando así se requiere.

Por ejemplo, una red con un millón de STB, con una media de 20.000 usuarios conectados, genera 2.000 solicitudes por segundo de páginas HTML a la SGW y a la "caché de objetos compilados" (a menos que parte de las páginas solicitadas procedan de banda ancha). Suponiendo que esas páginas fueran estáticas y debieran ser atendidas

inmediatamente desde la "caché de objetos compilados", el proxy H2O tendrá que atender 200 solicitudes por segundo. Suponiendo que una página HTML típica incrusta 10 imágenes, y 8 de cada 10 son JPEG, "70 Proxy envía 10 solicitudes por cada solicitud entrante. La "caché de objetos no compilados" atiende 2.000 solicitudes por segundo.

Es preferible que la conversión MPG se lleve a cabo por adelantado, siempre que sea posible. Un rastreador web puede solicitar las páginas e imágenes HTML por la noche para convertirlas por adelantado, contribuyendo a la resolución de este problema. Los compiladores, por consiguiente, interactúan con H2O. H2O (248) es la solución preferida de cliente/servidor suministrada en la PS que permite a los programadores de contenido de Internet crear contenido, aplicaciones y servicios de televisión interactiva para operadores de red que trabajan en la PS. Por lo tanto, H2O permite hacer llegar el amplio conjunto de talentos y contenido de Internet al vasto mercado mundial en expansión de las aplicaciones de televisión interactiva. El servidor H2O convierte el contenido en Internet (páginas HTML, secuencias de comandos ECMA y formateado de páginas HTML) en activos de la PS. El cliente de H2O, H2OC, procesa los activos e interactúa con los clientes. En los campos del comercio televisivo y el comercio electrónico, H2O permite a las tiendas de comercio electrónico y televisivo utilizar las herramientas web existentes para crear servicios de compras y comunicarse con la PS preferida (operador), sirviéndose para ello de protocolos Web estándar. De esta forma la presente invención resulta fácil de usar, proporcionando API a través de metodologías conocidas.

H2O actúa como un proxy para la SGW y las herramientas de difusión con el fin de convertir el contenido Web en contenido PS. De esta forma los programadores de sitios web pueden utilizar sus servidores HTTP y servidores de aplicaciones actuales para generar contenidos de televisión interactiva de forma económica. En una manifestación preferida, H2O convierte HTML, JavaScript y gráficos de Internet. No obstante, también se puede agregar cualquier otro contenido o protocolo conocidos o desarrollados en Internet a la funcionalidad de proxy de H2O. H2O permite a la PS mostrar páginas web en STB que no son compatibles totalmente con navegadores y crear interfaces de usuario originales. H2O permite la conexión PS a cualquier motor de comercio que utiliza únicamente HTML. H2O es responsable de la conversión de todo el contenido actual y futuro de banda ancha y Web (como por ejemplo páginas HTML, imágenes JPG, archivos de audio WAV, etc.) a recursos PS.

La parte de servidor de H2O, H2OS, es un proxy HTTP. Para otros fines se puede empaquetar como una biblioteca de vínculos dinámicos (DLL) o una herramienta de lotes. La parte de cliente de H2O, H2OC, es una aplicación en código objeto del STB. H2OC está ubicado por encima de otros componentes de cliente PS, como por ejemplo la biblioteca SGW o la biblioteca de carga de carrusel. H2O permite la utilización de direcciones URL para dirigirse a documentos y servicios. H2O también permite realizar un seguimiento en los entornos de difusión en línea. H2OS proporciona una funcionalidad de proxy HTTP. Las aplicaciones PS solicitan un documento a través de H2O, H2O obtiene el documento, lo analiza, lo compila y lo devuelve al solicitante. Esta funcionalidad de H2O permite el uso del mismo motor para usos diferentes, en línea y por difusión, facilita la escalabilidad y permite un uso flexible de H2O. El análisis depende del tipo de documento y puede ser análisis de HTML, de una imagen GIF, de imágenes JPEG, etc. Para permitir su expansión, H2O es capaz de aceptar complementos y funcionar con nuevos filtros de terceros.

H2O permite la utilización de secuencias de comandos en diferentes lenguajes. Preferentemente, todos los componentes de servidores PS se normalizan alrededor de las tareas de seguimiento, en particular la capacidad de administrar remotamente los diferentes procesos. Se utiliza SNMP para administrar las funciones básicas ("proceso OK" y capturas de problemas importantes). Se proporciona un intérprete de línea de comandos en un socket para inspeccionar el estado. El establecimiento de parámetros permite una administración remota y proporciona una interfaz con la Web a través de secuencias de comandos Web. En una modalidad preferida, se proporcionan avisos normalizados y registros de error.

HTML/JS no permiten compartir información de una página a otra en la Web, ya que el servidor se encarga del contexto. En modo de difusión, este sistema resulta insuficiente. La presente invención proporciona un modo de difusión, preferentemente, en el que se proporciona un objeto permanente global que no es eliminado cuando se inicia una nueva página. El objeto permanente mantiene el contexto entre las páginas. Otros objetos base proporcionados por la PS también se hacen permanentes en la transición (por ejemplo, control de estación, OSD). Los dispositivos se definen a través de un lenguaje de definición de interfaz para permitir la creación de nuevos dispositivos y la modificación de dispositivos y para permitir la adición de métodos sin modificar el compilador.

La característica de carrusel H2O proporciona una actualización en tiempo real de los catálogos, manteniendo la coherencia de los catálogos durante las actualizaciones y proporcionando modelos seguros de transacciones. El carrusel H2O permite la actualización de una sola página o de un conjunto completo de páginas en una sola transacción. La administración de carrusel proporciona la administración de un índice o directorio de carrusel. El índice contiene información para acceder y buscar datos en el carrusel. Es decir, para una página determinada, el administrador de carrusel contiene una lista de todos los módulos necesarios, de forma que H2OC solicita todos los módulos necesarios al mismo tiempo para acelerar el proceso.

El carrusel proporciona compresión de datos, metadatos en páginas (por ejemplo, prioridad relativa de página, frecuencia de envío de página) y seguimiento de página (*stream* elemental). El cliente de carrusel es una biblioteca OCOD de STB que administra la carga de recursos. El cliente de carrusel administra la dinámica de los recursos, es decir, los nuevos recursos, los recursos borrados y los recursos modificados. La administración dinámica de recursos permite al cliente (H2OC) de esta biblioteca presentar un contenido dinámico. El cliente de carrusel administra la asignación de memoria, la prebúsqueda y almacenamiento en caché de recursos y la descompresión de módulos. El

cliente de carrusel administra subíndices/directorios y un "conjunto" de recursos, en lugar de un "árbol" de recursos, lo que facilita la tarea de compartir recursos. Los subconjuntos de un único árbol de recursos pueden asignarse a procesos independientes, permitiendo de esta manera compartir los recursos.

H2O realiza un seguimiento de los desencadenadores, el rendimiento y el ancho de banda de los televisores, por ejemplo los recursos compartidos en los módulos compartidos. H2O optimiza la utilización de ancho de banda. H2O proporciona multipistas, multiprioridades y la administración del volumen de oferta de datos. H2O proporciona prebúsqueda en tiempo de ejecución y almacenamiento en caché en el nivel de módulo. H2O admite módulos comprimidos. H2O admite navegación por teclas de flecha y teclas directas (por ejemplo de dígito o color), gestionando lenguaje internacional (chino), metadatos en páginas (por ejemplo, prioridad relativa de página, frecuencia de su envío) y seguimiento de páginas (*stream* elemental). Se comparte un GUI global, es decir, se suministra un enlace directo por teclas, de forma que cualquier página de información puede compartirse con cualquier otra página.

H2O administra páginas y subpáginas para gestionar aquellos casos en los que las páginas son demasiado grandes para ajustarse a una pantalla sin necesidad de desplazarse. H2O permite el uso de HTML para presentar contenido, en línea, punto a punto y difusión. H2O permite la composición de una página con una mezcla de componentes de difusión y en línea. Por ejemplo, una página puede proceder de un servidor en línea, mientras que su fondo es difundido. H2O permite la combinación de contenido en el STB. Por ejemplo, una aplicación bancaria puede enviar las últimas 20 transacciones con tarjeta de crédito de un telespectador desde su servidor mientras se envía por difusión la página HTML. Preferentemente, una función de Java Script (similar a HTML) solicita XML al servidor, utilizando el resultado y algunas funciones DOM para enmendar una tabla con el resultado.

Preferentemente, se proporciona seguridad para una autenticación segura del telespectador, la cual se realiza en la SGW en vez de en H2O. No obstante, H2O puede incluir, alternativamente, una funcionalidad de autenticación. H2O también envía datos cifrados (por ejemplo, el envío del número de una tarjeta de crédito) entre un STB y un servidor en línea. Para algunos servicios, es apropiado atravesar un proxy de seguridad cerca de la conversión de HTML a PS. La PS puede utilizar HTTPS desde el proxy al proveedor de servicios, y un SSL como la biblioteca de OCOD desde el STB al proxy. En otros casos (por ejemplo, un banco), se proporciona seguridad de extremo a extremo, en cuyo caso H2O no realiza normalmente la traducción. Ese escenario se reserva, preferentemente, para datos que el STB es capaz de procesar sin necesidad de traducirlos a través de H2O. El cifrado puede realizarse, alternativamente, en la SGW o en el STB.

Se ha descrito la presente invención en el campo de la televisión interactiva a través de una modalidad preferida. Sin embargo, la presente invención también puede manifestarse en un sistema informático distribuido que comprende un servidor y un dispositivo cliente. En otra modalidad, la presente invención se implementa como un conjunto de instrucciones en un medio legible por ordenador que comprende ROM, RAM. CD-ROM, Flash o cualquier otro medio legible por ordenador, conocido o desconocido en la actualidad. Cuando un sistema informático distribuido lo ejecuta permite a dicho sistema informático distribuido implementar el método de la presente invención.

Aunque la invención mostrada anteriormente se ha ilustrado por medio de una modalidad preferida, ello se ha hecho únicamente a efectos de ejemplo y no se pretende limitar el ámbito de la invención, el cual se define en las siguientes reivindicaciones.

REIVINDICACIONES

1. Un método para proporcionar una comunicación en un sistema de televisión interactiva que comprende:

la recepción en una plataforma de servicios (50) de un primer mensaje dirigido a un servidor de aplicaciones (200). Este primer mensaje se recibe desde un dispositivo cliente (212) e incluye un identificador de usuario;

El método se **caracteriza** por las siguientes acciones:

la asignación del identificador de usuario recibido a un identificador de sesión; y

la transmisión al servidor de aplicaciones de un segundo mensaje que se corresponde con el primer mensaje;

en el que el segundo mensaje contiene el identificador de sesión y no contiene el identificador de usuario.

2. El método mencionado en la reivindicación 1, que además comprende la resolución de un identificador de sesión a un identificador de usuario, en el que únicamente la plataforma de servicios resuelve los identificadores de sesión a sus correspondientes identificadores de usuario.

