

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 388 428**

51 Int. Cl.:
H04L 12/56 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 96 Número de solicitud europea: **02768881 .1**
96 Fecha de presentación: **23.09.2002**
97 Número de publicación de la solicitud: **1430661**
97 Fecha de publicación de la solicitud: **23.06.2004**

54 Título: **Sistema y método para el procesamiento de paquetes**

30 Prioridad:
24.09.2001 US 962707

45 Fecha de publicación de la mención BOPI:
15.10.2012

45 Fecha de la publicación del folleto de la patente:
15.10.2012

73 Titular/es:
ERICSSON, INC.
6300 LEGACY, MS EVW 2-C-2
PLANO, TX 75024, US

72 Inventor/es:
BERGGREEN, Arthur y
HARDING-JONES, William, Paul

74 Agente/Representante:
de Elizaburu Márquez, Alberto

ES 2 388 428 T3

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

DESCRIPCIÓN

Sistema y método para el procesamiento de paquetes.

5 CAMPO DE LA INVENCION
La presente invención se refiere en general al campo de las comunicaciones, y más en particular, a un sistema y un método para el procesamiento de paquetes.

10 ANTECEDENTES DE LA INVENCION
La creciente demanda de comunicaciones de datos ha fomentado el desarrollo de técnicas que proporcionen medios más económicos y eficientes de utilización de redes de comunicación para manejar más información y nuevos tipos de información. Una de esas técnicas consiste en segmentar la información, que puede ser una comunicación de voz o de datos, en paquetes. Un paquete es típicamente un grupo de dígitos binarios, que incluye al menos información de datos y de control. Las redes de paquetes integrados (típicamente redes de paquetes rápidos) se utilizan en general para transportar dos (2) clases de tráfico, las cuales pueden incluir, por ejemplo, tasa de bits continua ("CBR"), conversación ("Voz en Paquetes"), datos ("Datos en Tramas"), imagen, y así sucesivamente. Las redes de paquetes originan, sumergen y/o envían paquetes de protocolo. Cada paquete tiene un formato bien definido y consiste en una o más cabeceras de paquete y algunos datos. La cabecera contiene típicamente información que proporciona control y/o información de la dirección, tal como la fuente y el destino del paquete.

20 La creación de la cabecera del paquete requiere típicamente una cantidad significativa de recursos de sistema, tal como una unidad central de procesamiento ("CPU") y/o un conmutador. Como resultado, la capacidad de tratamiento de un conmutador de comunicaciones está limitada o restringida por la capacidad de la CPU dentro del conmutador y de las demás funciones de procesamiento que la CPU debe proporcionar también. Tales restricciones de procesamiento provocan problemas de congestión y de Calidad de Servicio (QoS) en el interior del conmutador. Además, la capacidad de tratamiento del conmutador está determinada principalmente por la capacidad de la estructura de conmutación. Además, gran parte de la capacidad de procesamiento del conmutador está dedicada a procesar cabeceras de paquetes, las cuales típicamente no varían apreciablemente entre paquetes consecutivos. Como resultado, la capacidad de tratamiento del conmutador está limitada por el número de paquetes que éste puede procesar, a pesar del hecho de que el procesamiento es con frecuencia repetitivo. En consecuencia, existe una necesidad de un sistema y un método para el procesamiento de paquetes que incremente la capacidad de tratamiento del conmutador.

35 El libro blanco "Enrutamiento Basado en Políticas" de Cisco, describe ya un método de enrutamiento de paquetes de datos mediante el cual los clientes pueden implementar políticas que provoquen selectivamente que los paquetes tomen diferentes trayectorias. El enrutador hace pasar los paquetes a través de filtros denominados mapas de ruta que contienen cláusulas de emparejamiento que definen criterios sobre si los paquetes cumplen una política particular. En base a los criterios definidos en los mapas de ruta, los paquetes son reenviados a un siguiente salto. Los paquetes que no cumplen los criterios son enviados a través de canales de reenvío normales realizando un enrutamiento basado en el destino.

45 La solicitud de Patente US 5.732.079 se refiere a un sistema de procesamiento de datos con un bus de datos subdividido en una pluralidad de trayectorias de datos independientes y una pluralidad de componentes de sistema acoplados al bus de datos. Un circuito de control de asignación acoplado a los componentes del sistema asigna oportunidades de transmisión.

50 El documento "Diseño, Simulación y Evaluación de Conmutación TCP", de Bo Yang y Fen Wang, 4 de Diciembre de 2000, describe arquitecturas de conmutación en las que un clasificador de flujo determina si se deben asignar recursos a un flujo.

Sin embargo, ninguno de los documentos de la técnica anterior resuelve los problemas mencionados en lo que antecede.

55 SUMARIO DE LA INVENCION
La presente invención proporciona un sistema y un método para el procesamiento de datos en paquete, o paquetes, a través de un conmutador de comunicaciones que utiliza un sistema de Envío de Flujo Rápido ("FFF"). El FFF proporciona un envío expedito de paquetes basado en reglas de coincidencia de patrones y de manipulación de datos, que atraviesan los límites de la capa de protocolo. El FFF puede ser implementado en muchos entornos de protocolo para incrementar la eficacia del conmutador identificando paquetes asociados a un flujo particular. Un flujo es una corriente de paquetes correlacionados que se origina a partir de una fuente específica y que son suministrados a uno o más destinos específicos. Típicamente, esos flujos tendrán las mismas direcciones de origen y de destino, y otros criterios en común, que se originan a partir de una única sesión de cliente-servidor.

65 La presente invención proporciona un método para el procesamiento de un paquete en el que el paquete es recibido

y procesado utilizando una o más reglas de transformación en caso de que el paquete satisfaga una o más reglas de identificación. En otro caso, el paquete es procesado utilizando un proceso estándar en caso de que el paquete no satisfaga la una o más reglas de identificación. Este método puede ser implementado utilizando un producto de programa de ordenador que tenga un segmento de código para ejecutar cada etapa del método cuando se cargue en un conmutador de comunicaciones.

Adicionalmente, la presente invención proporciona un conmutador de comunicaciones que tiene una o más tarjetas de entrada, una o más tarjetas de control, una o más tarjetas de salida, y un bus de comunicaciones. El bus de comunicaciones acopla comunicativamente las tarjetas de entrada, las tarjetas de control y las tarjetas de salida entre sí. Cada tarjeta de control tiene al menos un procesador. Además, cada tarjeta de entrada recibe uno o más paquetes, procesa cada paquete utilizando una o más reglas de transformación en caso de que el paquete satisfaga una o más reglas de identificación, y envía cada paquete a uno de los procesadores para su procesamiento utilizando un proceso estándar en caso de que el paquete no satisfaga la una o más reglas de identificación.

