

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 672 999**

51 Int. Cl.:

**G06F 9/445** (2008.01)

**H04L 29/08** (2006.01)

**H04L 12/771** (2013.01)

**G06F 9/38** (2008.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **12.10.2013 PCT/CN2013/085138**

87 Fecha y número de publicación internacional: **17.04.2014 WO14056455**

96 Fecha de presentación y número de la solicitud europea: **12.10.2013 E 13846219 (7)**

97 Fecha y número de publicación de la concesión europea: **25.04.2018 EP 2907025**

54 Título: **Procedimientos para un procesamiento de paquetes ininterrumpido y sin pérdidas durante actualizaciones de software de sistema**

30 Prioridad:

**12.10.2012 US 201213650363**

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

**19.06.2018**

73 Titular/es:

**HUAWEI TECHNOLOGIES CO., LTD. (100.0%)  
Huawei Administration Building, Bantian  
Longgang District,  
Shenzhen, Guangdong 518057, CN**

72 Inventor/es:

**XU, JUN**

74 Agente/Representante:

**LEHMANN NOVO, María Isabel**

ES 2 672 999 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

Procedimientos para un procesamiento de paquetes ininterrumpido y sin pérdidas durante actualizaciones de software de sistema

5 Campo de la invención

La presente invención se refiere a una red de comunicación y, más en particular, a un procesamiento de paquetes ininterrumpido y sin pérdidas durante actualizaciones de software de sistema.

10 Antecedentes

Algunos dispositivos de red requieren una alta fiabilidad, tal como en redes de compañías de telecomunicaciones y grandes redes empresariales. En tales redes, el tiempo útil de conectividad de red puede superar el 99 por ciento aproximadamente, incluidas actualizaciones de software planificadas de sistema y tiempos de inactividad no planificados. Debido a actualizaciones de características en curso y a nuevas versiones de software de los proveedores de dispositivos de red, los dispositivos de red pueden necesitar una actualización de software planificada durante el funcionamiento de un dispositivo, lo que también se denomina en el presente documento actualización de software en servicio (ISSU). La industria ha utilizado diferentes esquemas ISSU para reducir el tiempo de inactividad de un sistema o dispositivo durante una ISSU planificada.

En un esquema ISSU, el dispositivo de red comprende placas duales de circuito impreso (o tarjetas de línea duales) o implementa una actualización en el plano de control pero no una actualización en el plano de datos. La ISSU implica cargar el nuevo software o la actualización de software (para el plano de control) en una de las placas duales, mientras sigue ejecutando la versión de software anterior en la otra placa. Cuando finaliza la actualización, el dispositivo puede conmutar a la placa con el nuevo software cargado o la actualización de software cargada. Este esquema ISSU está descrito en el documento "*Cisco IOS High Availability (HA) - In-Service Software Upgrade (ISSU)*", 2006, de Cisco Systems, Inc. En otro esquema ISSU, el dispositivo de red comprende placas duales en el plano de datos y/o placas duales en el plano de control que pueden usarse para cargar actualizaciones para el plano de datos y/o el plano de control y conmutar desde las versiones anteriores a las versiones actualizadas. Sin embargo, en función de este esquema, los dispositivos de red necesitan duplicar los recursos de sistema para las placas duales y pueden seguir experimentando cortos retardos o cortos tiempos de inactividad durante la conmutación entre placas. Además, permitir características de control de estado para la ISSU puede ser complicado y tener altos costes usando este esquema. Este esquema ISSU se describe en el documento "*Cisco ASR 1000 Series: ISSU Deployment Guide and Case Study*", 2008, de Cisco Systems, Inc. y en el documento "*NE40E Universal Service Router*", de Huawei. Los esquemas ISSU anteriores pueden aumentar el coste del sistema y la complejidad asociada al uso de placas duales. Además, los sistemas y dispositivos subyacentes que usan estos esquemas pueden seguir experimentando algunos efectos en los servicios de sistema, tal como cierto retardo en el procesamiento y reenvío de paquetes, durante tiempos de inactividad planificados para la actualización de software.

El documento US2006/0070054 describe la recepción de instrucciones de ejecución en un procesador que incluye múltiples núcleos de procesamiento programables integrados en una única pastilla, la selección de un subconjunto de al menos uno de los núcleos, y la carga de al menos una parte de las instrucciones de ejecución en el subconjunto de núcleos mientras que al menos uno de los otros núcleos sigue procesando paquetes recibidos.

45 Resumen

La presente invención se describe en la reivindicación de procedimiento 1 independiente y en la reivindicación de sistema 9 correspondiente. En un aspecto, un procedimiento implementado en un componente de red para una ISSU comprende seleccionar un núcleo de reenvío (FC) para su actualización a partir de una pluralidad de FC que se ejecutan en un plano de datos en una placa de circuito impreso del componente de red para procesar y reenviar una pluralidad de paquetes, y reajustar el FC seleccionado con una actualización de software, donde al menos uno de los otros FC sigue procesando y reenviando paquetes cuando se actualiza el FC seleccionado.

En otro aspecto, un componente de red comprende una placa de circuito impreso que comprende una pluralidad de FC, donde cada uno de los FC está configurado para gestionar y reenviar paquetes, y un procesador acoplado a la pluralidad de FC y configurado para seleccionar un FC a partir de la pluralidad de FC, desactivar el FC seleccionado para dejar de reenviar paquetes, cargar una versión de software de FC actualizada para el FC seleccionado, asignar nuevos recursos al FC seleccionado, configurar el FC seleccionado para que se una a la gestión y reenvío de paquetes con los otros FC ejecutándose en la placa de circuito impreso, y activar el FC seleccionado después de configurar el FC seleccionado, donde al menos uno de los FC está configurado para reenviar paquetes mientras que el FC seleccionado está desactivado.

En otro aspecto adicional, un componente de red comprende una placa de procesamiento que comprende al menos un procesador configurado para ejecutar un controlador en un plano de control y una pluralidad de FC en un plano de datos, donde cada uno de la pluralidad de FC está configurado para procesar y reenviar una pluralidad de

paquetes, y donde el controlador está configurado para desactivar uno o más de los FC para su actualización mientras que al menos uno o más FC restantes procesan y reenvían paquetes.

5 Estas y otras características se entenderán más claramente a partir de la siguiente descripción detallada tomada junto con los dibujos y reivindicaciones adjuntas.

Breve descripción de los dibujos

10 Para un entendimiento más completo de esta divulgación, a continuación se hace referencia a la breve descripción ofrecida a continuación, tomada junto con los dibujos adjuntos y la descripción detallada, donde números de referencia similares representan partes similares.

La FIG. 1 es un diagrama esquemático de una forma de realización de una arquitectura de reenvío de múltiples núcleos.

15 La FIG. 2 ilustra una forma de realización de un esquema ISSU para una arquitectura de reenvío de múltiples núcleos.

La FIG. 3 ilustra otra forma de realización de un esquema ISSU para una arquitectura de reenvío de múltiples núcleos.

La FIG. 4 es un diagrama de flujo de una forma de realización de un procedimiento ISSU.

La FIG. 5 es un diagrama esquemático de una forma de realización de una placa de circuito impreso.