3. El método mencionado en la reivindicación 2, en el que la plataforma de servicios además comprende una base de datos de perfiles de abonados configurada para almacenar la información de usuario, y en el que el método también comprende: la recepción en la plataforma de servicios de un mensaje que indica una solicitud de acceso a la base de datos; y la concesión o denegación del acceso solicitado en función de las reglas establecidas dentro de la plataforma de servicios; en el que las mencionadas reglas especifican los derechos especiales de acceso para la aplicación que dio origen al mensaje.

4. El método mencionado en la reivindicación 3, en el que las mencionadas reglas se corresponden con acuerdos comerciales suscritos entre un operador de la plataforma de servicios y un operador del servidor de aplicaciones.

5. El método mencionado en la reivindicación 3, que además comprende la conversión de un protocolo de transporte de un dispositivo cliente a un formato que sea compatible con un protocolo de transporte de un servidor de aplicaciones.

6. El método mencionado en la reivindicación 5, que además comprende:

la conversión de contenido recibido del servidor de aplicaciones a un formato compatible con el dispositivo cliente;

y la conversión de contenido recibido del dispositivo cliente a un formato compatible con el servidor de aplicaciones.

7. Un equipo para facilitar la comunicación en un sistema de televisión interactivo. Dicho equipo comprende:

medios configurados para comunicarse con un dispositivo cliente (212) por medio de un primer enlace de comunicación; medios configurados para comunicarse con un servidor de aplicaciones (200) por medio de un segundo enlace de comunicación; y una plataforma de servicios (50) configurada para recibir un primer mensaje, incluido un identificador de usuario del dispositivo cliente. La plataforma de servicios se **caracteriza** por estar configurada para: asignar el identificador de usuario recibido a un identificador de sesión y enviar un segundo mensaje al servidor de aplicaciones. Dicho segundo mensaje se corresponde con el primer mensaje, contiene el identificador de sesión y no contiene el identificador de usuario.

8. El equipo mencionado en la reivindicación 7, en el que únicamente se configura la plataforma de servicios para resolver un identificador de sesión a un identificador de usuario.

9. El equipo mencionado en la reivindicación 8, en el que la plataforma de servicios comprende además una base de datos de perfiles de abonados configurada para almacenar información de usuario.

10. El equipo mencionado en la reivindicación 9, en el que la plataforma de servicios incluye además un mecanismo de control de transacciones (106 y 266) configurado para:

recibir un mensaje que indica una solicitud para acceder a la base de datos; y conceder o denegar el acceso solicitado basándose en las reglas establecidas dentro de la plataforma de servicios;

en el que las mencionadas reglas especifican derechos especiales de acceso para la aplicación en la que se originó el mensaje.

11. El equipo mencionado en la reivindicación 10, en el que las mencionadas reglas se corresponden con los acuerdos comerciales realizados entre un operador de la plataforma de servicios y un operador del servidor de aplicaciones.

12. El equipo mencionado en la reivindicación 10, en el que la mencionada plataforma de servicios comprende además un mecanismo de conversión de transporte (108 Y 246) configurado para convertir un protocolo de transporte del dispositivo cliente a un formato que resulte compatible con un protocolo de transporte del servidor de aplicaciones.

13. El equipo mencionado en la reivindicación 12, en el que la mencionada plataforma de servicios también comprende un mecanismo de control de contenido (204 y 248) configurado para:

convertir el contenido recibido del servidor de aplicaciones a un formato compatible con el dispositivo cliente;

convertir el contenido recibido del dispositivo cliente a un formato compatible con el servidor de aplicaciones.

14. El equipo mencionado en la reivindicación 7, en el que la plataforma de servicios está configurada para comunicarse con el dispositivo cliente a través de un canal de difusión y un canal punto a punto.

15. Un medio legible por ordenador que posee instrucciones que al ser ejecutadas por un sistema informático distribuido hacen que dicho sistema informático distribuido:

envíe un primer mensaje dirigido a un servidor de aplicaciones (200) desde un dispositivo cliente (212). Dicho primer mensaje incluye un identificador de usuario;

el medio se **caracteriza** por el hecho de que las instrucciones, cuando se ejecutan, hacen que el sistema informático distribuido:

reciba el primer mensaje en una plataforma de servicios (50). Dicha plataforma de servicios:

asigna el identificador de usuario recibido a un identificador de sesión; y envía un segundo mensaje que se corresponde con el primer mensaje:

en el que el segundo mensaje contiene el identificador de sesión y no contiene el identificador de usuario;

reciba el segundo mensaje en un servidor de aplicaciones (200).

16. El medio legible por ordenador, tal y como se menciona en la reivindicación 15, en el que las mencionadas instrucciones están configuradas además para que cuando se ejecuten hagan que la plataforma de servicios resuelva un identificador de sesión a un identificador de usuario, en el que únicamente la plataforma de servicios está configurada para resolver los identificadores de sesión a sus identificadores de usuario correspondientes.

17. El medio legible por ordenador, tal y como se menciona en la reivindicación 16, en el que la plataforma de servicios también comprende una base de datos de perfiles de abonados configurada para almacenar la información de usuario, y en el que las mencionadas instrucciones están configuradas para que cuando se ejecuten hagan que la plataforma de servicios:

reciba un mensaje que indica una solicitud de acceso a la base de datos; y conceda o deniegue el acceso solicitado, basándose para ello en las reglas establecidas dentro de la plataforma de servicios;

en el que las mencionadas reglas especifiquen derechos especiales de acceso para la aplicación en la que se originó el mensaje.

18. El medio legible por ordenador, tal y como se menciona en la reivindicación 17, en el que las mencionadas reglas se corresponden con los acuerdos comerciales realizados entre un operador de la plataforma de servicios y un operador del servidor de aplicaciones.

19. El medio legible por ordenador, tal y como se menciona en la reivindicación 17, en el que las mencionadas instrucciones están además configuradas para que cuando se ejecuten hagan que la plataforma de servicios convierta un protocolo de transporte de dispositivo de cliente a un formato que resulte compatible con un protocolo de transporte de servidor de aplicaciones.

20. El medio legible por ordenador, tal y como se menciona en la reivindicación 19, en el que las mencionadas instrucciones están además configuradas para que cuando se ejecuten hagan que la plataforma de servicios:

convierta el contenido recibido del servidor de aplicaciones a un formato compatible con el dispositivo cliente;

convierta el contenido recibido del dispositivo cliente a un formato compatible con el servidor de aplicaciones.

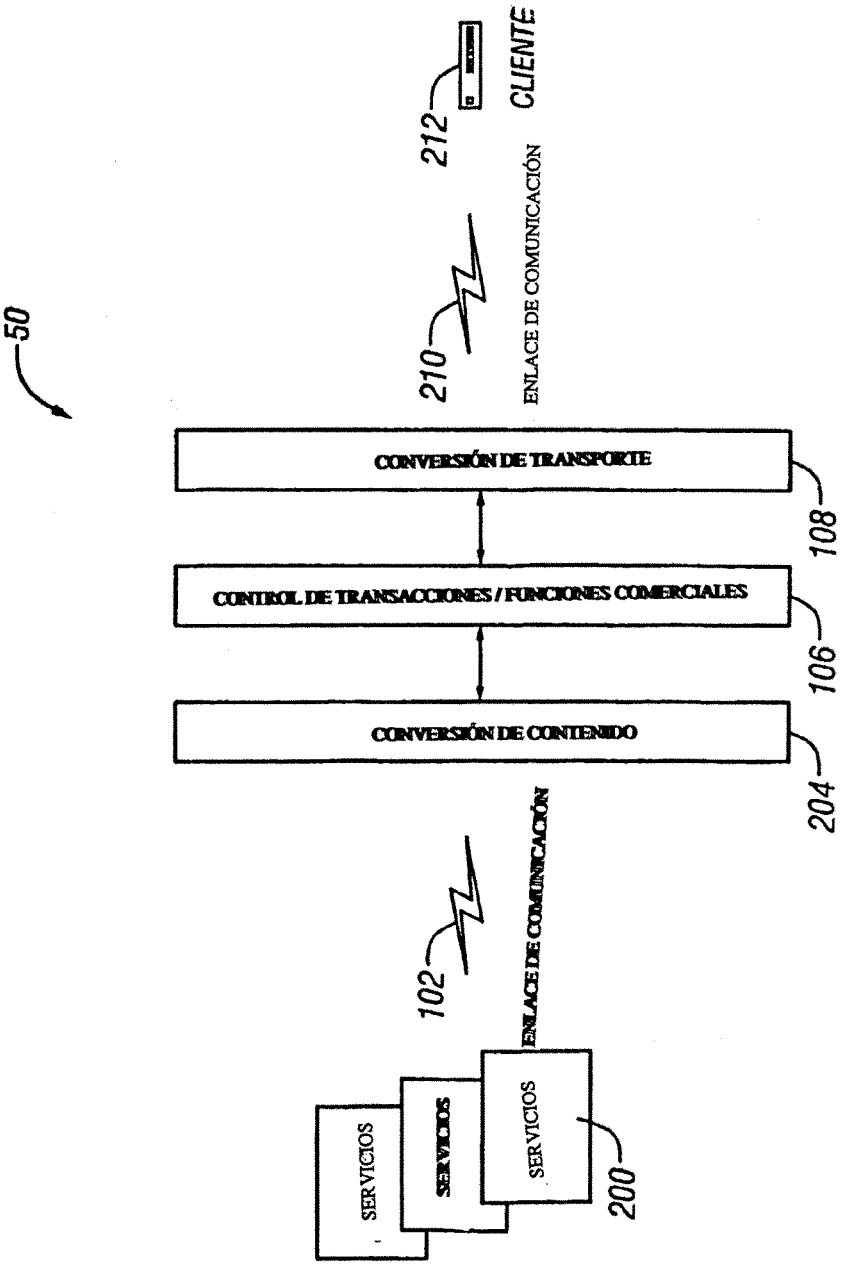
50
212
CLIENTE
210
ENLACE DE COMUNICACIÓN
CONVERSIÓN DE TRANSPORTE
108
CONTROL DE TRANSACCIONES / FUNCIONES COMERCIALES
106
CONVERSIÓN DE CONTENIDO
204
102
ENLACE DE COMUNICACIÓN
SERVICIOS
SERVICIOS
SERVICIOS
200