La presente invención proporciona también un conmutador de comunicaciones que tiene una o más tarjetas de entrada, una o más tarjetas de control, una o más tarjetas de procesamiento de señal, una o más tarjetas de salida, una estructura de conmutación y un bus TDM. Cada tarjeta de control tiene al menos un procesador. Además, cada tarjeta de procesamiento de señal contiene una batería de procesadores de señales digitales. Cada procesador de señal digital crea uno o más paquetes y envía el uno o más paquetes a un motor de reenvío de flujo rápido. Cada motor de reenvío de flujo rápido recibe el uno o más paquetes, procesa cada paquete utilizando una o más reglas de transformación en caso de que el paquete satisfaga una o más reglas de identificación, y envía cada paquete a uno de los procesadores para su procesamiento utilizando un proceso estándar en caso de que el paquete no satisfaga las una o más reglas de identificación. La estructura de conmutación acopla comunicativamente las tarjetas de entrada, las tarjetas de procesamiento de señal, las tarjetas de control y las tarjetas de salida entre sí. El bus TDM acopla comunicativamente las tarjetas de entrada, las tarjetas de procesamiento de señal, las tarjetas de control, y las tarjetas de salida.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

Para una mejor comprensión de la invención, y para mostrar a título de ejemplo cómo puede ser llevada a cabo la misma, se hará ahora referencia a la descripción detallada de la invención junto con los dibujos que se acompañan, en los que los números correspondientes de las diferentes figuras se refieren a partes correspondientes, y en los que:

La Figura 1 es un diagrama de bloques de una realización de un conmutador de comunicaciones de acuerdo con la presente invención;

La Figura 2 es un diagrama de flujo de un controlador de envío de flujo rápido de acuerdo con la presente invención;

La Figura 3 es un diagrama de flujo de un motor de envío de flujo rápido de acuerdo con la presente invención;

La Figura 4 es un diagrama de un conmutador de red de paquetes de acuerdo con una realización de la presente invención, y

La Figura 5 es un diagrama esquemático de un conmutador de red de paquetes de acuerdo con una realización de la presente invención.

DESCRIPCIÓN DETALLADA DE LA INVENCION

Mientras que en lo que sigue se discute con detalle la realización y utilización de varias realizaciones de la presente invención, debe apreciarse que la presente invención proporciona muchos conceptos inventivos aplicables, que pueden ser materializados según una amplia diversidad de contextos específicos. Por ejemplo, además de en sistemas de telecomunicaciones, la presente invención puede ser aplicable a otras formas de comunicaciones o de procesamiento general de datos. Otras formas de comunicaciones pueden incluir las comunicaciones entre redes, comunicaciones vía satélite, o cualquier forma de comunicaciones no conocida aún por el hombre en la fecha de la presente invención. Las realizaciones específicas discutidas en la presente memoria son únicamente ilustrativas de formas específicas de realización y uso de la invención, y no limitan el alcance de la invención.

La presente invención proporciona un sistema y un método para el procesamiento de datos en paquetes, o paquetes de datos, mediante un conmutador de comunicaciones que utiliza un sistema de Envío de Flujo Rápido ("FFF"). El FFF proporciona el envío expedito de paquetes basado en reglas de emparejamiento de patrones y de manipulación de datos que cruzan los límites de la capa de protocolo. El FFF puede ser implementado en muchos entornos de protocolos, para incrementar la eficacia de la conmutación mediante identificación de paquetes asociados a un flujo particular. Un flujo es una corriente de paquetes correlacionados que se origina a partir de una fuente específica y que son suministrados a uno o más destinos específicos. Típicamente, estos flujos tendrán las mismas direcciones de origen y de destino, y otros criterios en común, que se originan a partir de una única sesión de servidor-cliente.

Por ejemplo, en el caso de un protocolo de voz por Internet ("VoIP"), una conversación de voz que podría comprender muchos paquetes de Protocolo de Internet ("IP") con diferentes VoIP de datos, es la capacidad de hacer

llamadas de teléfono y enviar faxes sobre redes de datos basadas en IP. Una red integrada de voz/datos permite una mayor estandarización y reduce las necesidades totales de equipamiento. El VoIP puede soportar aplicaciones multimedia y multi-servicio. Sin embargo, todos los paquetes asociados a una conversación específica tienen típicamente una información de cabecera igual o similar, y pueden así ser descritos como flujo. Una vez que la presente invención detecta un flujo, y las etapas de procesamiento estándar son registradas, el mismo tratamiento de procesamiento puede ser definido por unas pocas reglas generales que pueden ser aplicadas mediante dispositivos relativamente simples y no inteligentes (desconocimiento de protocolo). Por otra parte, en sistemas de procesamiento estándar (procesamiento de protocolo convencional), los paquetes de IP asociados a un flujo particular son conmutados individualmente en base a un proceso de software de protocolo por capas, mencionado como pila de protocolo. Sin embargo, este método reduce la eficacia de un conmutador debido a que cada paquete individual es procesado de forma similar por medio de los procesadores del conmutador, reduciendo con ello la capacidad de tratamiento del sistema y/o introduciendo una latencia de paquete inaceptable. Evitando el procesamiento de protocolo por capas de la trayectoria de procesamiento estándar para el resto del flujo, un conmutador que utilice el FFF de la presente invención permite una capacidad de tratamiento significativamente más alta para aquellos paquetes identificados como parte de un flujo.

IP especifica el formato de los paquetes, también llamados datagramas, y el esquema de direccionamiento. La mayor parte de las redes combinan IP con un protocolo de nivel más alto. Un protocolo de ese tipo es el llamado Protocolo de Control de Transporte ("TCP"), el cual establece una conexión virtual entre un destino y un origen. IP permite que un paquete sea direccionado y omitido en un sistema, pero no existe un enlace directo entre el remitente y el receptor. TCP/IP establece, por otra parte, una conexión entre dos anfitriones de modo que éstos pueden enviar mensajes de ida y vuelta durante un período de tiempo.

Otra cabecera de paquete de IP es el protocolo de transporte en tiempo real ("RTP"), el cual es un estándar de Internet para el transporte de datos en tiempo real, incluyendo audio y video. RTP se utiliza para identificar paquetes como contenedores de una muestra de voz en un formato de codificación particular. Se utiliza típicamente un número de secuencia y de fecha y hora para re-ensamblar una corriente de voz síncrona a partir de un caudal de paquetes de RTP. El RTP puede ser usado también para servicios multimedia bajo demanda y servicios interactivos tales como telefonía de IP. Por otra parte, la cabecera de protocolo de datagrama de usuario ("UDP") proporciona un transporte de datos eficiente pero no fiable (sin garantía). Éste se utiliza para el transporte de datos de voz en tiempo real puesto que la retransmisión de los datos en tiempo real podría añadir demasiado retardo a la conversación de voz. IP, sin embargo, proporciona una encapsulación estándar de datos para su transmisión por la red. Ésta contiene una dirección de origen y de destino utilizada para enrutamiento. MAC realiza funciones de gestión y maneja el protocolo de resolución de dirección ("ARP") para el dispositivo.

Haciendo ahora referencia a la Figura 1, se ha mostrado un diagrama de bloques de una realización de un conmutador 100 de comunicaciones de acuerdo con la presente invención. El conmutador 100 incluye una o más tarjetas 102 de control, una o más tarjetas 104 de entrada, y una o más tarjetas 106 de salida. Las tarjetas 102 de control, las tarjetas 104 de entrada y las tarjetas 106 de salida, están acopladas comunicativamente entre sí por medio de un bus 108 de comunicaciones, tal como un bus de interconexión de componentes periféricos ("PCI"). Las tarjetas 102 de control están también acopladas comunicativamente a una conexión 110 Ethernet a través de una interfaz 112 de Ethernet.

Las tarjetas 102 de control incluyen una o más CPUs o controladores 114 acoplados comunicativamente con la interfaz 116 de PCI, la cual permite el acceso al bus 108 de PCI. El controlador 114 está acoplado comunicativamente a un gestor 118 de protocolo, el cual contiene uno o más agentes que procesan paquetes en cada nivel de la pila 120 de protocolo. Como resultado, existe al menos un gestor 118 de protocolo y una pila 120 de protocolo por cada tipo de protocolo que está siendo procesado por la tarjeta 102 de control. Por ejemplo, una pila de protocolo de IP podría incluir, desde la parte inferior hasta la superior, una capa 120a de Ethernet, una capa 120b de IP, una capa 120c de UDP y una capa 120d de RTP. De manera similar, el gestor 118 de protocolo tiene un gestor 118a, 118b, 118c y 118d de gestor de capa correspondiente por cada capa de la pila 120 de protocolo. El número de capas de protocolo y de gestores de capa dependerá del protocolo que se esté procesando. El controlador 114, el gestor 118 de protocolo y la pila 120 de protocolo proporcionan el procesamiento de protocolo estándar para la presente invención.