20 La FIG. 6 es un diagrama esquemático de una forma de realización de un componente de red de propósito general.

Descripción detallada

25 Para comenzar, debe entenderse que aunque posteriormente se proporciona una implementación ilustrativa de una o más formas de realización, los sistemas y/o procedimientos dados a conocer pueden implementarse usando cualquier número de técnicas, ya sean actualmente conocidas o existentes. La divulgación no está limitada en modo alguno a las implementaciones ilustrativas, dibujos y técnicas ilustrados posteriormente, incluidos los diseños e implementaciones a modo de ejemplo ilustrados y descritos en el presente documento, pero puede modificarse dentro del alcance de las reivindicaciones adjuntas junto con su alcance total de equivalencias.

30 En el presente documento se describen sistemas, procedimientos y aparatos para proporcionar una ISSU para componentes de red, tales como encaminadores o componentes de reenvío de datos similares. El esquema ISSU dado a conocer puede usar una única placa de circuito impreso (también denominada placa o PCB) o sistemas o arquitecturas de una sola tarjeta de línea, y puede tener un coste y una complejidad reducidos en comparación con los esquemas ISSU anteriores que usan sistemas o arquitecturas de placa dual. Una ISSU, como se describe en el presente documento, puede implementarse para componentes de red que comprenden uno o más procesadores o  
35 unidades centrales de procesamiento (CPU) con una arquitectura de múltiples núcleos, es decir, donde cada CPU o chip de procesamiento puede ejecutar una pluralidad de núcleos para procesar y reenviar paquetes. En lugar de reajustar los núcleos para el procesamiento y reenvío de paquetes mediante una conmutación entre placas duales, como se realiza en otros esquemas ISSU, la ISSU propuesta puede gestionar de manera arbitraria o secuencial los núcleos individuales del sistema de placa única para actualizar un software en ejecución sin interrumpir el reenvío de  
40 paquetes mediante todos los núcleos. Específicamente, la ISSU puede gestionar un núcleo a la vez mientras mantiene a los otros núcleos de la placa procesando paquetes y gestionando el reenvío de datos. Al actualizar un núcleo cada vez de esta manera, la placa de múltiples núcleos para el componente de reenvío puede conseguir una ISSU con un procesamiento y un reenvío de paquetes ininterrumpido. Ventajas de la ISSU propuesta incluyen una pérdida nula o prácticamente nula de paquetes durante el procedimiento ISSU, reducir los tiempos de inactividad del  
45 sistema durante una actualización planificada, etapas ISSU de sistema más sencillas en comparación con otros esquemas ISSU, y ahorrar recursos de sistema y, por tanto, costes, usando una única placa de procesamiento en lugar de placas duales.

50 La FIG. 1 ilustra una forma de realización de una arquitectura de reenvío de múltiples núcleos 100 que puede usarse para gestionar, procesar y reenviar paquetes o datos en una red. La arquitectura de reenvío de múltiples núcleos 100 puede ser parte de, estar ubicada en o estar acoplada a un componente de red, tal como un encaminador, un conmutador, un puente o cualquier otro componente de procesamiento/reenvío de datos. El componente de red puede comprender una única placa o tarjeta de línea (LC) que comprende una o más CPU o procesadores, tal como un procesador de múltiples núcleos, que pueden implementar la arquitectura de reenvío de múltiples núcleos 100. La  
55 arquitectura de reenvío de múltiples núcleos 100 puede comprender un controlador 110 que corresponde a un plano de control y una pluralidad de núcleos de reenvío (FC) 120 que corresponden a un plano de datos. Por ejemplo, la arquitectura de reenvío de múltiples núcleos 100 puede comprender un número entero de n+1 FC 120 etiquetados desde el FC0 120 hasta el FCn 120 en la FIG. 1. El controlador 110 y los FC 102 pueden implementarse usando componentes de software (por ejemplo, programas de software) que se ejecutan en la una o más CPU. El controlador 110 puede ejecutarse en el plano o capa de control para gestionar y controlar los FC 120. Los FC 120  
60 pueden ejecutarse en el plano o capa de datos para gestionar, procesar y reenviar los datos o paquetes.

La FIG. 2 ilustra una forma de realización de un esquema ISSU 200 que puede implementarse para una arquitectura de reenvío de múltiples núcleos, tal como la arquitectura de reenvío de múltiples núcleos 100. El esquema ISSU 200 puede usarse para actualizar una pluralidad de FC 220 en el plano de datos (por ejemplo, actualizar las versiones de software de FC) de la arquitectura de reenvío de múltiples núcleos. Los FC 220 pueden ser similares a los FC 120. El esquema ISSU 200 puede conseguir una ISSU ininterrumpida, donde al menos algunos de los FC 220 pueden estar gestionando, procesando y reenviando datos o paquetes en cualquier momento durante la ISSU. Uno o varios FC 220 pueden desactivarse y actualizarse a la vez, mientras que uno o más de los FC 220 siguen ejecutando y gestionando datos (en el plano de datos).

Inicialmente, una pluralidad de  $n+1$  FC 220 (etiquetados como FC0 220, FC1 220, ..., FCn 220) puede estar ejecutándose antes de la ISSU. Durante la ISSU, un controlador 210 de la arquitectura de reenvío de múltiples núcleos puede desactivar uno de los FC en ejecución 220, por ejemplo el FC0 220, y liberar los recursos del FC 220, tales como la memoria asignada, la potencia de procesamiento y las conexiones de entrada/salida. El FC 220 puede seleccionarse de manera arbitraria, en función de su estado, tal como la cantidad de paquetes o de carga gestionada por el FC 220, o según algún orden o secuencia predeterminados de los FC 220. En este ejemplo se selecciona en primer lugar el FC0 220. Después de que el FC0 220 se desactive, el controlador 210 puede empezar a actualizar el FC0 220 desactivado. El proceso de actualización puede comprender especificar una o más opciones para que el proceso de reenvío vuelva a comenzar en el FC0 220 desactivado, por ejemplo incluyendo la imagen de arranque, comprobaciones de versiones y/o comprobaciones de estructuras de datos. Cuando todas las comprobaciones son satisfactorias y la actualización de software para el FC0 220 desactivado finaliza con éxito, el controlador 210 puede asignar nuevos recursos al FC0 220 y permitir que el FC0 220 recién cargado y actualizado se una a la gestión y al procesamiento de paquetes con los otros FC 220 activos (por ejemplo, FC1 220, ..., FCn 220).

Tras actualizar un FC 220 de este modo, el controlador 210 puede seguir el mismo procedimiento para actualizar los otros FC 220 restantes uno a uno, por ejemplo de manera secuencial en el mismo orden, hasta que se actualicen todos los FC 220. Por ejemplo, el controlador 210 puede, a continuación, desactivar el FC1 220, actualizar el FC1 220 (por ejemplo, realizar una actualización de software de FC) y cargar el FC1 220 recién actualizado para empezar a ejecutarse en el plano de datos y gestionar el reenvío de paquetes. El controlador 210 puede seguir actualizando los FC 220 restantes, por ejemplo, del FC2 220 al FCn 220, uno a uno de manera similar hasta que todos los FC 220 de la arquitectura de reenvío de múltiples núcleos se actualicen. La FIG. 2 muestra el progreso del procedimiento ISSU en múltiples fases a lo largo del tiempo, donde los FC 220 actualizados a medida que transcurre el tiempo se muestran como bloques sombreados en la FIG. 2.