FIG. 1

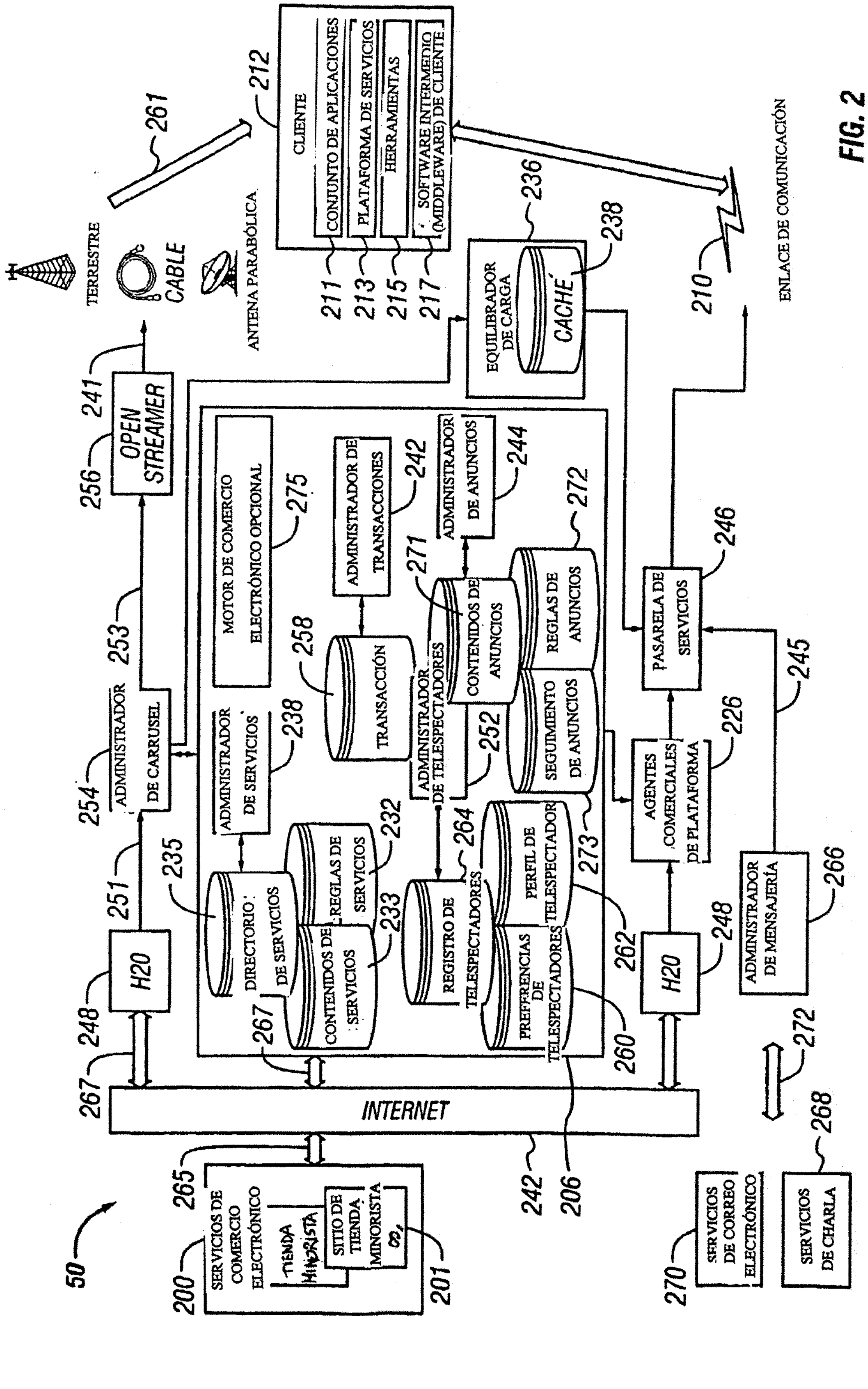
50
SERVICIOS DE COMERCIO ELECTRÓNICO
TIENDA MINORISTA
SITIO DE TIENDA MINORISTA
200
201
265
INTERNET
267
248
H20
251
254
ADMINISTRADOR DE CARRUSEL
253
256
OPEN STREAMER
241
TERRESTRE
CABLE
ANTENA PARABÓLICA
261
212
CLIENTE
CONJUNTO DE APLICACIONES
PLATAFORMA DE SERVICIOS
HERRAMIENTAS
SOFTWARE INTERMEDIO (MIDDLEWARE) DE CLIENTE
211
213
215
217
EQUILIBRADOR DE CARGA
CACHÉ
236
238
210
ENLACE DE COMUNICACIÓN
235
DIRECTORIO DE SERVICIOS
REGLAS DE SERVICIOS
232
CONTENIDOS DE SERVICIOS
233
ADMINISTRADOR DE SERVICIOS
238
MOTOR DE COMERCIO ELECTRÓNICO OPCIONAL
275
TRANSACCIÓN
258
ADMINISTRADOR DE TRANSACCIONES
242
ADMINISTRADOR DE ANUNCIOS
244
ADMINISTRADOR DE TELESPECTADORES
252
264
REGISTRO DE TELESPECTADORES
PERFIL DE TELESPECTADOR
PREFERENCIAS DE TELESPECTADORES
262
260
CONTENIDOS DE ANUNCIOS
271
REGLAS DE ANUNCIOS
272
SEGUIMIENTO DE ANUNCIOS
273
242
206
H20
248
AGENTES COMERCIALES DE PLATAFORMA
226
PASARELA DE SERVICIOS
246
245
ADMINISTRADOR DE MENSAJERÍA
266
272
SERVICIOS DE CORREO ELECTRÓNICO
270
SERVICIOS DE CHARLA
268

FIG. 2

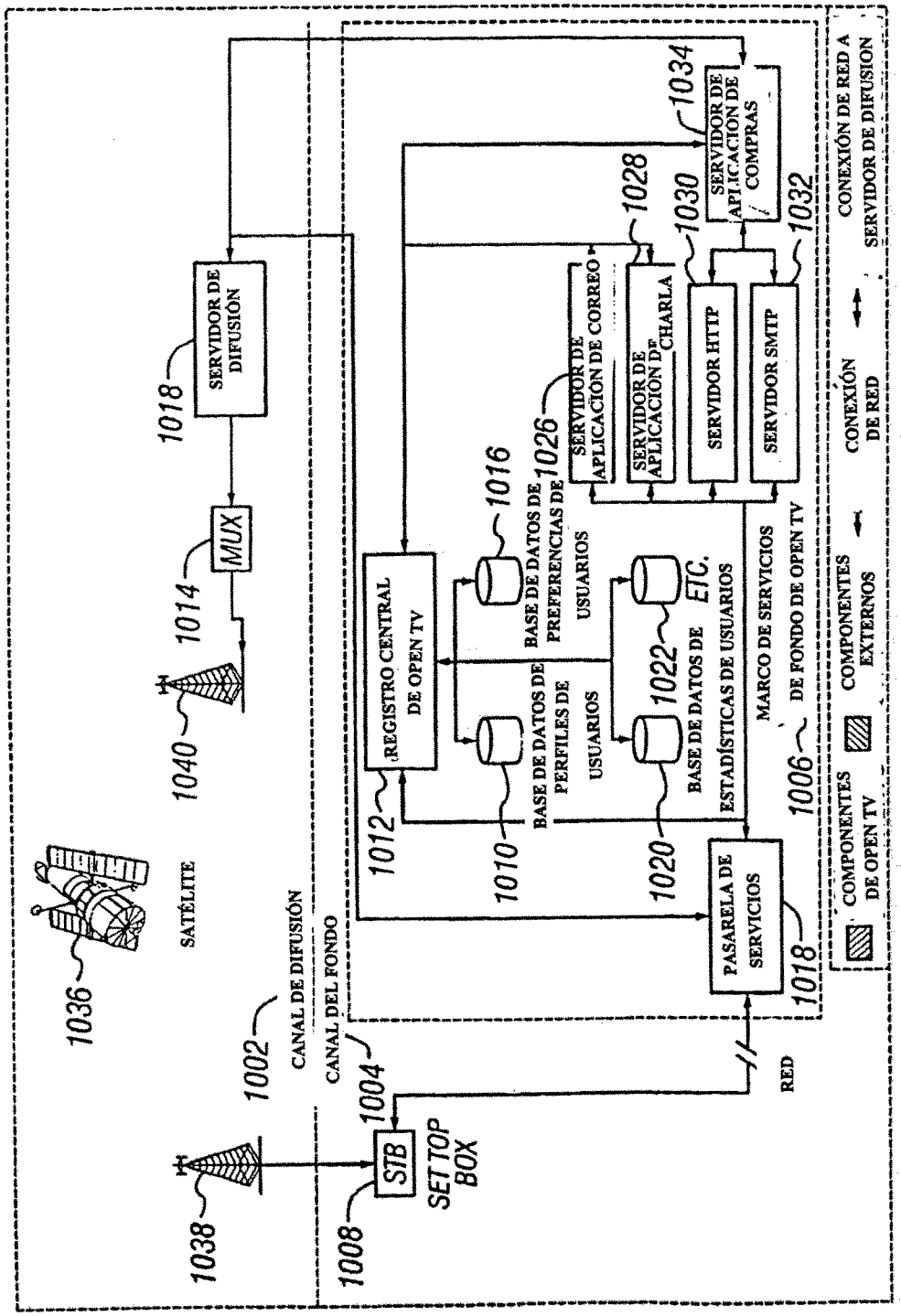
1036
SATÉLITE
1040
1014
MUX
1018
SERVIDOR DE DIFUSIÓN
1038
1008
STB
SET TOP BOX
1002
CANAL DE DIFUSIÓN
1004
CANAL DEL FONDO
1012
REGISTRO CENTRAL DE OPEN TV
1010
BASE DE DATOS DE PERFILES DE USUARIOS
1016
BASE DE DATOS DE PREFERENCIAS DE USUARIOS
1022
BASE DE DATOS DE ESTADÍSTICAS DE USUARIOS
ETC.
1020
PASARELA DE SERVICIOS
1018
RED
1006
MARCO DE SERVICIOS DE FONDO DE OPEN TV
1026
SERVIDOR DE APLICACIÓN DE CORREO
1028
SERVIDOR DE APLICACIÓN DE CHARLA
1030
SERVIDOR HTTP
1032
SERVIDOR SMTP
1034
SERVIDOR DE APLICACIÓN DE COMPRAS
COMPONENTES DE OPEN TV
COMPONENTES EXTERNOS
CONEXIÓN DE RED
CONEXIÓN DE RED A SERVIDOR DE DIFUSION