Las tarjetas 102 de control de la presente invención incluyen también un controlador 122 de FFF, el cual incluye al menos un ejemplo de gestor 124 de FFF y una aplicación 126 de FFF. El controlador 122 de FFF está acoplado comunicativamente al controlador 114, a la interfaz 116 de PCI y al gestor 118 de protocolo. La operación del controlador 122 de FFF, del gestor 124 de FFF y de la aplicación 126 de FFF, va a ser descrita con mayor detalle en relación con la Figura 2. Las tarjetas 102 de control pueden incluir también un motor 128 de FFF acoplado comunicativamente a la interfaz 112 de Ethernet, a la pila 120 de protocolo, al controlador 122 de FFF y a una base de datos 130 de FFF. La operación del motor 128 de FFF y de la base de datos 130 de FFF, va a ser descrita con mayor detalle con referencia a la Figura 3. En otro caso, la pila 120 de protocolo está acoplada comunicativamente a la interfaz 112 de Ethernet.

Las una o más tarjetas 104 de entrada incluyen una interfaz 132 de red de entrada para recibir comunicaciones 134, una interfaz 136 de PCI acoplada comunicativamente al bus 108 de PCI y un excitador 138 de entrada acoplado comunicativamente a la interfaz 132 de red de entrada y a la interfaz 136 de PCI. Las tarjetas 104 de entrada incluyen también uno o más motores 140 de FFF acoplados comunicativamente a la interfaz 132 de red de entrada, a la interfaz 136 de PCI, al excitador 138 de entrada y a una base de datos 142 de FFF. La operación del motor 140 de FFF y de la base de datos 142 de FFF va a ser descrita con mayor detalle con referencia a la Figura 3. Las una o más tarjetas 106 de salida incluyen una interfaz 144 de red de salida para recibir comunicaciones 146, una interfaz 148 de PCI acoplada comunicativamente al bus 108 de PCI y un excitador 150 de salida acoplado comunicativamente a la interfaz 144 de red de salida y a la interfaz 148 de PCI.

Las una o más aplicaciones 126 de FFF monitorizan el gestor 118 de protocolo y la pila 120 de protocolo para detectar nuevos flujos y los cambios en los flujos existentes. Las aplicaciones 126 de FFF trabajan con el (los) gestor(es) 118a-d de capa para detectar, crear y borrar reglas de identificación, validación y/o transformación para un flujo particular. Las aplicaciones 126 de FFF están configuradas para construir un conjunto completo de reglas de ajuste de caudal, las cuales pueden incluir una o más reglas de identificación, una o más reglas de validación, y/o una o más reglas de transformación, para su instalación en un motor de FFF, tal como el 128 o el 140. El gestor 124 de FFF maneja la interfaz de gestión para el sistema de FFF y controla la comunicación entre las aplicaciones 126 de FFF y los motores 128 y 140 de FFF. Además, el gestor 124 de FFF acepta peticiones de adición, eliminación y/o averiguación de flujo procedentes de las aplicaciones 126 de FFF y traduce la(s) petición(es) a un formato entendible por los motores 128 y 140 de FFF. Adicionalmente, el gestor 124 de flujo comunica reglas de identificación, validación y/o transformación a los motores 128 y 140 de FFF, los cuales almacenan las reglas en las bases de datos 130 y 142 de FFF, respectivamente. Un gestor de bases de datos de FFF (no representado) controla las bases de datos 130 y 142 de FFF. Los motores 128 y 140 de FFF, y el gestor de bases de datos de FFF (no representado), han sido previstos a modo de una o más rutinas de librería en cualquier excitador, tal como el excitador 138 de entrada, que está configurado para participar en FFF. La(s) rutina(s) de librería está(n) implementada(s) en software o está(n) acelerada(s) por hardware.

El gestor de bases de datos de FFF (no representado) almacena las reglas de identificación, validación y/o transformación en forma de árbol de decisión para facilitar un procesamiento rápido de trama por parte de los motores 128 y 140 de FFF. El árbol de decisión incluye uno o más nodos de los que cada nodo es una tabla hash. El uso de tablas hash en árboles de decisión es bien conocido por los expertos en la materia. Adicionalmente, los motores 128 y 140 de FFF determinan si un paquete de datos forma parte de un flujo identificado por emparejamiento de una trama entrante respecto a patrones de flujo existentes, que están almacenados a modo de una o más reglas de identificación almacenadas en las bases de datos 130 y 142 de FFF. Las una o más reglas de identificación pueden incluir una secuencia de patrones de datos, máscaras de datos, y/o derivaciones relativas de los paquetes de datos de IP que identifiquen unívocamente los paquetes de datos de IP como pertenecientes a un flujo específico. Los motores 128 y 140 de FFF pueden también validar la trama de llegada utilizando una o más reglas de validación almacenadas en las bases de datos 130 y 142 de FFF. Las una o más reglas de validación se utilizan para verificar además que el paquete reúne los requisitos para el FFF. Si el paquete es sucesivamente identificado y validado, los motores 128 y 140 de FFF lo procesan utilizando una o más reglas de transformación almacenadas en las bases de datos 130 y 142 de FFF. Típicamente, una regla de transformación final contiene una identificación de una interfaz de salida para que los paquetes transformados sean transmitidos. Una vez transformado, el paquete es enviado directamente a la tarjeta 106 de salida a través del bus 108 de PCI y de las interfaces 136 y 148 de PCI. Si, no obstante, el sistema contiene un puerto de salida que coexista en una misma placa de circuito impreso con los motores 128 y 140 de FFF, el paquete puede ser enviado directamente al puerto de salida.

Tras la inicialización del sistema, las bases de datos 130 y 142 de FFF (árbol de decisión) están típicamente vacías. A continuación de la inicialización del sistema, las bases de datos 130 y 142 (árbol de decisión) pueden ser cargadas con la información almacenada. Si la información de las bases de datos almacenadas no está disponible o no existe, los motores 128 y 140 de FFF fallan en cuanto al enrutamiento de los paquetes asociados a un flujo particular hasta la pila 120 de protocolo para su procesamiento estándar hasta que las bases de datos 130 y 142 de FFF son cargadas con el ajuste de caudal. Además, las bases de datos 130 y 142 de FFF (árbol de decisión) se modifican dinámicamente según se añade, modifica y extrae ajuste de caudal desde los motores 128 y 140 de FFF.