En algunas formas de realización, si la capacidad de procesamiento del sistema es suficiente, por ejemplo si el sistema comprende suficientes CPU o procesadores en la placa o LC y/u otros recursos de sistema necesarios, el sistema puede tener más de un FC 220 sujeto al procedimiento ISSU descrito anteriormente. Por ejemplo, el controlador 210 puede ordenar a dos o más de los FC 220 que dejen de procesar paquetes y empezar a actualizar los FC 220 seleccionados casi al mismo tiempo. Los FC seleccionados 220 pueden actualizarse de manera independiente o prácticamente sin sincronización. Aunque algunos de los FC 220 se estén actualizando, el sistema puede tener suficientes FC 220 restantes que se estén ejecutando para procesar el reenvío de paquetes e impedir la pérdida de paquetes.

En el esquema ISSU 200, los FC individuales 220 pueden gestionarse mediante el controlador 210 en el plano de control para pasar de estar en ejecución (en las una o más CPU de la única placa) a estar suspendidos temporalmente. Durante la ejecución, un FC 220 puede extraer paquetes de una memoria intermedia para el procesamiento de paquetes y/o responder a una interrupción de entrada/salida (ES). La memoria intermedia de paquetes y la interrupción de ES pueden compartirse por todos o una pluralidad de los FC 220 en la placa. Cuando el controlador 210 envía un comando de reajuste para actualizar un FC 220 seleccionado, el FC 220 puede dejar de extraer paquetes de la memoria intermedia compartida, por ejemplo después de finalizar el procesamiento de sus paquetes pendientes. El FC 220 seleccionado también puede dejar de responder a la interrupción de ES, por ejemplo después de finalizar cualquier procesamiento de paquetes pendientes. Sin embargo, los FC 220 en ejecución restantes pueden no verse afectados por el comando de reajuste del controlador 210 para el FC 220 seleccionado y seguir su procesamiento normal en relación con la interrupción de ES y/o la memoria intermedia compartida. Por tanto, la suspensión del FC 220 seleccionado puede no provocar la pérdida de paquetes o un retardo considerable en el procesamiento de paquetes, donde los paquetes pueden seguir gestionándose por los FC 220 en ejecución restantes.

En algunos casos, el FC 220 seleccionado puede seguir necesitando más paquetes para completar su propio procesamiento de paquetes; por ejemplo, en el caso de la desfragmentación del protocolo de Internet (IP), la secuencia del protocolo de control de transmisión (TCP), u otros esquemas de transmisión, el FC 220 seleccionado puede seguir extrayendo los paquetes necesarios de la memoria intermedia compartida tras recibir el comando de reajuste desde el controlador 210. El FC 220 seleccionado puede extraer los paquetes de la memoria intermedia hasta que el FC 220 complete su propio procesamiento de paquetes usando los paquetes necesarios e interrumpir posteriormente su procesamiento habitual en relación con la memoria intermedia compartida. Por consiguiente, no se perderán paquetes. Como alternativa, en lugar de completar su propio procesamiento de paquetes, el FC 220

seleccionado puede insertar los paquetes extraídos en una segunda memoria intermedia compartida o cola de paquetes para que otros FC 220 en ejecución los capten y los procesen. El sistema puede tratar esta segunda memoria intermedia dedicada o cola como una cola de alta prioridad para el procesamiento de paquetes, mientras que otros FC 220 en ejecución pueden captar los paquetes de la cola para un procesamiento de alta prioridad.

5 Después de que finalice el propio procesamiento de paquetes del FC 220 seleccionado, el FC 220 puede liberar otros recursos compartidos usados por este FC 220, por ejemplo, instrucciones de bloqueo, entradas de tabla, estructuras de datos, memoria u otros recursos compartidos. El controlador 210 puede ayudar al FC 220 a liberar los recursos asociados, incluidos la memoria intermedia de paquetes y cualquier estructura de datos asociada,  
10 memoria, instrucciones de bloqueo o combinaciones de las mismas. El FC 220 seleccionado puede pasar por un ciclo delimitado, pero no definido de manera estricta, para completar las tareas anteriores. Después de que el FC 220 seleccionado libere los recursos, el FC 220 puede ajustar su ritmo para implementar las siguientes etapas del procedimiento ISSU. Después de liberar los recursos compartidos, el FC 220 puede pasar por una pluralidad de etapas de actualización de manera independiente a los otros FC 220 en ejecución, por ejemplo, reinicio e  
15 inicialización, para ponerse de nuevo en funcionamiento con una nueva versión de software durante su actualización de software. Los otros FC 220 en ejecución pueden seguir en el estado de ejecución hasta que el controlador 210 les ordene o indique que se reajusten para actualizarse y, por tanto, el sistema, en su totalidad, puede seguir procesando paquetes sin detenerse.

20 El FC 220 actualizado con la nueva versión de software puede iniciar después el acceso a la memoria intermedia compartida y/o a la interrupción de ES y extraer paquetes para su propio procesamiento y reenvío. En esta fase, el FC 220 actualizado puede realizar su(s) función(es) de reenvío de paquetes habitual(es), mientras que la totalidad del sistema puede seguir estando sujeto al procedimiento ISSU (es decir, mientras que uno o más otros FC 220 pueden actualizarse). El controlador 210 y las versiones de software de FC pueden ser compatibles antes y después  
25 de la ISSU.

La FIG. 3 ilustra una forma de realización de otro esquema ISSU 300 que puede implementarse para una arquitectura de reenvío de múltiples núcleos, tal como la arquitectura de reenvío de múltiples núcleos 100. El esquema ISSU 300 puede usarse para actualizar un controlador 310 en el plano de control y una pluralidad de FC  
30 320 en el plano de datos de la arquitectura de reenvío de múltiples núcleos. El controlador 310 y los FC 320 pueden ser similares al controlador 110 y a los FC 120, respectivamente. El esquema ISSU 300 puede conseguir una ISSU ininterrumpida, donde al menos algunos de los FC 320 pueden estar gestionando, procesando y reenviando datos o paquetes en cualquier momento durante la ISSU.

35 Inicialmente, una pluralidad de  $n+1$  FC 320 (etiquetados como FC0 320, FC1 320, ..., FCn 320) puede estar ejecutándose antes de la ISSU. Durante la ISSU, el controlador 310 y los FC 320 pueden actualizarse (por ejemplo, someterse a una actualización de software). Para conseguir una actualización progresiva del controlador y los FC en la que el procesamiento y el reenvío de paquetes no se vean interrumpidos y no se pierda ningún paquete, un nuevo controlador 315 con una versión de actualización de software de controlador puede introducirse en el plano de datos  
40 durante la ISSU. El nuevo controlador 315 puede usar cualquier procedimiento existente o admitido para recuperar información de estado del antiguo controlador 310. El nuevo controlador 315 puede generar sus tablas de reenvío locales y tablas compartidas por los FC. Cuando comienza el esquema ISSU 300, todos los FC 320 pueden estar asociados a (es decir, estar gestionados por) el antiguo controlador 310, y el nuevo controlador 315 puede no tener inicialmente ningún FC 320 asociado al mismo (fase 1 en la FIG. 3).