FIG. 3

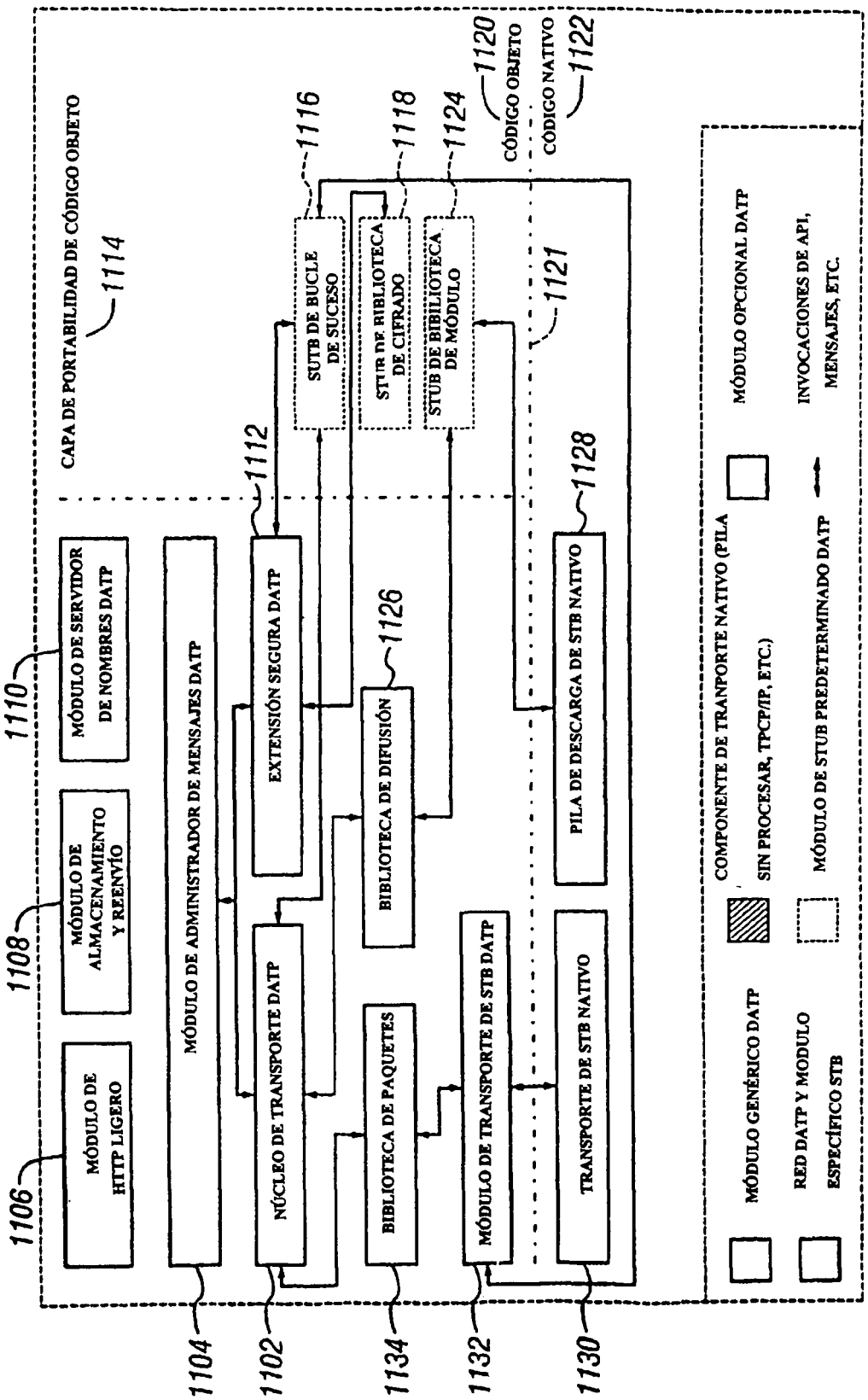
CAPA DE PORTABILIDAD DE CÓDIGO OBJETO
1114
MÓDULO DE SERVIDOR DE NOMBRES DATP
1110
MÓDULO DE ALMACENAMIENTO Y REENVÍO
1108
MÓDULO DE HTTP LIGERO
1106
MÓDULO DE ADMINISTRADOR DE MENSAJES DATP
1104
NÚCLEO DE TRANSPORTE DATP
1102
BIBLIOTECA DE PAQUETES
1134
MÓDULO DE TRANSPORTE DE STB DATP
1132
TRANSPORTE DE STB NATIVO
1130
EXTENSIÓN SEGURA DATP
1112
BIBLIOTECA DE DIFUSIÓN
1126
PILA DE DESCARGA DE STB NATIVO
1128
STUB DE BUCLE DE SUCESO
1116
STUB DE BIBLIOTECA DE CIFRADO
1118
STUB DE BIBLIOTECA DE MÓDULO
1124
1121
CÓDIGO OBJETO
1120
CÓDIGO NATIVO
1122
MÓDULO GENÉRICO DATP
RED DATP Y MODULO ESPECÍFICO STB
COMPONENTE DE TRANPORTE NATIVO (PILA SIN PROCESAR, TPCP/IP, ETC.)
MÓDULO DE STUB PREDETERMINADO DATP
MÓDULO OPCIONAL DATP
INVOCACIONES DE API, MENSAJES, ETC.

FIG. 4

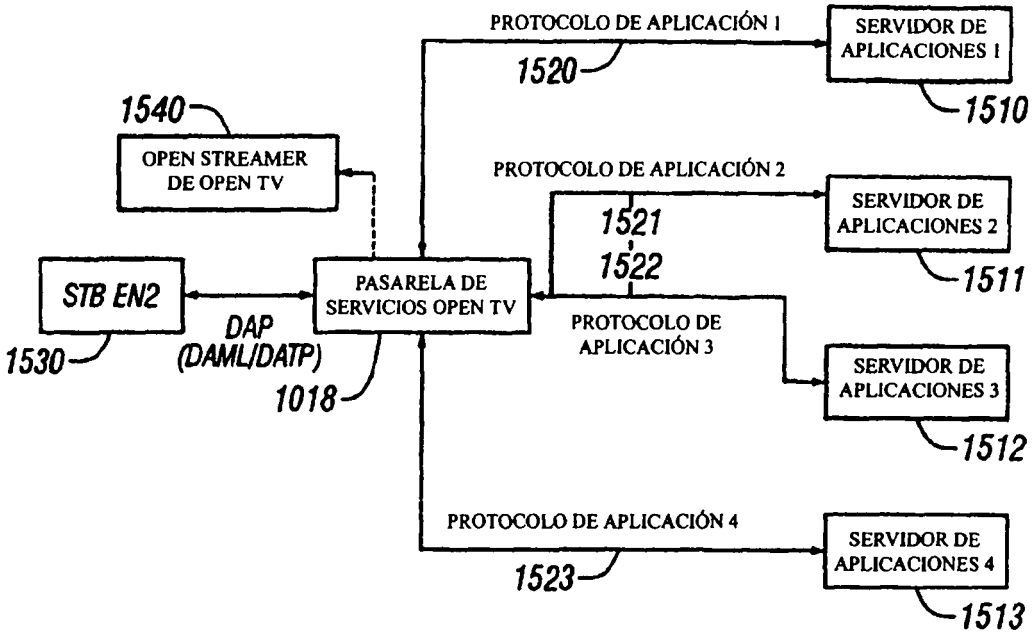
PROTOCOLO DE APLICACIÓN 1
1520
SERVIDOR DE APLICACIONES 1
1510
1540
OPEN STREAMER DE OPEN TV
PROTOCOLO DE APLICACIÓN 2
1521
1522
SERVIDOR DE APLICACIONES 2
1511
STB EN2
1530
DAP (DAML/DATP)
1018
PASARELA DE SERVICIOS OPEN TV
PROTOCOLO DE APLICACIÓN 3
SERVIDOR DE APLICACIONES 3
1512
PROTOCOLO DE APLICACIÓN 4
1523
SERVIDOR DE APLICACIONES 4
1513

FIG. 5

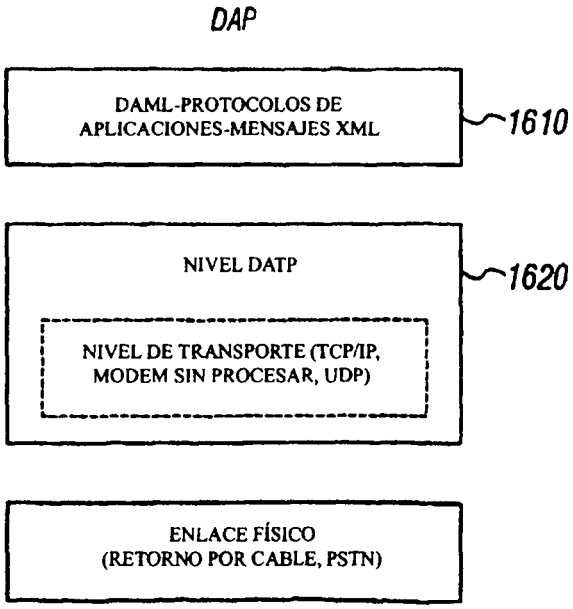
DAP
DAML-PROTOCOLOS DE APLICACIONES-MENSAJES XML
1610
NIVEL DATP
1620
NIVEL DE TRANSPORTE (TCP/IP, MODEM SIN PROCESAR, UDP)
ENLACE FÍSICO (RETORNO POR CABLE, PSTN)

FIG. 6

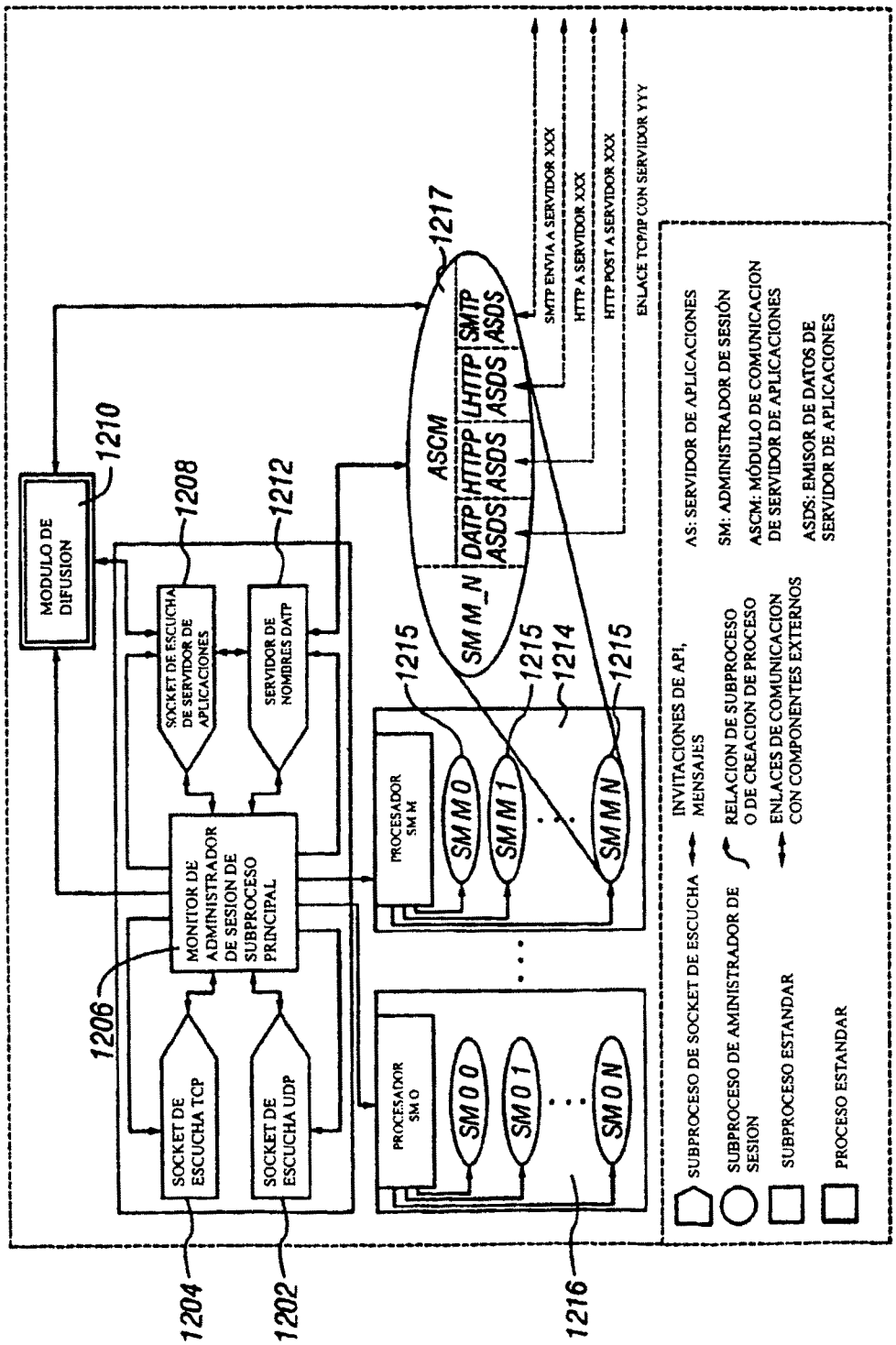
MODULO DE DIFUSION
1210
1208
1212
SOCKET DE ESCUCHA DE SERVIDOR DE APLICACIONES
SERVIDOR DE NOMBRES DATP
ASCM
DATP ASDS
HTTPP ASDS
LHTTP ASDS
SMTP ASDS
1217
SM M_N
SMTP ENVIA A SERVIDOR XXX
HTTP A SERVIDOR XXX
HTTP POST A SERVIDOR XXX
ENLACE TCP/IP CON SERVIDOR YYY
1206
MONITOR DE ADMINISTRADOR DE SESION DE SUBPROCESO PRINCIPAL
1215
1214
PROCESADOR SM M
SM M 0
SM M 1
SM M N
SOCKET DE ESCUCHA TCP
SOCKET DE ESCUCHA UDP
PROCESADOR SM 0
SM 0 0
SM 0 1
SM 0 N
1204
1202
1216
SUBPROCESO DE SOCKET DE ESCUCHA
INVITACIONES DE API, MENSAJES
SUBPROCESO DE AMINISTRADOR DE SESION
RELACION DE SUBPROCESO O DE CREACION DE PROCESO
SUBPROCESO ESTANDAR
ENLACES DE COMUNICACION CON COMPONENTES EXTERNOS
PROCESO ESTANDAR
AS: SERVIDOR DE APLICACIONES
SM: ADMINISTRADOR DE SESIÓN
ASCM: MÓDULO DE COMUNICACION DE SERVIDOR DE APLICACIONES
ASDS: EMISOR DE DATOS DE SERVIDOR DE APLICACIONES