Haciendo ahora referencia a ambas Figuras 1 y 2, la Figura 2 representa un diagrama de flujo de un controlador de envío de flujo rápido de acuerdo con la presente invención. El proceso para crear, actualizar y borrar las reglas de ajuste de caudal (identificación, verificación y transformación) se inicia en el bloque 200. El controlador 122 de FFF, y más específicamente las una o más aplicaciones 126 de FFF, monitoriza el procesamiento estándar de paquetes en la pila 120 de protocolo por medio del gestor 118 de protocolo en el bloque 202. Si el controlador 122 de FFF no ha recibido ningún dato externo, tal como información de establecimiento de llamada, según se determina en el bloque 204 de decisión, y se ha detectado un nuevo flujo, según se determina en el bloque 206 de decisión, las una o más reglas de identificación, una o más reglas de verificación o validación y las una o más reglas de transformación, son creadas en el bloque 208. El gestor 124 de FFF realiza a continuación las reglas de ajuste de flujo (una o más reglas de identificación, una o más reglas de verificación o validación, y las una o más reglas de

transformación) disponibles para el motor 128 ó 140 de FFF que está manejando el flujo detectado en el bloque 210. Como resultado, el motor 128 ó 140 de FFF es habilitado en el bloque 212. A continuación, el controlador 122 de FFF continúa monitorizando el procesamiento estándar de paquetes a través de las aplicaciones 126 de FFF en el bloque 202.

5 Si, no obstante, el controlador 122 de FFF recibe datos externos, tal como información de establecimiento de llamada, según se determina en el bloque 204 de decisión, y los datos externos son suficientemente predictivos para permitir la creación con anterioridad de las reglas de ajuste de caudal para la llamada, según se determina en el bloque 214 de decisión, las reglas de establecimiento de ajuste de caudal (una o más reglas de identificación, una o más reglas de verificación o validación, y las una o más reglas de transformación) son creadas en el bloque 208 y el proceso continúa según se ha descrito anteriormente. Si, no obstante, los datos externos no son suficientemente predictivos para permitir la creación con anterioridad de las reglas de ajuste de caudal para la llamada, según se determina en el bloque 214 de decisión, el controlador 122 de FFF continúa monitorizando el procesamiento estándar de paquetes a través de las aplicaciones 126 de FFF en el bloque 202.

15 Si, no obstante, no se ha detectado un nuevo flujo, según se determina en el bloque 206 de decisión, pero se ha detectado un cambio en un flujo ya existente, según se determina en el bloque 216 de decisión, las una o más reglas de identificación, una o más reglas de verificación o validación y las una o más reglas de transformación, son actualizadas en el bloque 218. El gestor 124 de FFF realiza a continuación la actualización de las reglas de ajuste de caudal (una o más reglas de identificación, una o más reglas de verificación o validación y las una o más reglas de transformación) disponibles para el motor 128 ó 140 de FFF que está manejando el flujo detectado en el bloque 220. A continuación, el controlador 122 de FFF continúa monitorizando el procesamiento estándar de paquetes a través de las aplicaciones 126 de FFF en el bloque 202.

25 Si, no obstante, no se ha detectado un cambio en un flujo existente, según se determina en el bloque 216 de decisión, sino que se ha detectado una condición de retardo, terminación o reposición, según se determina en el bloque 222 de decisión, las una o más reglas de identificación, una o más reglas de verificación o validación y las una o más reglas de transformación son puestas a cero o borradas en el bloque 224. Como resultado, el motor 128 ó 140 de FFF aplicable es deshabilitado en el bloque 226. A continuación, el controlador 122 de FFF sigue monitorizando el procesamiento estándar de paquetes a través de las aplicaciones 126 de FFF en el bloque 202.

35 Haciendo ahora referencia a ambas Figuras 1 y 3, la Figura 3 representa un diagrama de flujo de un motor de envío de flujo rápido de acuerdo con la presente invención. El motor 128 ó 140 de FFF que procesa los paquetes, se inicia en el bloque 300. El motor 128 ó 140 de FFF recibe el paquete en el bloque 302. Si el motor 128 ó 140 de FFF está deshabilitado o las reglas de ajuste de caudal no están cargadas en las bases de datos 130 ó 142 de FFF, según se determina en el bloque 304 de decisión, el paquete es procesado y enviado utilizando el proceso estándar en el bloque 306. Esto significa que el paquete es enviado a la pila 120 de protocolo para su procesamiento. A continuación, si el motor 128 ó 140 de FFF recibe una condición de retardo, terminación o reposición, según se determina en el bloque 308 de decisión, las una o más reglas de identificación, una o más reglas de verificación o validación y las una o más reglas de transformación, son puestas a cero o borradas en el bloque 310 y el proceso del motor 128 ó 140 de FFF finaliza en el bloque 312. Como resultado, el motor 128 ó 140 de FFF aplicable es deshabilitado.

45 Si, no obstante, el motor 128 ó 140 de FFF es habilitado y las reglas de ajuste de caudal son cargadas en la base de datos 130 ó 142 de FFF, según se determina en el bloque 304 de decisión, el paquete es comprobado respecto a las una o más reglas de identificación en el bloque 314. Si el proceso de identificación no tiene éxito, según se determina en el bloque 316 de decisión, el paquete es procesado y enviado utilizando el proceso estándar en el bloque 306 y el proceso continúa según ha sido previamente descrito. Esto significa que el paquete es enviado a la pila 120 de protocolo para su procesamiento. Si, no obstante, el proceso de identificación ha tenido éxito, según se determina en el bloque 316 de decisión, el paquete es comprobado frente a las una o más reglas de validación o verificación en el bloque 318. Si el proceso de validación o verificación no tiene éxito, según se determina en el bloque 320 de decisión, el paquete es procesado y enviado utilizando el proceso estándar en el bloque 306 y el proceso continúa según ha sido descrito con anterioridad. Esto significa que el paquete es enviado a la pila 120 de protocolo para su procesamiento. Si, no obstante, el proceso de validación o verificación ha tenido éxito, según se determina en el bloque 320 de decisión, el paquete es procesado utilizando una o más de las reglas de transformación en el bloque 322 y el paquete procesado o transformado es enviado directamente al puerto de salida asignado en el bloque 324. A continuación, el proceso realiza un bucle de retorno hasta el bloque 308 de decisión, según se ha descrito anteriormente, y lo más probable es que reciba el siguiente paquete en el bloque 302 y repita el proceso.

60 Haciendo ahora referencia a la Figura 4, se va a describir un conmutador 400 de comunicaciones conforme a la presente invención. El conmutador 400 de red de paquetes puede ser usado para procesar VoIP, voz sobre Frame Relay ("VoFR") y otros tipos de llamadas. Además, el conmutador de red 400 de paquetes es similar a un conmutador de modo de transferencia asíncrona ("ATM"). El ATM es una tecnología orientada de conexión utilizada tanto en entornos de red de área local ("LAN") como de red de área amplia ("WAN"). Consiste en una tecnología de

conmutación rápida de paquetes que permite una asignación libre de capacidad a cada canal. El conmutador 400 de red de paquetes incluye una o más tarjetas 402a y 402b de entrada, una o más tarjetas 404 de procesamiento de señal, una o más tarjetas 406 de control, una o más tarjetas 408a y 408b de salida, una estructura 410 de conmutación y un bus 412 de TDM. Cada tarjeta 404 de procesamiento de señal contiene una batería de procesadores de señal digital ("DSP") (no representados) y cada tarjeta 406 de control contiene uno o más procesadores (no representados). La estructura 410 de conmutación acopla comunicativamente las tarjetas 402 de entrada, las tarjetas 404 de procesamiento de señal, las tarjetas 406 de control y las tarjetas 408 de salida entre sí. El bus 412 de TDM acopla también comunicativamente las tarjetas 402 de entrada, las tarjetas 404 de procesamiento de señal, las tarjetas 406 de control y las tarjetas 408 de salida entre sí. Con preferencia, las tarjetas 402, 404, 406 y 408 pueden ser insertadas en cualquier orden dentro del conjunto 400 de red de paquetes. Además, el conmutador 400 de red de paquetes podría incluir cantidades suficientes de tarjetas redundantes para que sirvan como tarjetas de seguridad en caso de que una tarjeta 402, 404, 406 y 408 falle.