45 Después de que el nuevo controlador 315 se haya establecido y estabilizado (esté listo para empezar a funcionar), el antiguo controlador 310 puede empezar a desactivar un FC 320 cada vez (o algunos FC 320 a la vez si los recursos del sistema lo permiten), que puede seleccionarse de manera arbitraria, en función de su estado o según un orden predeterminado. El FC 320 seleccionado (por ejemplo, el FC0 320 en la fase 2) puede entonces interrumpir el procesamiento de paquetes y liberar sus recursos. El antiguo controlador 310 puede entonces liberar el control del FC 320 seleccionado. Usando comunicaciones permitidas de controlador a controlador, el nuevo controlador 315  
50 puede apropiarse entonces del FC 320 liberado, actualizar y designar el proceso de reenvío al FC 320, asignar recursos al FC 320 y permitir que el FC 320 recién cargado y actualizado empiece a gestionar paquetes (fase 2 en la FIG. 3). Las etapas de actualización de FC de los esquemas ISSU 300 pueden ser similares a las etapas de actualización de FC del esquema FC 200 y pueden implementarse sustancialmente por el nuevo controlador 315 en el esquema ISSU 300.

55 Siguiendo el mismo procedimiento, todos los FC 320 controlados por el antiguo controlador 310 pueden actualizarse y asociarse al nuevo controlador 315, por ejemplo de uno en uno, hasta que todos los FC 320 se actualicen y migren desde el antiguo controlador 310 al nuevo controlador 315 (fase 3 en la FIG. 3). Después de que todos los FC 320 se hayan actualizado y hayan migrado, el antiguo controlador 310 (la versión de software antigua del controlador) puede suprimirse junto con sus tablas y recursos compartidos (fase 4 en la FIG. 3). El esquema ISSU 300 puede proporcionar un procedimiento conciso o preciso para gestionar los recursos compartidos. Un recurso que puede ser fijo y que no puede liberarse puede ser una cola de entrada de paquetes, que puede estar asignada a nivel de sistema. El esquema ISSU 300 puede proporcionar la actualización para todo el plano de datos, incluidos los FC 320  
60 y otros componentes y recursos asociados, o para los FC 320 solamente en el plano de datos.

La FIG. 4 ilustra una forma de realización de un procedimiento ISSU 400 que puede implementarse en una arquitectura de reenvío de múltiples núcleos, tal como la arquitectura de reenvío de múltiples núcleos 100. Por ejemplo, el procedimiento ISSU 400 puede implementarse como parte del esquema ISSU 200 o del esquema ISSU 300 mediante un componente de reenvío. El procedimiento 400 puede comenzar en la etapa o bloque 410, donde uno o más de una pluralidad de FC que se ejecutan (en el plano de datos) en una única placa o LC pueden seleccionarse para su actualización. Por ejemplo, un controlador (por ejemplo, el controlador 210) puede seleccionar uno o más de los FC que se ejecutan en una o más CPU de la placa (por ejemplo, los FC 220) para su desactivación y actualización con una versión de software de FC nueva o actualizada. Más de un FC puede seleccionarse para su actualización casi al mismo tiempo si los recursos del sistema (por ejemplo, memoria, procesador(es), etc.) lo permiten. Un FC puede seleccionarse de manera arbitraria, en función de algunos criterios relacionados con su estado de funcionamiento, por ejemplo, un FC inactivo a ligeramente cargado, o en un orden o secuencia predeterminados de los FC de la placa.

En el bloque 420, el/los FC seleccionado(s) puede(n) desactivarse para dejar de gestionar y reenviar paquetes. Como se ha descrito anteriormente, el FC puede desactivarse de manera paulatina, donde el procesamiento de paquetes pendientes en una memoria intermedia puede completarse o los paquetes pueden transferirse a otros FC antes de que se desactive el FC, lo que puede impedir la pérdida de paquetes. En el bloque 430, una versión de software de FC nueva o actualizada puede cargarse para el/los FC seleccionado(s). En el bloque 440, nuevos recursos pueden asignarse al/a los FC actualizado(s). Los recursos pueden incluir memoria, una CPU y otros recursos de sistema para permitir el procesamiento y el reenvío de paquetes. En el bloque 450, el/los FC actualizado(s) puede(n) configurarse para unirse al procesamiento y la gestión de paquetes con los otros FC en ejecución en la placa.

En el bloque de decisión 460, el procedimiento 400 puede determinar si cualquier FC restante en la placa tiene que actualizarse. Por ejemplo, el procedimiento 400 puede determinar si todos los FC están actualizados con una nueva versión de software de FC. Si se satisface la condición en el bloque 460, es decir, hay uno o más FC restantes que tienen que actualizarse, entonces el procedimiento puede volver al bloque 410 para seleccionar uno o más de los FC restantes para su actualización. En caso contrario, el procedimiento 400 puede finalizar. En otras formas de realización, un procedimiento similar que comprende etapas similares puede implementarse para actualizar los FC en el plano de datos y el controlador en el plano de control. Este procedimiento también puede comprender etapas adicionales para añadir un nuevo controlador (por ejemplo, una versión de software nueva o actualizada del controlador) y reasignar los FC actualizados, por ejemplo, uno a uno, desde el antiguo controlador (por ejemplo, la versión de software antigua del controlador) al nuevo controlador, y después eliminar el antiguo controlador y los recursos asociados, como se describió en el esquema ISSU 300.

La FIG. 5 ilustra una forma de realización de una placa de circuito impreso 500 (denominada placa en lo sucesivo) para implementar una arquitectura de reenvío de múltiples núcleos descrita en el presente documento, tal como la arquitectura de reenvío de múltiples núcleos 100. La placa 500 puede comprender puertos de entrada 510, receptores (Rx) 512, una memoria intermedia compartida 520, una memoria 522, un controlador 524, un bus de comunicación compartido 526, una matriz de conmutación 528 que incluye núcleos de reenvío FC0, FC1, ..., FCn, transmisores (Tx) 532 y puertos de salida 530. Cada uno de los puertos de entrada puede estar acoplado a un receptor 512, como se muestra, para recibir señales y paquetes/datos desde otros componentes de red, tales como encaminadores. Los paquetes de cada uno de los receptores pueden almacenarse en una memoria intermedia compartida 520 para un procesamiento adicional. Los paquetes de la memoria intermedia 520 pueden transferirse a través del bus 526 a una matriz de conmutación 528 que comprende n+1 núcleos de reenvío, como se muestra. La memoria 522 puede almacenar software para una o más actualizaciones relativas a los núcleos de reenvío mostrados. Además, la memoria 522 puede proporcionar software y datos para configurar el funcionamiento del controlador 524. El controlador 524 puede ser un controlador tal como el controlador 210 descrito anteriormente. El controlador 524 puede comunicarse con la memoria intermedia 524 y los núcleos de reenvío FC0, FC1, ..., FCn a través del bus 526. Los núcleos de reenvío pueden transmitir paquetes a través de los transmisores 532, cada uno de los cuales puede estar acoplado a un puerto de salida 530. Como se ha indicado anteriormente, un núcleo de reenvío seleccionado para su actualización puede extraer los paquetes de la memoria intermedia 520 hasta que el núcleo de reenvío seleccionado complete su propio procesamiento de paquetes usando los paquetes necesarios e interrumpir posteriormente su procesamiento habitual en relación con la memoria intermedia compartida. Por consiguiente, no se perderán paquetes. Además, la placa 500 puede comprender un segundo controlador 525 cargado con una versión de software actualizada del software en el controlador 524 con el fin de implementar el esquema ISSU 300 descrito con respecto a la FIG. 3. El controlador 524 y el segundo controlador 525 pueden implementarse en un único procesador o pueden implementarse en procesadores diferentes.