FIG. 7

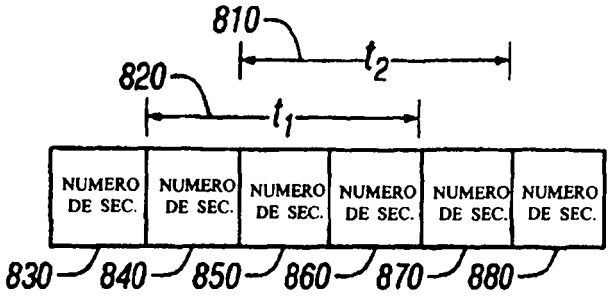
810
t_2
820
t_1
NUMERO DE SEC.
NUMERO DE SEC.
NUMERO DE SEC.
NUMERO DE SEC.
NUMERO DE SEC.
NUMERO DE SEC.
830
840
850
860
870
880
LISTA DE RECHAZO

FIG. 8

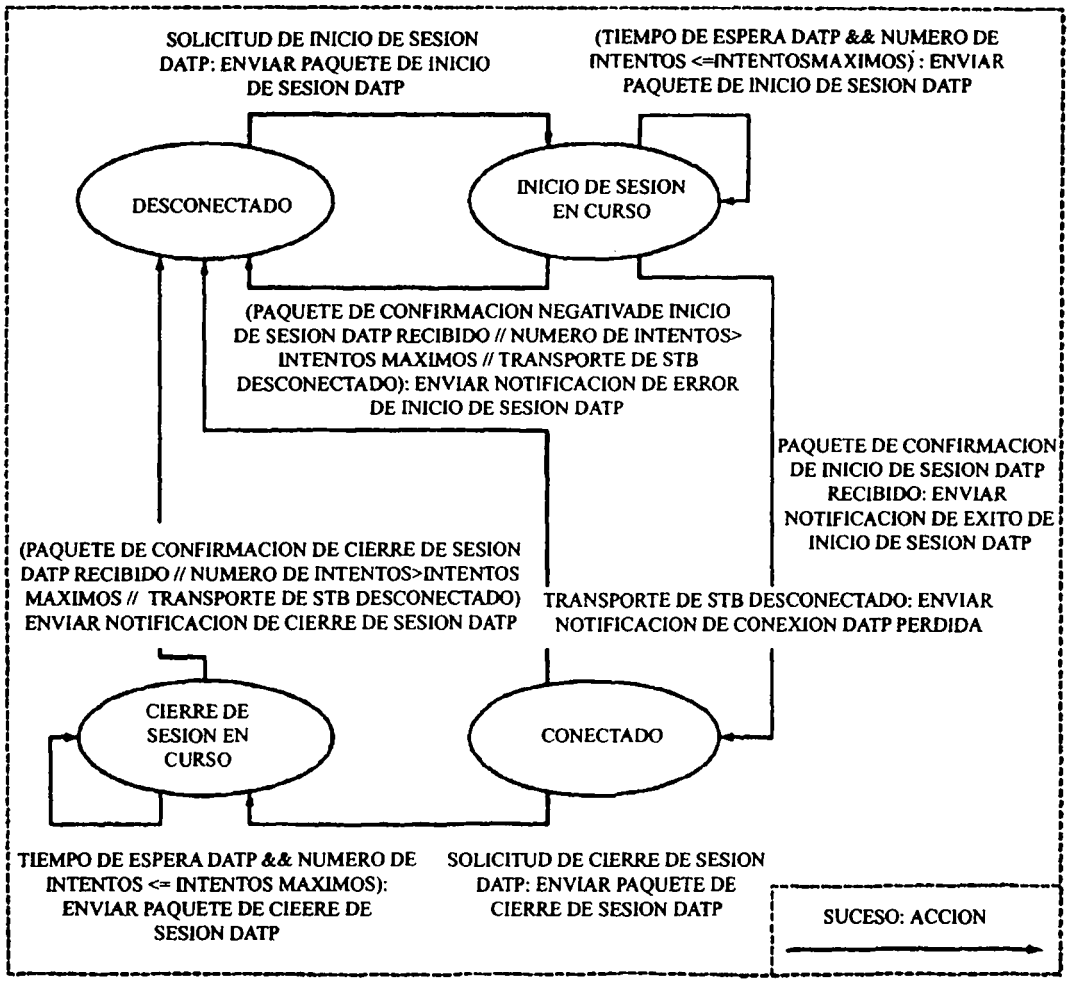
SOLICITUD DE INICIO DE SESION DATP: ENVIAR PAQUETE DE INICIO DE SESION DATP
(TIEMPO DE ESPERA DATP && NUMERO DE INTENTOS <=INTENTOSMAXIMOS) : ENVIAR PAQUETE DE INICIO DE SESION DATP
DESCONECTADO
INICIO DE SESION EN CURSO
(PAQUETE DE CONFIRMACION NEGATIVADE INICIO DE SESION DATP RECIBIDO // NUMERO DE INTENTOS> INTENTOS MAXIMOS // TRANSPORTE DE STB DESCONECTADO): ENVIAR NOTIFICACION DE ERROR DE INICIO DE SESION DATP
PAQUETE DE CONFIRMACION DE INICIO DE SESION DATP RECIBIDO: ENVIAR NOTIFICACION DE EXITO DE INICIO DE SESION DATP
(PAQUETE DE CONFIRMACION DE CIERRE DE SESION DATP RECIBIDO // NUMERO DE INTENTOS>INTENTOS MAXIMOS // TRANSPORTE DE STB DESCONECTADO) ENVIAR NOTIFICACION DE CIERRE DE SESION DATP
TRANSPORTE DE STB DESCONECTADO: ENVIAR NOTIFICACION DE CONEXION DATP PERDIDA
CIERRE DE SESION EN CURSO
CONECTADO
TIEMPO DE ESPERA DATP && NUMERO DE INTENTOS <= INTENTOS MAXIMOS): ENVIAR PAQUETE DE CIEERE DE SESION DATP
SOLICITUD DE CIERRE DE SESION DATP: ENVIAR PAQUETE DE CIERRE DE SESION DATP
SUCESO: ACCION

FIG. 10

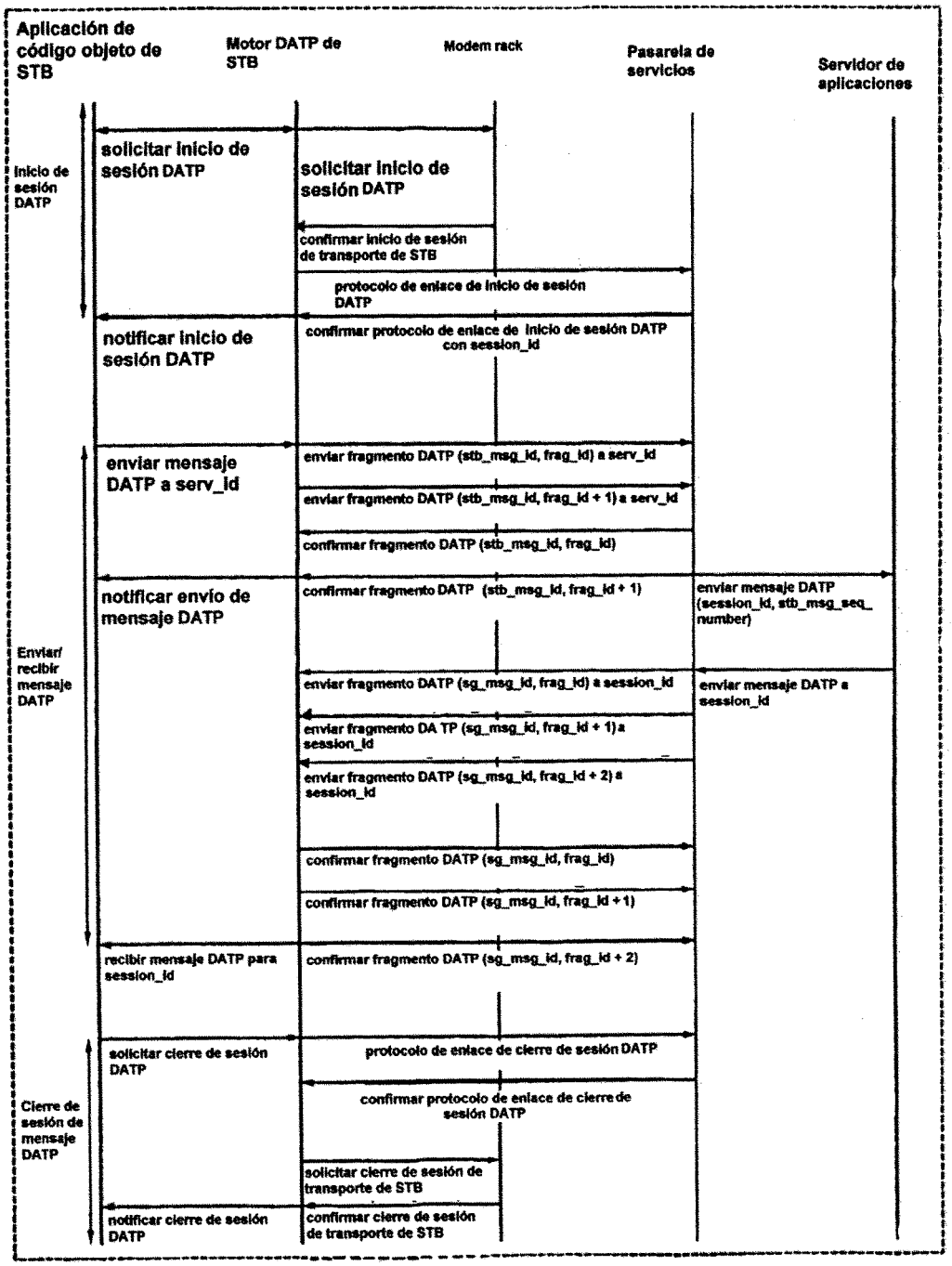
Aplicación de código objeto de STB
Motor DATP de STB
Modem rack
Pasarela de servicios
Servidor de aplicaciones
Inicio de sesión DATP
solicitar inicio de sesión DATP
solicitar inicio de sesión DATP
confirmar inicio de sesión de transporte de STB
protocolo de enlace de inicio de sesión DATP
notificar inicio de sesión DATP
confirmar protocolo de enlace de inicio de sesión DATP con session_id
enviar mensaje DATP a serv_id
enviar fragmento DATP (stb_msg_id, frag_id) a serv_id
enviar fragmento DATP (stb_msg_id, frag_id + 1) a serv_id
confirmar fragmento DATP (stb_msg_id, frag_id)
notificar envío de mensaje DATP
confirmar fragmento DATP (stb_msg_id, frag_id + 1)
enviar mensaje DATP (session_id, stb_msg_seq_number)
Enviar/ recibir mensaje DATP
enviar fragmento DATP (sg_msg_id, frag_id) a session_id
enviar mensaje DATP a session_id
enviar fragmento DA TP (sg_msg_id, frag_id + 1) a session_id
enviar fragmento DATP (sg_msg_id, frag_id + 2) a session_id
confirmar fragmento DATP (sg_msg_id, frag_id)
confirmar fragmento DATP (sg_msg_id, frag_id + 1)
recibir mensaje DATP para session_id
confirmar fragmento DATP (sg_msg_id, frag_id + 2)
solicitar cierre de sesión DATP
protocolo de enlace de cierre de sesión DATP
confirmar protocolo de enlace de cierre de sesión DATP
Cierre de sesión de mensaje DATP
solicitar cierre de sesión de transporte de STB
notificar cierre de sesión DATP
confirmar cierre de sesión de transporte de STB