La función principal de un conmutador 400 de red de paquetes consiste en remitir células de datos de usuario desde puertos de entrada hasta los puertos de salida apropiados. Cuando una llamada o una comunicación debe ser manejada por el conmutador 400 de red de paquetes, un controlador de red (no representado) proporciona a la tarjeta 408 de control la información necesaria de establecimiento de llamada. La tarjeta 408 de control utiliza esta información de establecimiento de llamada para asignar un puerto en las tarjetas 402a o 402b de entrada para recibir la llamada desde la Red de Telefonía Conmutada Pública ("PSTN"), un DSP dentro de la tarjeta 404 de procesamiento para procesar la llamada, y un puerto en las tarjetas 408a o 408b de salida para enviar la llamada a la red de IP (no representada). Cada tarjeta 408 de control tiene su propia memoria y de ese modo evita los problemas típicos asociados a la memoria compartida, tal como llamadas recursivas y problemas de sincronización e interrupción. Las comunicaciones o mensajes en base a TDM entran a través de las tarjetas 402a o 402b de entrada y son enrutadas hasta la tarjeta 404 de procesamiento apropiada a través del Bus 412 de TDM. Los DSPs de la tarjeta 404 de procesamiento convierten los mensajes entre formatos de información analógicos y digitales, y proporcionan funciones de compresión digital y conmutación. En una realización, cada tarjeta 404 de procesamiento está capacitada para procesar 1024 sesiones simultáneas. La tarjeta 404 de procesamiento envía entonces los mensajes desde el DSP hasta la estructura 410 de conmutador de célula, la cual es principalmente responsable del enrutamiento y transferencia de mensajes o células de datos, la unidad de transmisión básica entre elementos de conmutación. La estructura 410 de conmutación puede proporcionar también memoria de almacenamiento intermedio de célula concentración y multiplexado de tráfico, redundancia para tolerancia de fallo, radiodifusión o multidifusión, y planificación de célula en base a prioridades de retardo y monitorización de congestión. La estructura 410 de conmutación enruta finalmente los mensajes hasta las tarjetas 408a o 408b de salida. En una realización, cada tarjeta 408 de salida está capacitada para manejar al menos 8000 llamadas. Las tarjetas 408a y 408b de salida envían típicamente los mensajes a una Ethernet de un gigabits (no representada). Como su nombre indica, la Ethernet de un gigabits soporta tasas de datos de un (1) gigabits (1.000 megabits) por segundo.

Haciendo ahora referencia a la Figura 5, se ha mostrado un diagrama esquemático de un conmutador 500 de red de paquetes de acuerdo con una realización de la presente invención. El conmutador 500 de red de paquetes incluye tarjetas 502a y 502b de entrada acopladas comunicativamente a un bus 504 de TDM. El bus 504 de TDM está acoplado comunicativamente a un número de DSPs 506a, 506b, 506c, ..., 506n. Los DSPs 506a, 506b, 506c, ..., 506n están configurados típicamente según una batería de DSPs situados sobre una o más tarjetas de procesamiento de señal. Cada uno de los DSP 506a, 506b, 506c, ..., 506n está acoplado comunicativamente a un motor 508a, 508b, 508c, ..., 508n de FFF, cada uno de los cuales posee una base de datos (no representada) de FFF según se ha descrito anteriormente. Cada motor 508a, 508b, 508c, ..., 508n de FFF está acoplado comunicativamente a una estructura 510 de conmutación. La estructura 510 de conmutación está acoplada comunicativamente a tarjetas 512a y 512b de salida. El conmutador 500 de red de paquetes incluye también una o más CPUs 514, las cuales están situadas típicamente sobre una o más tarjetas de control. La CPU 514 está acoplada comunicativamente a las tarjetas 502a y 502b de entrada, a los DSPs 506a, 506b, 506c, ..., 506n, y a las tarjetas 512a y 512b de salida. Un controlador 516 de FFF, que incluye un gestor (no representado) de FFF y una o más aplicaciones (no representadas) de FFF, está acoplado comunicativamente a la CPU 514 y a los motores 508a, 508b, 508c, ..., 508n de FFF.

Durante la conversión de una comunicación 518a o 518b basada en multiplexado por división de tiempo ("TDM") en una comunicación 520a o 520b basada en IP, la CPU 514 recibe instrucciones 522 de señalización para la llamada y asigna un puerto de la tarjeta 502a, 502b de entrada, y un puerto de la tarjeta 510a, 510b de salida, y un DSP 506a, 506b, 506c, ..., 506n para procesar la llamada. De forma similar, el controlador 516 de FFF (gestor de FFF) asigna un motor 508a, 508b, 508c, ..., 508n de FFF para procesar el paquete después de que éste haya sido creado por el DSP 506a, 506b, 506c, ..., 506n respectivo. El DSP 506a, 506b, 506c, ..., 506n recibe información de establecimiento de llamada desde la CPU 514 y solicita una plantilla desde la CPU 514 en base a la información de establecimiento de llamada o al tipo de portador. El DSP 506a, 506b, 506c, ..., 506n recibe y carga la plantilla. La plantilla contiene los parámetros operativos necesarios para configurar apropiadamente el DSP 506a, 506b, 506c, ..., 506n para procesar un cierto tipo de llamada. La carga en tiempo real de plantillas permite que cada DSP 506a, 506b, 506c, ..., 506n procese cualquier tipo de llamada. El uso de plantillas permite también que el conmutador 500 de red de paquetes sea actualizado para procesar nuevos tipos de llamadas o procesar más eficazmente tipos de

llamadas ya existentes mediante actualizaciones o descargas de software. Adicionalmente, el conmutador 500 de red de paquetes puede usar la asignación de plantillas para controlar dinámicamente la asignación de ancho de banda a los diversos tipos de llamadas para asegurar estándares de QoS y/o el cumplimiento de restricciones de licencia.

5 A continuación, el DSP 506a, 506b, 506c, ..., 506n procesa los datos modulados por código de pulso ("PCM") y lleva a cabo una discriminación adicional de los datos para determinar si se requiere una plantilla diferente. Si se necesita cambiar la plantilla, el DSP 506a, 506b, 506c, ..., 506n solicita una plantilla diferente, y recibe y carga la plantilla diferente. Por ejemplo, la información de establecimiento de llamada puede indicar que el tipo de soporte de la
10 llamada es voz incluso aunque el tipo de soporte pueda ser o bien voz o bien fax. De ese modo, si el DSP 506a, 506b, 506c, ..., 506n reconoce mediante discriminación adicional de los datos de PCM que la llamada es realmente un fax en vez de una llamada de voz, el DSP 506a, 506b, 506c, ..., 506n solicitará una plantilla diferente con el fin de configurar apropiadamente el DSP 506a, 506b, 506c, ..., 506n para que procese el fax.