Aunque en la FIG. 5 se muestran, con fines ilustrativos, tres puertos de entrada 510, tres receptores 512, tres transmisores 532 y tres puertos de salida 530, la placa 500 puede comprender cualquier número de estos elementos. Además, el bus 526 es opcional, ya que el controlador 524, la memoria intermedia 520 y la matriz de conmutación 528 pueden estar conectados por cable para comunicarse directamente entre sí. El controlador 524 y/o la matriz de conmutación 528 pueden implementarse a través de un procesador que puede estar implementado como uno o más chips de unidad central de procesamiento (CPU), núcleos (por ejemplo, un procesador de múltiples núcleos), matrices de puertas programables *in situ* (FPGA), circuitos integrados de aplicación específica (ASIC) y/o

procesadores de señales digitales (DSP). El controlador 524 y los núcleos de reenvío FC0, FC1, ..., FCn pueden implementarse en software en uno o más procesadores. Los componentes de red descritos anteriormente pueden implementarse en cualquier componente de red de propósito general, tal como un componente de red o de ordenador con una potencia de procesamiento, recursos de memoria y una capacidad de caudal de tráfico de red  
 5 suficientes para gestionar la carga de trabajo impuesto en el mismo. La FIG. 6 ilustra un componente de red de propósito general 600 típico y adecuado para implementar una o más formas de realización de los componentes dados a conocer en el presente documento. Por ejemplo, el componente de red 600 puede comprender o ser parte de, estar ubicado en o estar acoplado a cualquiera de un encaminador, componente de red o servidor que comprenda una arquitectura de reenvío de múltiples núcleos, tal como la arquitectura de reenvío de múltiples  
 10 núcleos 100. El componente de red 600 incluye un procesador 602 (que puede denominarse CPU) que está en comunicación con dispositivos de memoria, incluidos un almacenamiento secundario 604, una memoria de solo lectura (ROM) 606, una memoria de acceso aleatorio (RAM) 608, dispositivos de entrada/salida (E/S) 610 y dispositivos de conectividad en red 612. El procesador 602 puede estar implementado como uno o más chips de CPU, núcleos (por ejemplo, un procesador de múltiples núcleos), FPGA, ASIC y/o DSP, y/o puede ser parte de uno  
 15 o más ASIC. El procesador 602 puede estar configurado para implementar cualquiera de los esquemas y procedimientos ISSU descritos anteriormente. Por ejemplo, el procesador 602 puede estar configurado (a través de hardware y/o software) para implementar o admitir los esquemas 200 y/o 300, así como el procedimiento 400.

El almacenamiento secundario 604 está comprendido normalmente por una o más unidades de disco o una ROM programable y borrable (EPROM) y se usa para el almacenamiento no volátil de datos. El almacenamiento secundario 604 puede usarse para almacenar programas que se cargan en la RAM 608 cuando tales programas se seleccionan para su ejecución. La ROM 606 se usa para almacenar instrucciones, y quizá datos, que se leen durante la ejecución de programas. La ROM 606 es un dispositivo de memoria no volátil que tiene normalmente una  
 20 pequeña capacidad de memoria en comparación con la mayor capacidad de memoria del almacenamiento secundario 604. La RAM 608 se usa para almacenar datos volátiles y quizá para almacenar instrucciones. El acceso tanto a la ROM 606 como a la RAM 608 es normalmente más rápido que el acceso al almacenamiento secundario 604.

Debe entenderse que al programar y/o cargar instrucciones ejecutables en la placa 500, al menos uno de la memoria 522, el controlador 524 y la matriz de conmutación 528 que comprende núcleos de reenvío FC0, FC1, ..., FCn se modifican, transformando la placa 500, en parte, en una máquina o aparato particular, por ejemplo una  
 30 arquitectura de reenvío de múltiples núcleos, que cuenta con la funcionalidad novedosa descrita por la presente divulgación. Asimismo, debe entenderse que al programar y/o cargar instrucciones ejecutables en el componente de red 600, al menos uno del procesador 602, la ROM 606 y la RAM 608 se modifican, transformando el componente  
 35 de red 600, en parte, en una máquina o aparato particular, por ejemplo una arquitectura de reenvío de múltiples núcleos, que cuenta con la funcionalidad novedosa descrita por la presente divulgación. Es fundamental tanto para la ingeniería eléctrica como para la ingeniería del software, que la funcionalidad que puede implementarse cargando software ejecutable en un ordenador pueda convertirse en una implementación de hardware mediante reglas de diseño ampliamente conocidas. Las decisiones entre implementar un concepto en software o en hardware se basan  
 40 normalmente en consideraciones de estabilidad del diseño y en el número de unidades a producir, en lugar de en cuestiones relacionadas con conversiones desde el dominio de software al dominio de hardware. Por lo general, puede preferirse que un diseño que sigue sujeto a un cambio frecuente se implemente en software, ya que modificar una implementación en hardware es más caro que modificar un diseño de software. Por lo general, puede preferirse que un diseño que sea estable y que se producirá en grandes cantidades se implemente en hardware, por ejemplo  
 45 en un circuito integrado de aplicación específica (ASIC), ya que para una producción en grandes series, la implementación en hardware puede ser menos cara que la implementación en software. Con frecuencia, un diseño puede desarrollarse y probarse en forma de software y transformarse posteriormente, mediante reglas de diseño ampliamente conocidas, a una implementación en hardware equivalente en un circuito integrado de aplicación específica que cablea las instrucciones del software. De la misma manera que una máquina controlada por un nuevo  
 50 ASIC es una máquina o aparato particular, asimismo, un ordenador que se ha programado y/o cargado con instrucciones ejecutables puede considerarse como una máquina o aparato particular.

Al menos se da a conocer una forma de realización, y variaciones, combinaciones y/o modificaciones de la(s) forma(s) de realización y/o de las características de la(s) forma(s) de realización realizadas por expertos en la  
 55 técnica están dentro del alcance de la divulgación. Formas de realización alternativas que resultan de combinar, integrar y/u omitir características de la(s) forma(s) de realización también están dentro del alcance de la divulgación. Cuando haya intervalos o límites numéricos indicados de manera expresa, debe entenderse que tales intervalos o límites expresos incluyen intervalos o límites iterativos de magnitud similar que están dentro de los intervalos o límites indicados de manera expresa (por ejemplo, de 1 aproximadamente a 10 aproximadamente incluye 2, 3, 4, etc.; un valor superior a 0,10 incluye 0,11, 0,12, 0,13, etc.). Por ejemplo, si se da a conocer un intervalo numérico con un límite inferior,  $R_l$ , y un límite superior,  $R_u$ , cualquier número que esté dentro del intervalo se da a conocer de manera específica. En particular, se dan a conocer específicamente los siguientes números dentro del intervalo:  $R = R_l + k * (R_u - R_l)$ , donde  $k$  es una variable que oscila entre el 1 por ciento y 100 por ciento con un incremento del 1  
 60 por ciento, es decir,  $k$  es un 1 por ciento, un 2 por ciento, un 3 por ciento, un 4 por ciento, un 5 por ciento, ..., un 50 por ciento, un 51 por ciento, un 52 por ciento, ..., un 95 por ciento, un 96 por ciento, un 97 por ciento, un 98 por ciento, un 99 por ciento o un 100 por ciento. Además, cualquier intervalo numérico definido por dos números  $R$  como  
 65