FIG. 9

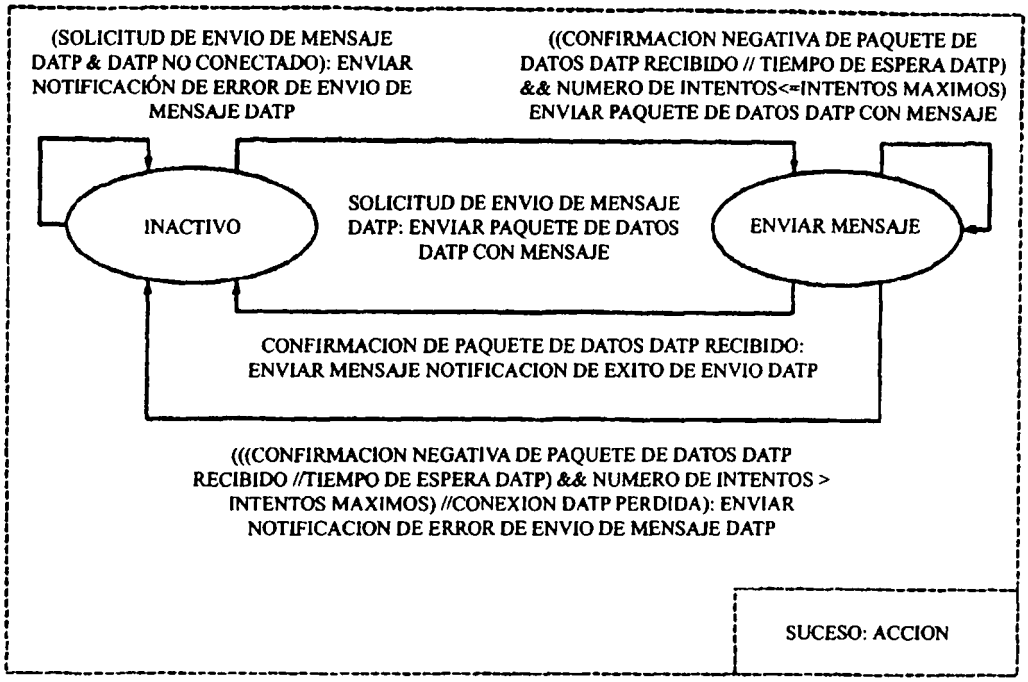
(SOLICITUD DE ENVIO DE MENSAJE DATP & DATP NO CONECTADO): ENVIAR NOTIFICACIÓN DE ERROR DE ENVIO DE MENSAJE DATP
((CONFIRMACION NEGATIVA DE PAQUETE DE DATOS DATP RECIBIDO // TIEMPO DE ESPERA DATP) && NUMERO DE INTENTOS<=INTENTOS MAXIMOS) ENVIAR PAQUETE DE DATOS DATP CON MENSAJE
INACTIVO
SOLICITUD DE ENVIO DE MENSAJE DATP: ENVIAR PAQUETE DE DATOS DATP CON MENSAJE
ENVIAR MENSAJE
CONFIRMACION DE PAQUETE DE DATOS DATP RECIBIDO: ENVIAR MENSAJE NOTIFICACION DE EXITO DE ENVIO DATP
(((CONFIRMACION NEGATIVA DE PAQUETE DE DATOS DATP RECIBIDO //TIEMPO DE ESPERA DATP) && NUMERO DE INTENTOS > INTENTOS MAXIMOS) //CONEXION DATP PERDIDA): ENVIAR NOTIFICACION DE ERROR DE ENVIO DE MENSAJE DATP
SUCESO: ACCION

FIG. 11

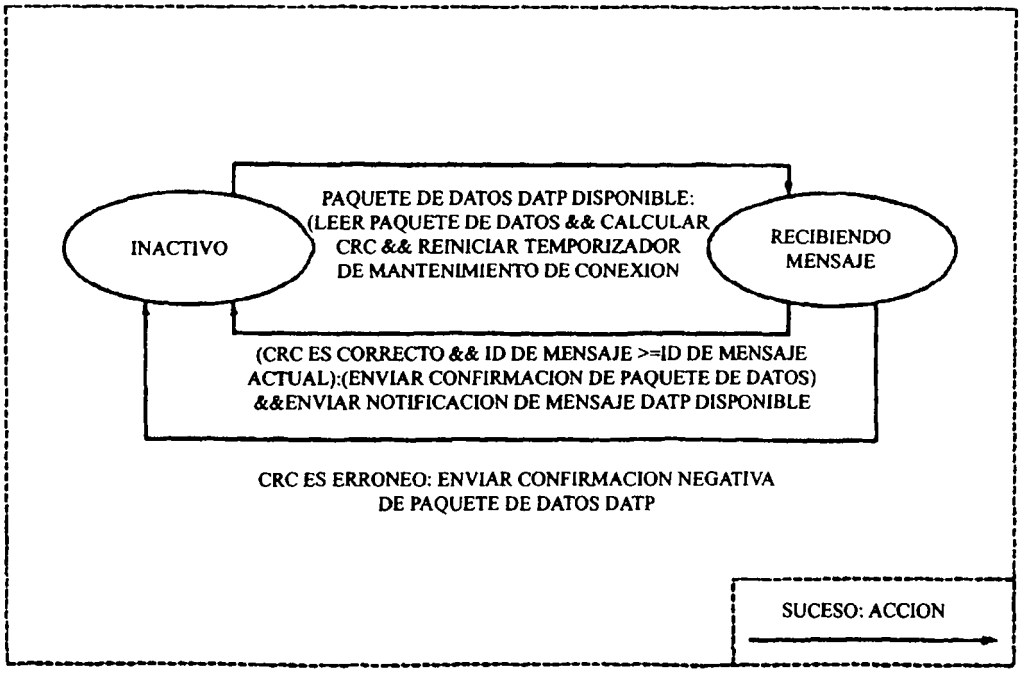
PAQUETE DE DATOS DATP DISPONIBLE: (LEER PAQUETE DE DATOS && CALCULAR CRC && REINICIAR TEMPORIZADOR DE MANTENIMIENTO DE CONEXION
INACTIVO
RECIBIENDO MENSAJE
(CRC ES CORRECTO && ID DE MENSAJE >=ID DE MENSAJE ACTUAL):(ENVIAR CONFIRMACION DE PAQUETE DE DATOS) &&ENVIAR NOTIFICACION DE MENSAJE DATP DISPONIBLE
CRC ES ERRONEO: ENVIAR CONFIRMACION NEGATIVA DE PAQUETE DE DATOS DATP
SUCESO: ACCION

FIG. 12

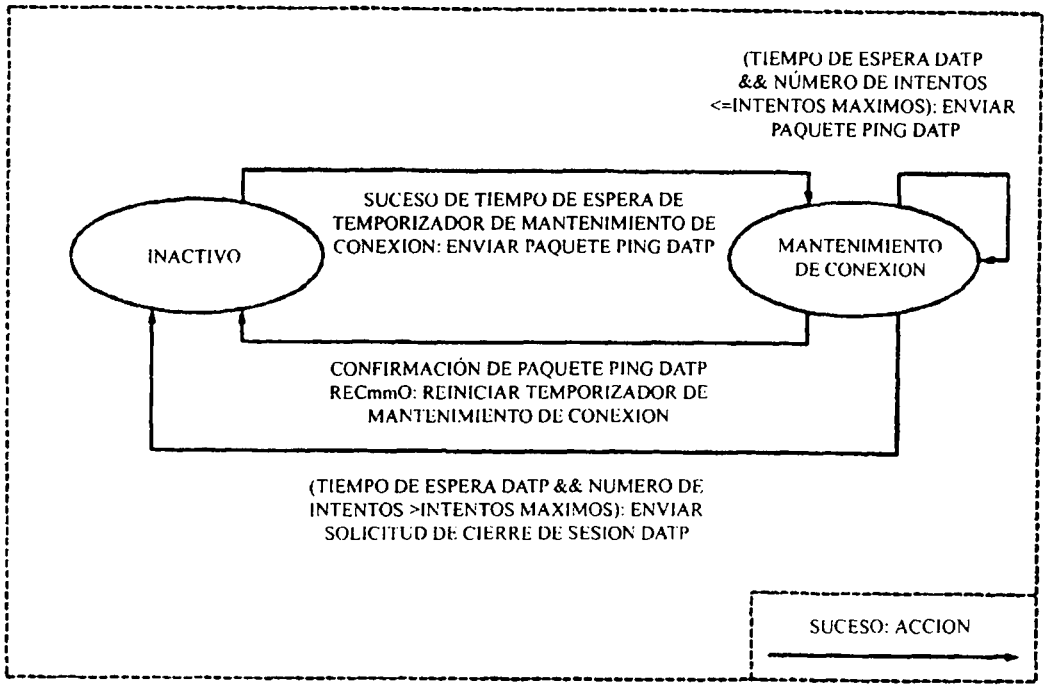
(TIEMPO DE ESPERA DATP
&& NÚMERO DE INTENTOS
<=INTENTOS MAXIMOS): ENVIAR
PAQUETE PING DATP
SUCESO DE TIEMPO DE ESPERA DE
TEMPORIZADOR DE MANTENIMIENTO DE
CONEXION: ENVIAR PAQUETE PING DATP
INACTIVO
MANTENIMIENTO
DE CONEXION
CONFIRMACIÓN DE PAQUETE PING DATP
RECmmO: REINICIAR TEMPORIZADOR DE
MANTENIMIENTO DE CONEXION
(TIEMPO DE ESPERA DATP && NUMERO DE
INTENTOS >INTENTOS MAXIMOS): ENVIAR
SOLICITUD DE CIERRE DE SESION DATP
SUCESO: ACCION