15 Una vez que se ha cargado la plantilla apropiada, el DSP 506a, 506b, 506c, ..., 506n recibe los datos de la llamada desde el puerto de la tarjeta 502a, 502b de entrada asignado a través del bus 504 de TDM. El DSP 506a, 506b, 506c, ..., 506n comprime entonces los datos de la llamada y crea una porción de datos del paquete. El DSP 506a, 506b, 506c, ..., 506n puede crear también una o más muestras digitales a partir de los datos de llamada comprimidos y crear la porción de datos del paquete utilizando las una o más muestras digitales. El DSP 506a, 506b,
20 506c, ..., 506n crea también una o más cabeceras, tal como una cabecera de RTP, una cabecera de UDP, una cabecera de IP y una cabecera de MAC, utilizando los datos de la llamada y la información de establecimiento de llamada. Más específicamente, las cabeceras de RTP y de UDP son generadas a partir de los datos de la llamada mientras que las cabeceras de IP y de MAC son generadas a partir de la información de establecimiento de llamada. Obsérvese que el DSP 506a, 506b, 506c, ..., 506n no se limita a la creación de ninguna cabecera específica, tal como una cabecera de RTP, una cabecera de UDP, una cabecera de IP o una cabecera de MAC, sino que puede ser también usado para crear cualquier cabecera necesaria para el suministro apropiado de un paquete.

El DSP 506a, 506b, 506c, ..., 506n fija a continuación las una o más cabeceras a la porción de datos del paquete. El DSP 506a, 506b, 506c, ..., 506n envía el paquete completo (datos más cabeceras) al motor 508a, 508b, 508c, ...,
30 508n de FFF asignado. El motor 508a, 508b, 508c, ..., 508n de FFF asignado procesa el paquete utilizando una o más reglas de transformación en caso de que el paquete satisfaga una o más reglas de identificación y/o de verificación o reglas de validación. En otro caso, el motor 508a, 508b, 508c, ..., 508n de FFF asignado procesa el paquete utilizando un proceso estándar en caso de que el paquete no satisfaga las una o más reglas de identificación y/o de verificación o reglas de validación. El proceso estándar ha sido descrito en lo que antecede con referencia a la Figura 1. Los procesos de FFF han sido descritos en lo que antecede con referencia a las Figuras 1-3. El paquete procesado es enviado a continuación al puerto de la tarjeta 512a, 512b de salida apropiada por medio de la estructura 510 de conmutación para su transmisión al exterior por la red de IP.

Las realizaciones y ejemplos expuestos en la presente memoria han sido presentados para explicar mejor la presente invención y su aplicación práctica y para permitir con ello que los expertos en la materia realicen y utilicen la invención. Sin embargo, los expertos en la materia reconocerán que la descripción y los ejemplos que anteceden han sido presentados con fines ilustrativos y de ejemplo solamente. No se pretende que la descripción realizada sea exhaustiva o limite la invención a la forma precisa divulgada. Son posibles muchas modificaciones y variaciones a la vista de las enseñanzas anteriores sin apartarse del alcance de las reivindicaciones que siguen.

45

REIVINDICACIONES

- 5 1.- Un método para procesar un paquete de un flujo de paquetes, que comprende las etapas de:
- 10 recibir el paquete;
 procesar (322) el paquete utilizando una o más reglas de transformación sin procesamiento de protocolo en capas en caso de que el paquete satisfaga una o más reglas de identificación para el flujo, en el que una regla de transformación final de las reglas de transformación contiene una identificación de una interfaz de salida para el flujo, y
 15 procesar (306) el paquete utilizando un proceso estándar de procesamiento de protocolo en capas en caso de que el paquete no satisfaga las una o más reglas de identificación para el flujo, enviando el paquete a una pila (120) de protocolo para la conmutación del paquete.
- 2.- El método según se expone en la reivindicación 1, en el que las una o más reglas de identificación comprenden además una o más reglas de validación.
- 3.- El método según se expone en la reivindicación 1, que comprende además la etapa (208) de crear las una o más reglas de identificación y las una o más reglas de transformación en base a datos externos.
- 20 4.- El método según se expone en la reivindicación 1, en el que los datos externos consisten en una información de establecimiento de llamada.
- 5.- El método según se expone en la reivindicación 1, que comprende además la etapa (324) de enviar el paquete a un puerto de salida después de que el paquete haya sido procesado utilizando las una o más reglas de transformación.
- 25 6.- El método según se expone en la reivindicación 1, que comprende además las etapas de:
- 30 monitorizar (202) el proceso estándar para detectar uno o más flujos de paquetes, y
 crear las una o más reglas de identificación y las una o más reglas de transformación en caso de que se detecte un flujo de paquetes.
- 7.- El método según se expone en la reivindicación 6, que comprende además la etapa de actualizar las una o más reglas de identificación y las una o más reglas de transformación en caso de que se detecte un cambio en el flujo de paquetes.
- 35 8.- El método según se expone en la reivindicación 1, en el que las una o más reglas de identificación comprenden un árbol de decisión.
- 40 9.- El método según se expone en la reivindicación 8, en el que el árbol de decisión comprende uno o más nodos y cada nodo es una tabla hash.
- 10.- El método según se expone en la reivindicación 1, en el que cada regla de transformación comprende una instrucción de procesamiento.
- 45 11.- El método según se expone en la reivindicación 1, en el que el paquete es un paquete de IP.
- 12.- El método según se expone en la reivindicación 1, en el que el paquete es un paquete de Ethernet.
- 50 13.- Un producto de programa de ordenador materializado sobre un medio legible con ordenador para el procesamiento de un paquete de un flujo de paquetes cuando se carga en un conmutador de comunicaciones, comprendiendo el producto de programa de ordenador segmentos de código para ejecutar las etapas de:
- 55 recibir el paquete
 procesar (322) el paquete utilizando una o más reglas de transformación sin procesamiento de protocolo en capas en caso de que el paquete satisfaga una o más reglas de identificación para el flujo, en el que una regla de transformación final de las reglas de transformación contiene una identificación de una interfaz de salida para el flujo, y
 60 procesar (306) el paquete utilizando un proceso estándar de procesamiento de protocolo en capas en caso de que el paquete no satisfaga las una o más reglas de identificación para el flujo, enviando el paquete a una pila (120) de protocolo para la conmutación del paquete.
- 14.- El producto de programa de ordenador según se expone en la reivindicación 13, en el que las una o más reglas de identificación comprenden además una o más reglas de validación.
- 65