se ha definido anteriormente también se da a conocer de manera específica. El uso del término "aproximadamente" significa un  $\pm 10\%$  del número subsiguiente, a no ser que se indique lo contrario. El uso del término "opcionalmente" con respecto a cualquier elemento de una reivindicación significa que se requiere el elemento o, como alternativa, no se requiere el elemento, estando ambas alternativas dentro del alcance de la reivindicación. Debe entenderse que el uso de términos más genéricos tales como "comprende", "incluye" y "que presenta", engloba el uso de términos más específicos tales como "que consiste en", "que consiste esencialmente en" y "comprendido sustancialmente por". Por consiguiente, el alcance de protección no está limitado por la descripción expuesta anteriormente, sino que está definido por las reivindicaciones siguientes, donde el alcance incluye todas las equivalencias del contenido de las reivindicaciones. Cada una de las reivindicaciones se incorpora como una divulgación adicional en la memoria descriptiva, y las reivindicaciones son formas de realización de la presente divulgación. El análisis de una referencia en la divulgación no implica que se refiera a la técnica anterior, especialmente cualquier referencia que tenga una fecha de publicación posterior a la fecha de prioridad de esta solicitud.

Los presentes ejemplos deben considerarse ilustrativos y no restrictivos, y la intención es que no deben limitarse a los detalles expuestos en el presente documento. Por ejemplo, los diversos elementos o componentes pueden combinarse o integrarse en otro sistema, o ciertas características pueden omitirse o no implementarse.

Además, las técnicas, sistemas, subsistemas y procedimientos descritos e ilustrados en las diversas formas de realización como discretos o separados pueden combinarse o integrarse con otros sistemas, módulos, técnicas o procedimientos. Otros elementos mostrados o descritos como acoplados o directamente acoplados o que se comunican entre sí, pueden estar indirectamente acoplados o comunicarse a través de alguna interfaz, dispositivo o componente intermedio, ya sea de manera eléctrica, mecánica o de otro modo. Los expertos en la técnica pueden concebir y realizar otros ejemplos de cambios, sustituciones y alteraciones.



**REIVINDICACIONES**

1. Un procedimiento implementado en un componente de red (600) para una actualización de software en servicio, ISSU, comprendiendo el procedimiento:

5           seleccionar un núcleo de reenvío, FC, a partir de una pluralidad de FC (220) que se ejecutan en un plano de datos en una placa de circuito impreso (500) del componente de red para procesar y reenviar una pluralidad de paquetes; y  
 reajustar el FC seleccionado con una actualización de software,  
 10           en el que al menos uno de los otros FC procesa y reenvía paquetes cuando el FC seleccionado se actualiza, caracterizado por que el procedimiento comprende:

                  transmitir un comando de reajuste al FC seleccionado para dejar de hacer al menos una de las siguientes acciones:

15                     extraer paquetes desde una memoria intermedia compartida (520) mediante el FC seleccionado y responder a una interrupción de entrada/salida, ES, para el FC seleccionado antes de que el FC seleccionado se reajuste; y  
 transferir, usando el componente de red, los paquetes asignados para el FC seleccionado desde la memoria intermedia compartida al al menos uno de los otros FC en ejecución para su procesamiento y reenvío antes de que se reajuste el FC seleccionado, donde los paquetes transferidos se ponen en cola en una memoria intermedia de alta prioridad a la que se accede mediante el al menos uno de los otros FC en ejecución para su procesamiento y reenvío.

25           2. El procedimiento según la reivindicación 1, que comprende además seleccionar secuencialmente otros FC para su actualización a partir de la pluralidad de FC hasta que no haya FC restantes que tengan que actualizarse.

3. El procedimiento según la reivindicación 2, en el que solo se selecciona a la vez un FC para su actualización mientras que los otros FC están procesando y reenviando los paquetes.

30           4. El procedimiento según la reivindicación 3, en el que más de uno de los FC se selecciona para actualizarse casi al mismo tiempo mientras que los otros FC están procesando y reenviando los paquetes si hay suficientes recursos de sistema para permitir la actualización de múltiples FC casi al mismo tiempo.

35           5. El procedimiento según la reivindicación 1, en el que el FC se selecciona para su actualización en función del estado de carga mediante un controlador (210) que se ejecuta en un plano de control en la placa de circuito impreso.

6. El procedimiento según la reivindicación 2, en el que los FC se seleccionan para su actualización según una secuencia predeterminada de los FC.

40           7. El procedimiento según la reivindicación 1, en el que los FC son componentes de software que se ejecutan en uno o más procesadores en la placa de circuito impreso.

45           8. El procedimiento según la reivindicación 2, en el que los FC se seleccionan y reajustan para su actualización mediante un software de controlador que se ejecuta en un procesador en la placa de circuito impreso.

9. Un componente de red (600), que comprende:

50           una placa de circuito impreso (500) que comprende una pluralidad de núcleos de reenvío, FC (220), donde cada uno de los FC está configurado para reenviar paquetes; y un procesador acoplado a la pluralidad de FC y configurado para:

                  seleccionar (410) un FC a partir de la pluralidad de FC;  
 desactivar (420) el FC seleccionado para dejar de reenviar paquetes;  
 cargar (430) una versión de software de FC actualizada para el FC seleccionado;

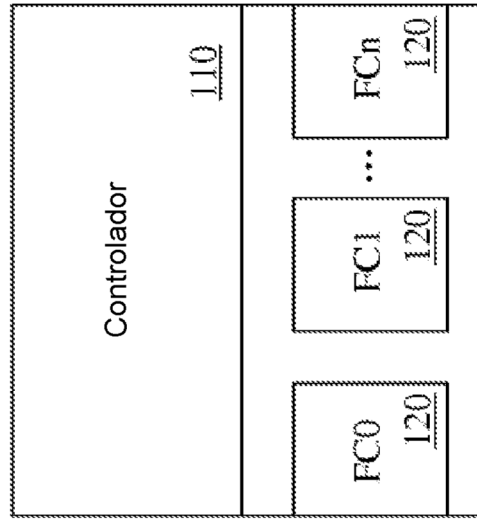
55           asignar (440) nuevos recursos al FC seleccionado;  
 configurar (450) el FC seleccionado para que se una al reenvío de paquetes con los otros FC en ejecución en la placa de circuito impreso; y

                  activar el FC seleccionado tras configurar el FC seleccionado,

60           donde al menos uno de los FC está configurado para reenviar paquetes mientras que el FC seleccionado está desactivado, caracterizado por que

- 5 el FC seleccionado deja de extraer paquetes desde una memoria intermedia compartida (520) o de responder a una interrupción de entrada/salida, ES, para los FC antes de desactivarse, donde el FC seleccionado transfiere sus paquetes desde una memoria intermedia compartida a los otros FC en ejecución para su gestión y reenvío antes de desactivarse, y donde los paquetes trasferidos se ponen en cola en una memoria intermedia de alta prioridad a la que se accede mediante los otros FC en ejecución para su gestión y reenvío.
- 10 10. El componente de red según la reivindicación 9, en el que más de uno de los FC se actualizan casi al mismo tiempo si hay suficientes recursos disponibles que lo permitan.
11. El componente de red según la reivindicación 10, en el que el más de uno de los FC se actualizan casi al mismo tiempo de manera independiente y sin sincronización.
- 15 12. El componente de red según la reivindicación 9, en el que todos los FC se actualizan seleccionando y desactivando uno o más de los FC a la vez sin desactivar todos los FC.
- 20 13. El componente de red según la reivindicación 9, en el que el FC seleccionado sigue extrayendo paquetes de una memoria intermedia compartida o respondiendo a una interrupción de ES para que los FC completen cualquier procesamiento de paquetes necesario antes de desconectarse.
- 25 14. El componente de red según la reivindicación 9, que comprende además una memoria intermedia acoplada a cada uno de la pluralidad de FC, donde la memoria intermedia está configurada para almacenar una pluralidad de paquetes para reenviarse mediante el FC seleccionado, y donde cualquier paquete destinado al FC seleccionado que estaba en la memoria intermedia en el momento en que se desactiva el FC seleccionado permanece en la memoria intermedia hasta que se active el FC seleccionado.