FIG. 13

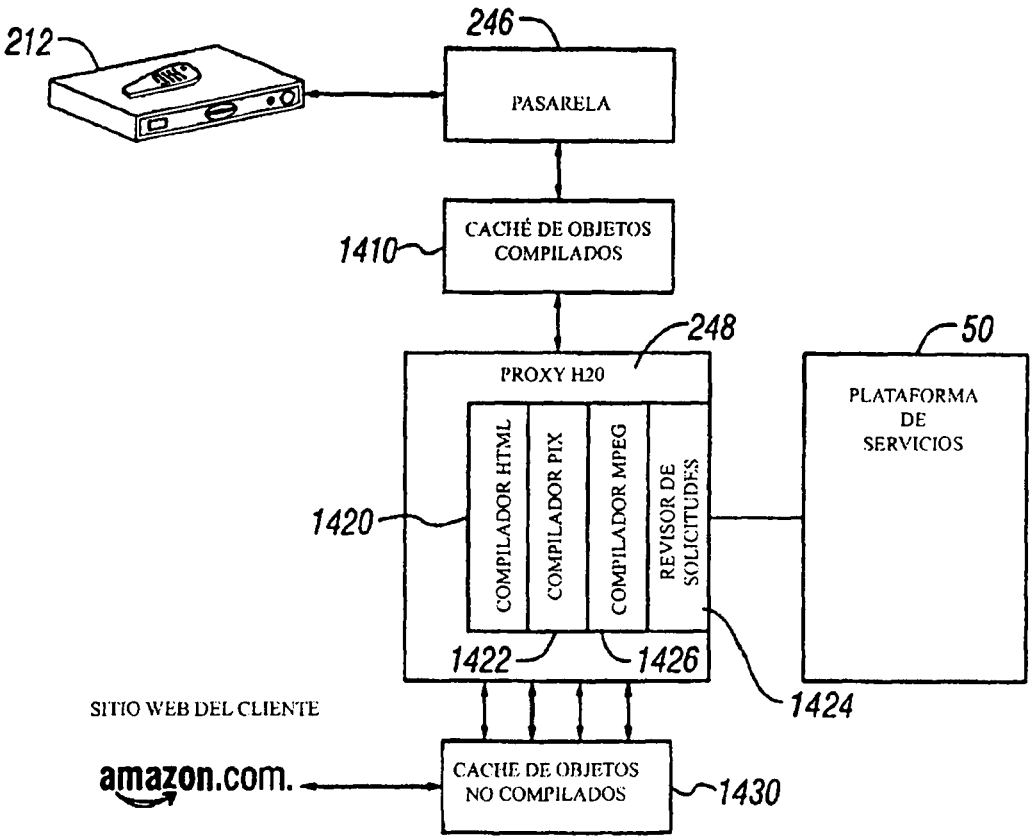
212
246
PASARELA
1410
CACHÉ DE OBJETOS COMPILADOS
248
PROXY H20
50
PLATAFORMA DE SERVICIOS
1420
COMPILADOR HTML
COMPILADOR PIX
COMPILADOR MPEG
REVISOR DE SOLICITUDES
1422
1426
1424
SITIO WEB DEL CLIENTE
amazon.com.
CACHE DE OBJETOS NO COMPILADOS
1430

FIG. 14

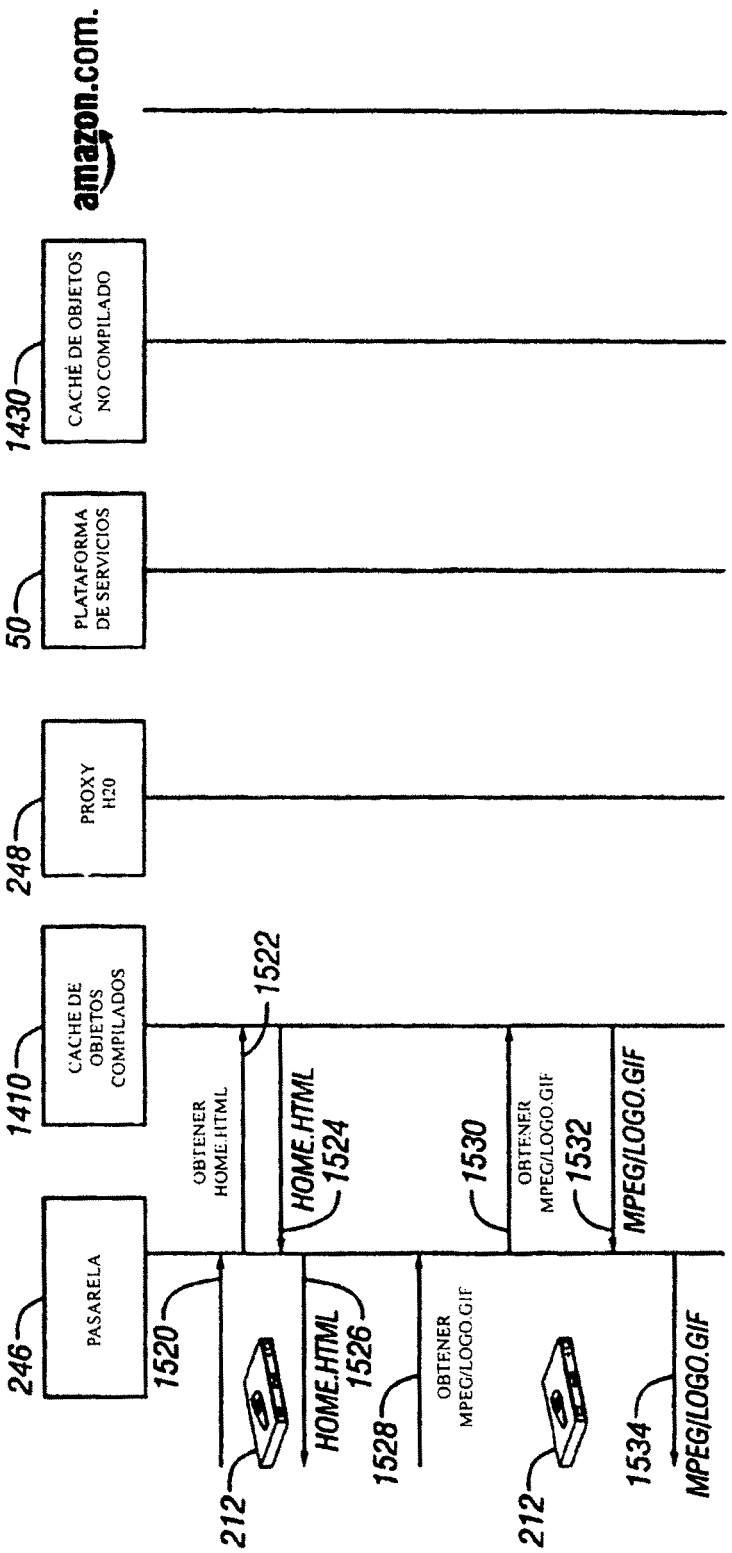
246
PASARELA
1410
CACHE DE OBJETOS COMPILADOS
248
PROXY H20
50
PLATAFORMA DE SERVICIOS
1430
CACHE DE OBJETOS NO COMPILADO
amazon.com.
1520
OBTENER HOME.HTML
1522
HOME.HTML 1524
1526
HOME.HTML
1528
OBTENER MPEG/LOGO.GIF
1530
OBTENER MPEG/LOGO.GIF 1532
MPEG/LOGO.GIF
1534
MPEG/LOGO.GIF
212
212