- 5 15.- El producto de programa de ordenador según se expone en la reivindicación 13, que comprende además un segmento de código para ejecutar la etapa de crear las una o más reglas de identificación y las una o más reglas de transformación en base a datos externos.
- 16.- El producto de programa de ordenador según se expone en la reivindicación 13, en el que los datos externos consisten en una información de establecimiento de llamada.
- 10 17.- El producto de programa de ordenador según se expone en la reivindicación 13, que comprende además un segmento de código para ejecutar la etapa de enviar el paquete a un puerto de salida después de que el paquete haya sido procesado utilizando las una o más reglas de transformación.
- 18.- El producto de programa de ordenador según se expone en la reivindicación 13, que comprende además:
- 15 un segmento de código para ejecutar la etapa de monitorizar el proceso estándar para detectar uno o más flujos de paquetes, y
un segmento de código para ejecutar la etapa de crear las una o más reglas de identificación y las una o más reglas de transformación en caso de que se detecte un flujo de paquetes.
- 20 19.- El producto de programa de ordenador según se expone en la reivindicación 18, que comprende además un segmento de código para ejecutar la etapa de actualizar las una o más reglas de identificación y las una o más reglas de transformación en caso de que se detecte un cambio en el flujo de paquetes.
- 25 20.- El producto de programa de ordenador según se expone en la reivindicación 13, en el que las una o más reglas de identificación comprenden un árbol de decisión.
- 21.- El producto de programa de ordenador según se expone en la reivindicación 20, en el que el árbol de decisión comprende uno o más nodos, y cada nodo es una tabla hash.
- 30 22.- El producto de programa de ordenador según se expone en la reivindicación 13, en el que cada regla de transformación comprende una instrucción de procesamiento.
- 35 23.- El producto de programa de ordenador según se expone en la reivindicación 13, en el que el paquete es un paquete de IP.
- 24.- El producto de programa de ordenador según se expone en la reivindicación 13, en el que el paquete es un paquete de Ethernet.
- 40 25.- Un conmutador de comunicaciones, que comprende:
una o más tarjetas (104) de entrada;
una o más tarjetas (102) de control, teniendo cada tarjeta (102) de control al menos un procesador;
una o más tarjetas (106) de salida;
45 un bus (108) de comunicaciones que acopla comunicativamente las tarjetas (104) de entrada, las tarjetas (102) de control y las tarjetas (106) de salida entre sí, y
estando cada tarjeta (104) de entrada adaptada para recibir uno o más paquetes de un flujo de paquetes, para procesar cada paquete utilizando una o más reglas de transformación sin procesamiento de protocolo en capas en caso de que el paquete satisfaga una o más de las reglas de identificación para el flujo, en el que
50 una regla de transformación final de las reglas de transformación contiene una identificación de una interfaz de salida para el flujo, y para enviar cada paquete a uno de los procesadores para su procesamiento utilizando un proceso estándar de procesamiento de protocolo en capas en caso de que el paquete no satisfaga las una o más reglas de identificación para el flujo enviando el paquete a una pila (120) de protocolo para la conmutación del paquete.
- 55 26.- El conmutador de comunicaciones según se expone en la reivindicación 25, en el que las una o más reglas de identificación comprenden además una o más reglas de validación.
- 60 27.- El conmutador de comunicaciones según se expone en la reivindicación 25, en el que cada tarjeta de control comprende además un controlador de envío de flujo rápido que crea las una o más reglas de identificación y las una o más reglas de transformación en base a datos externos.
- 28.- El conmutador de comunicaciones según se expone en la reivindicación 25, en el que los datos externos consisten en una información de establecimiento de llamada.
- 65 29.- El conmutador de comunicaciones según se expone en la reivindicación 25, en el que los paquetes son

enviados a una de las tarjetas de salida después de que el paquete haya sido procesado utilizando las una o más reglas de transformación.

- 5 30.- El conmutador de comunicaciones según se expone en la reivindicación 25, en el que cada tarjeta de control comprende además un controlador de envío de flujo rápido que monitoriza el proceso estándar para detectar uno o más flujos de paquetes, y crea las una o más reglas de identificación y las una o más reglas de transformación en caso de que se detecte un flujo de paquetes.
- 10 31.- El conmutador de comunicaciones según se expone en la reivindicación 30, en el que el controlador de envío de flujo rápido actualiza las una o más reglas de identificación y las una o más reglas de transformación en caso de que se detecte un cambio en el flujo de paquetes.
- 15 32.- El conmutador de comunicaciones según se expone en la reivindicación 25, en el que las una o más reglas de identificación comprenden un árbol de decisión.
- 33.- El conmutador de comunicaciones según se expone en la reivindicación 32, en el que el árbol de decisión comprende uno más nodos y cada nodo consiste en una tabla hash.
- 20 34.- El conmutador de comunicaciones según se expone en la reivindicación 25, en el que cada regla de transformación comprende una instrucción de procesamiento.
- 35.- El conmutador de comunicaciones según se expone en la reivindicación 25, en el que el paquete es un paquete de IP.
- 25 36.- El conmutador de comunicaciones según se expone en la reivindicación 25, en el que el paquete es un paquete de Ethernet.