100 →



Plano de control

Plano de datos

FIG. 1

200 ↗

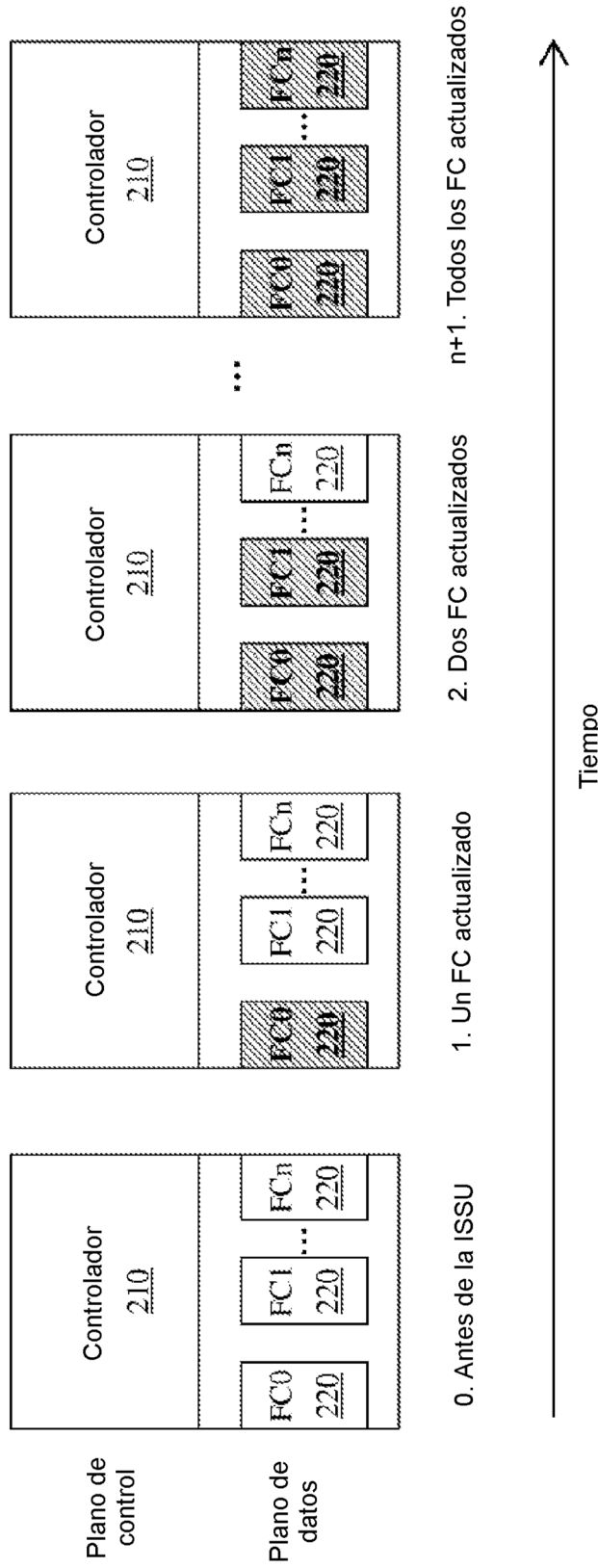


FIG. 2

300 ↗