FIG. 15

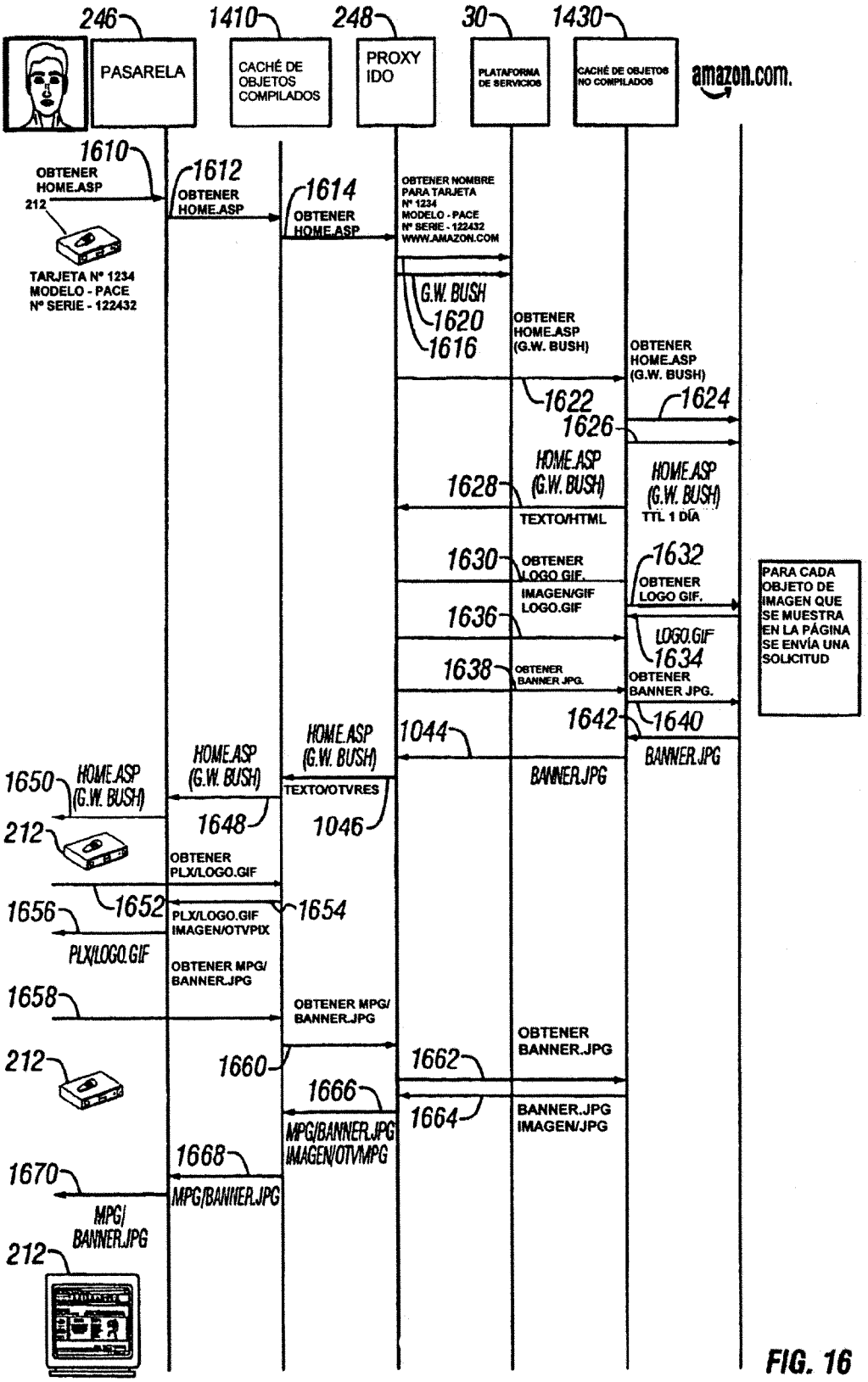
246
1410
248
30
1430
PASARELA
CACHÉ DE OBJETOS COMPILADOS
PROXY IDO
PLATAFORMA DE SERVICIOS
CACHÉ DE OBJETOS NO COMPILADOS
amazon.com.
1610
OBTENER HOME.ASP
212
TARJETA Nº 1234
MODELO - PACE
Nº SERIE - 122432
1612
OBTENER HOME.ASP
1614
OBTENER HOME.ASP
OBTENER NOMBRE PARA TARJETA Nº 1234 MODELO - PACE Nº SERIE - 122432 WWW.AMAZON.COM
G.W. BUSH
1620
1616
OBTENER HOME.ASP (G.W. BUSH)
OBTENER HOME.ASP (G.W. BUSH)
1622
1624
1626
HOME.ASP (G.W. BUSH)
HOME.ASP (G.W. BUSH)
TTL 1 DÍA
1628
TEXTO/HTML
1630
OBTENER LOGO GIF.
1632
OBTENER LOGO GIF.
IMAGEN/GIF LOGO.GIF
1636
LOGO.GIF
1634
1638
OBTENER BANNER JPG.
OBTENER BANNER JPG.
1642
1640
BANNER.JPG
PARA CADA OBJETO DE IMAGEN QUE SE MUESTRA EN LA PÁGINA SE ENVÍA UNA SOLICITUD
1044
BANNER.JPG
HOME.ASP (G.W. BUSH)
HOME.ASP (G.W. BUSH)
1650
HOME.ASP (G.W. BUSH)
TEXTO/OTVRES
1648
1046
212
OBTENER PLX/LOGO.GIF
1652
1654
1656
PLX/LOGO.GIF IMAGEN/OTVPIX
PLX/LOGO.GIF
OBTENER MPG/ BANNER.JPG
1658
OBTENER MPG/ BANNER.JPG
OBTENER BANNER.JPG
212
1660
1662
1666
1664
BANNER.JPG IMAGEN/JPG
MPG/BANNER.JPG IMAGEN/OTVMPG
1668
1670
MPG/BANNER.JPG
MPG/ BANNER.JPG
212

FIG. 16

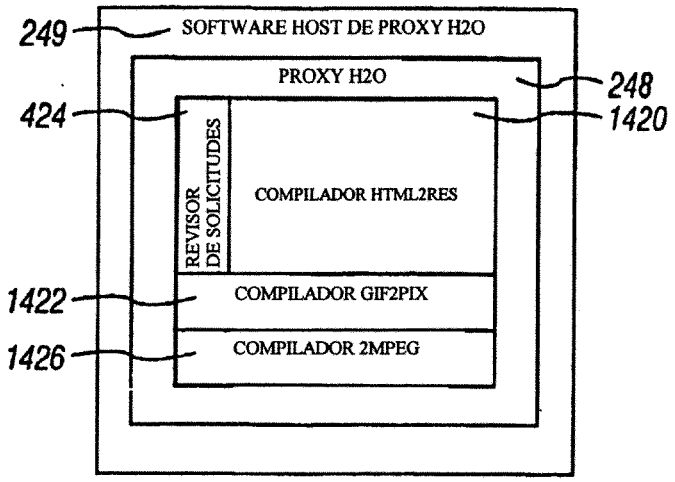
249
SOFTWARE HOST DE PROXY H2O
PROXY H2O
248
424
1420
REVISOR DE SOLICITUDES
COMPILADOR HTML2RES
1422
COMPILADOR GIF2PIX
1426
COMPILADOR 2MPEG

FIG. 17

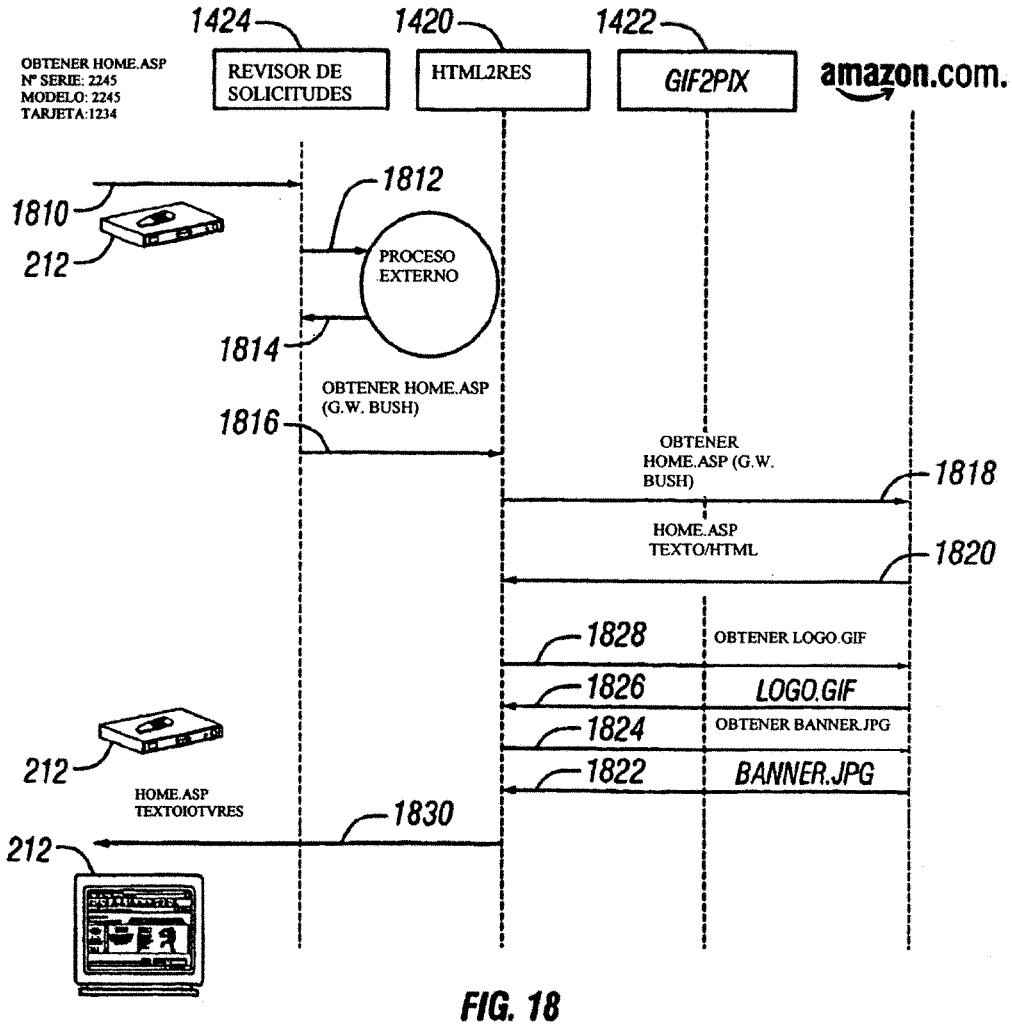
1424
1420
1422
OBTENER HOME.ASP
Nº SERIE: 2245
MODELO: 2245
TARJETA:1234
REVISOR DE SOLICITUDES
HTML2RES
GIF2PIX
amazon.com.
1812
1810
212
PROCESO EXTERNO
1814
OBTENER HOME.ASP (G.W. BUSH)
1816
OBTENER HOME.ASP (G.W. BUSH)
1818
HOME.ASP TEXTO/HTML
1820
1828
OBTENER LOGO.GIF
1826
LOGO.GIF
1824
OBTENER BANNER.JPG
1822
BANNER.JPG
212
HOME.ASP TEXTOIOTVRES
1830
212

FIG. 18

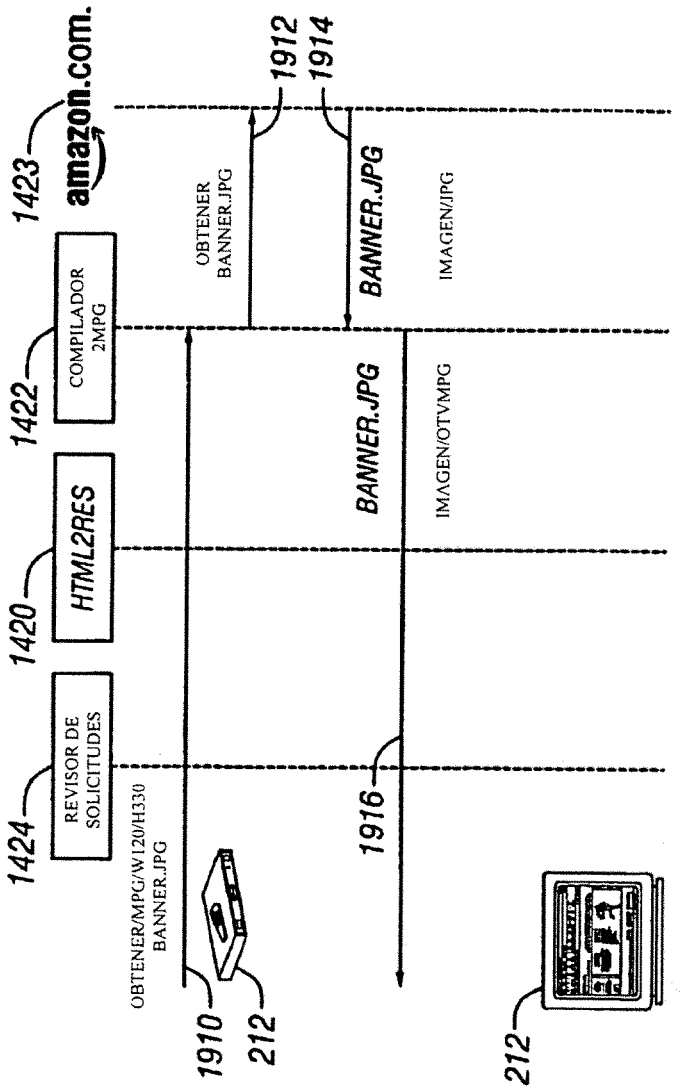
1423
amazon.com.
1912
1914
OBTENER BANNER.JPG
BANNER.JPG
IMAGEN/JPG
1422
COMPILADOR 2MPG
1420
HTML2RES
BANNER.JPG
IMAGEN/OTVMPG
1424
REVISOR DE SOLICITUDES
OBTENER/MPG/W120/H330 BANNER.JPG
1916
1910
212
212

FIG. 19