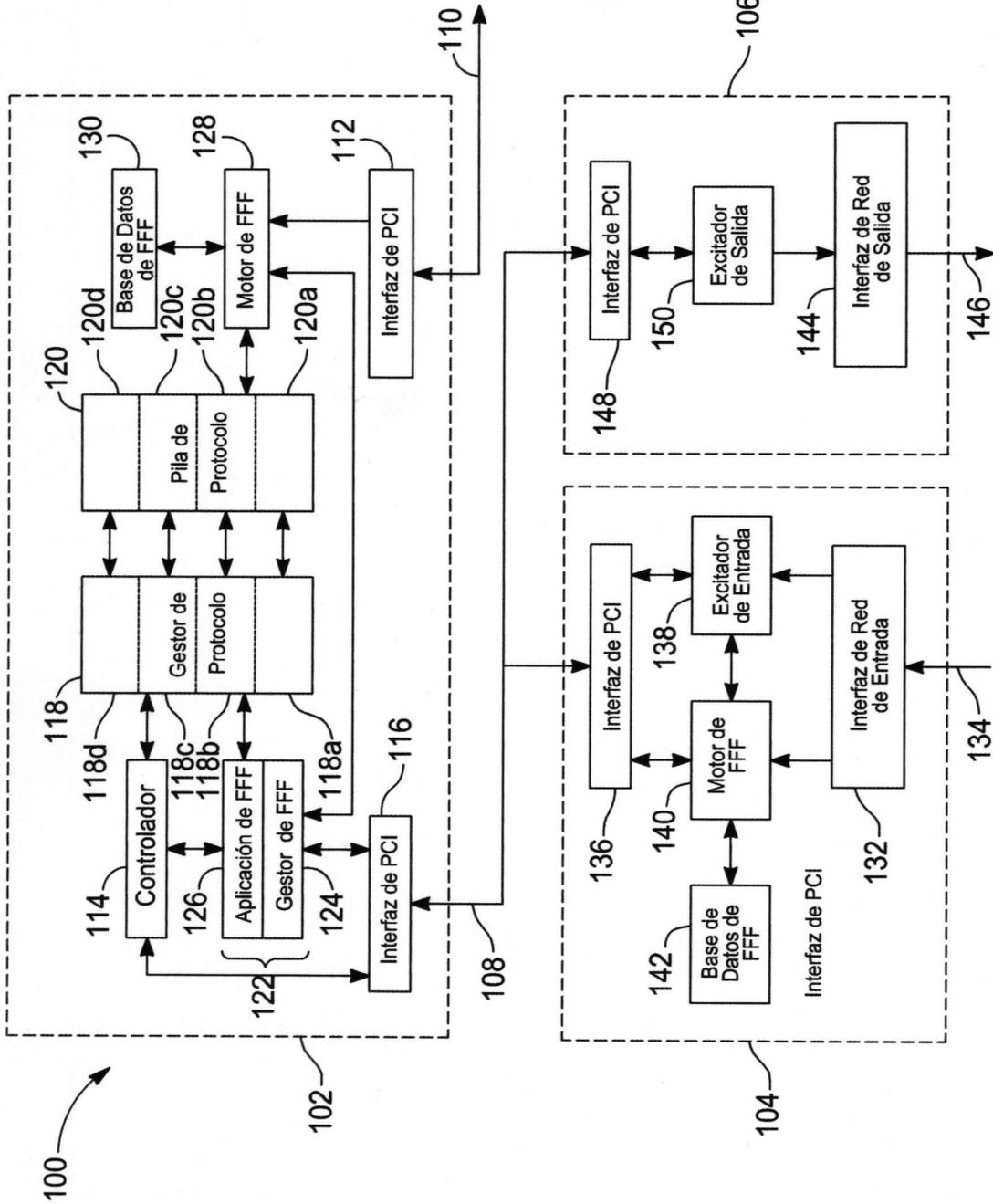


FIG. 1

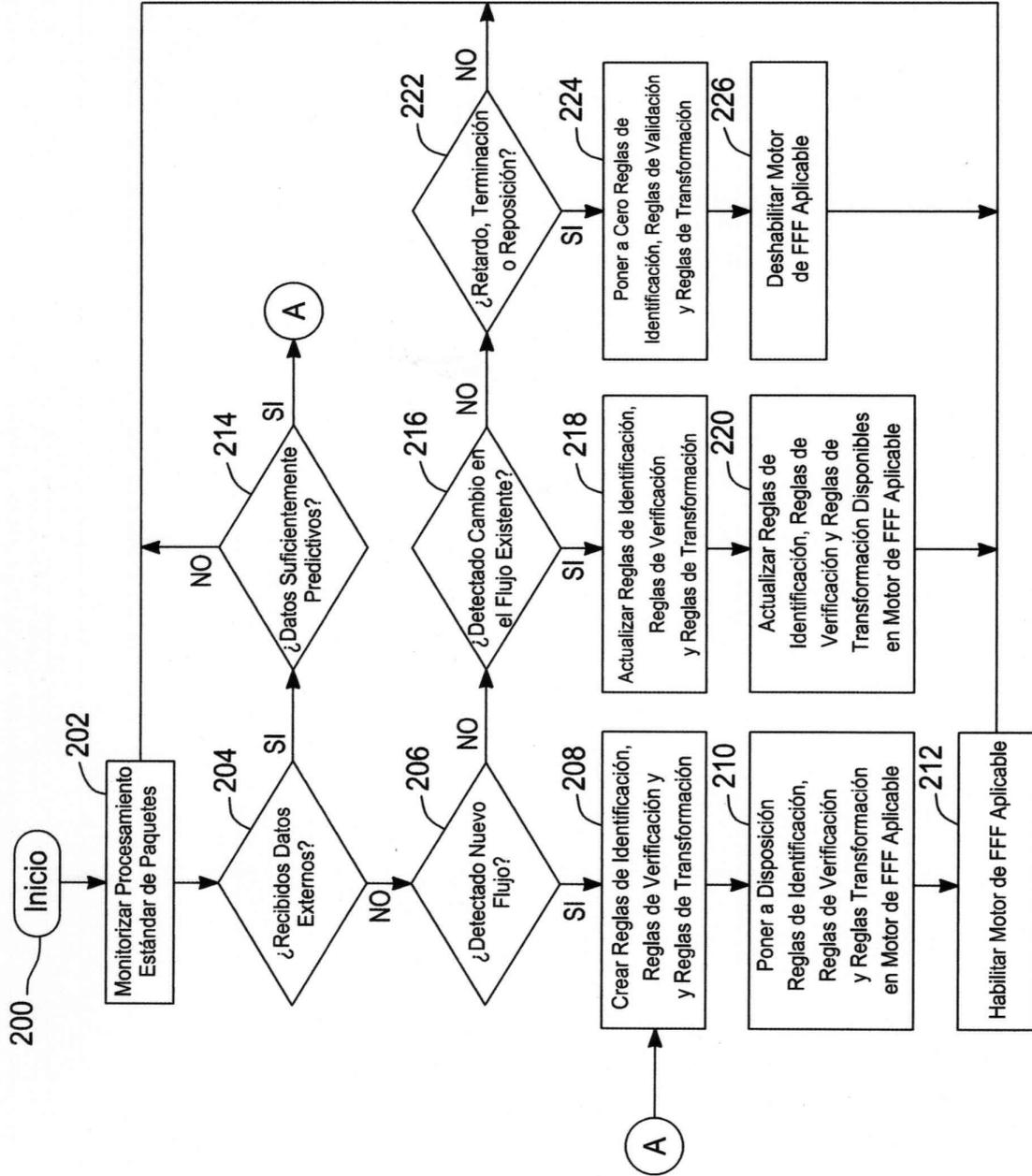


FIG. 2

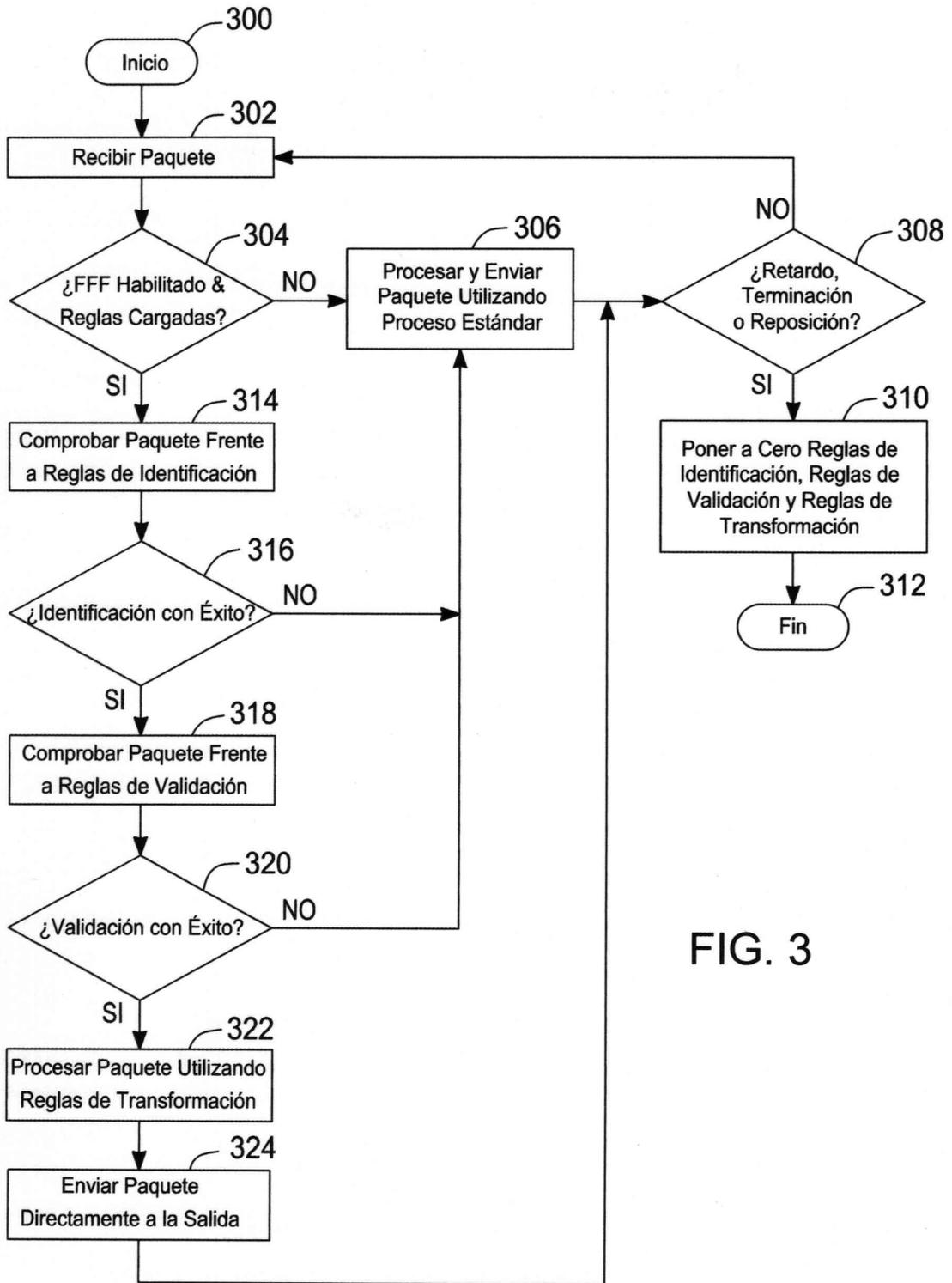


FIG. 3