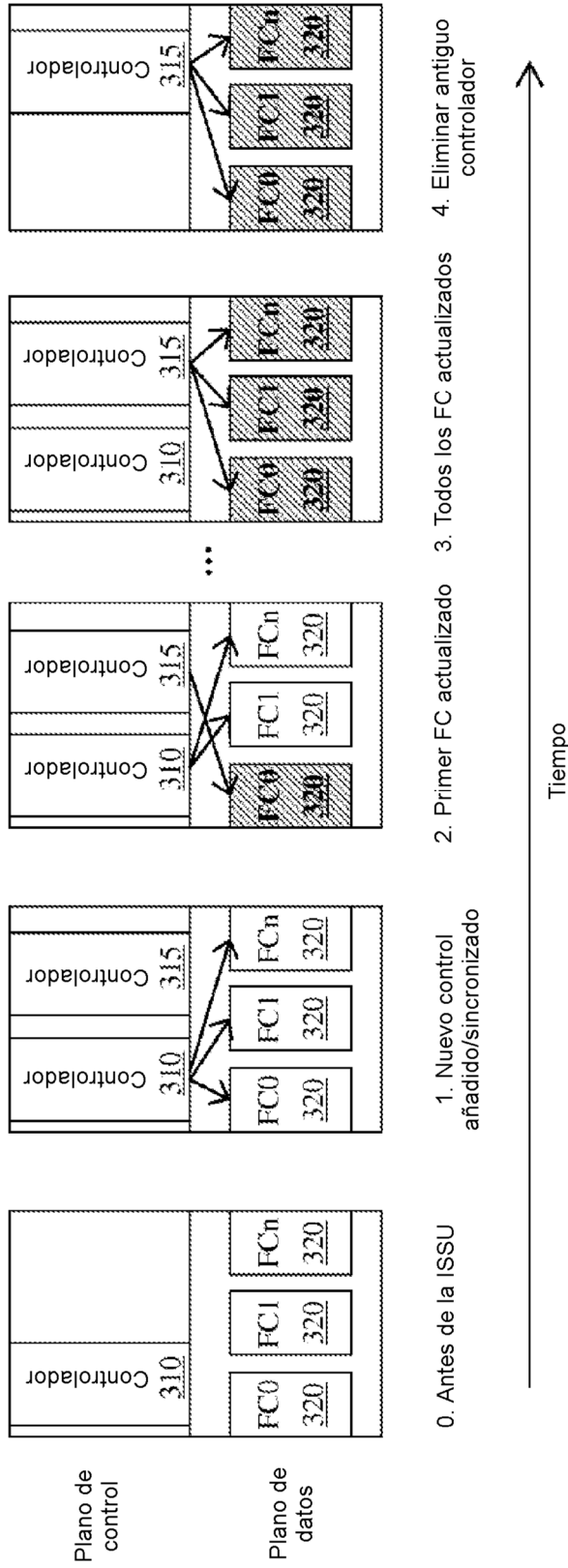


FIG. 3

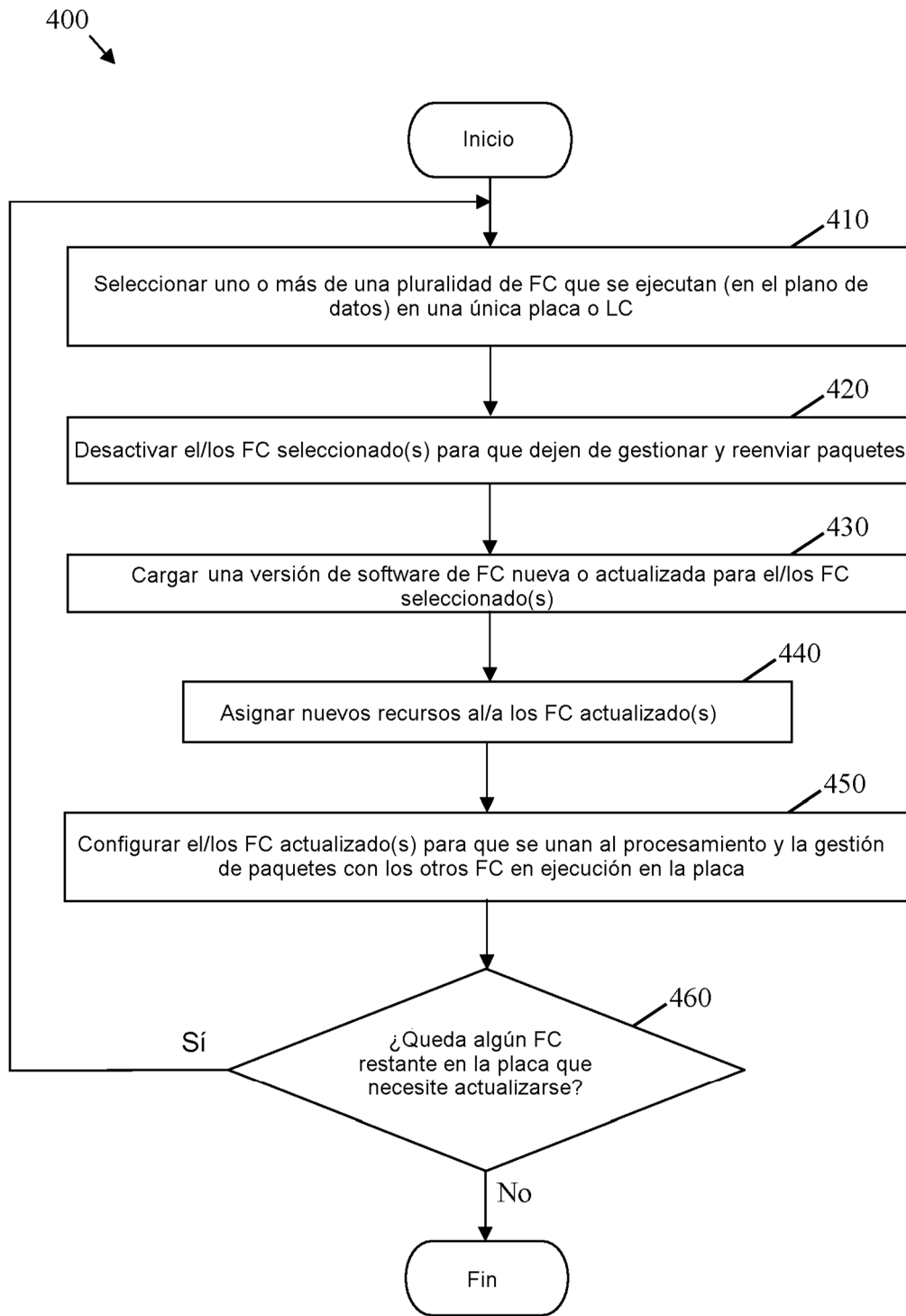


FIG. 4

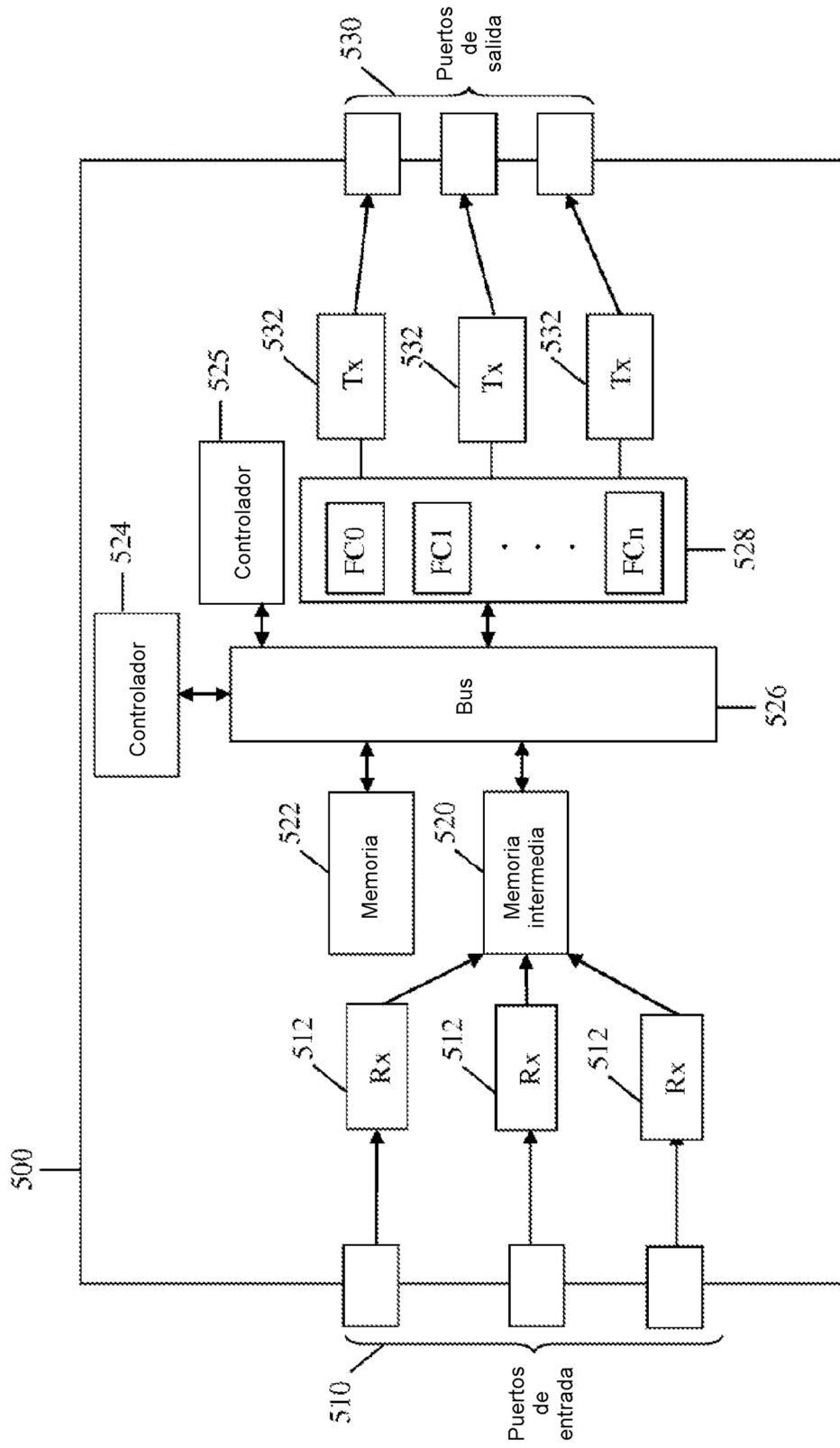


FIG. 5

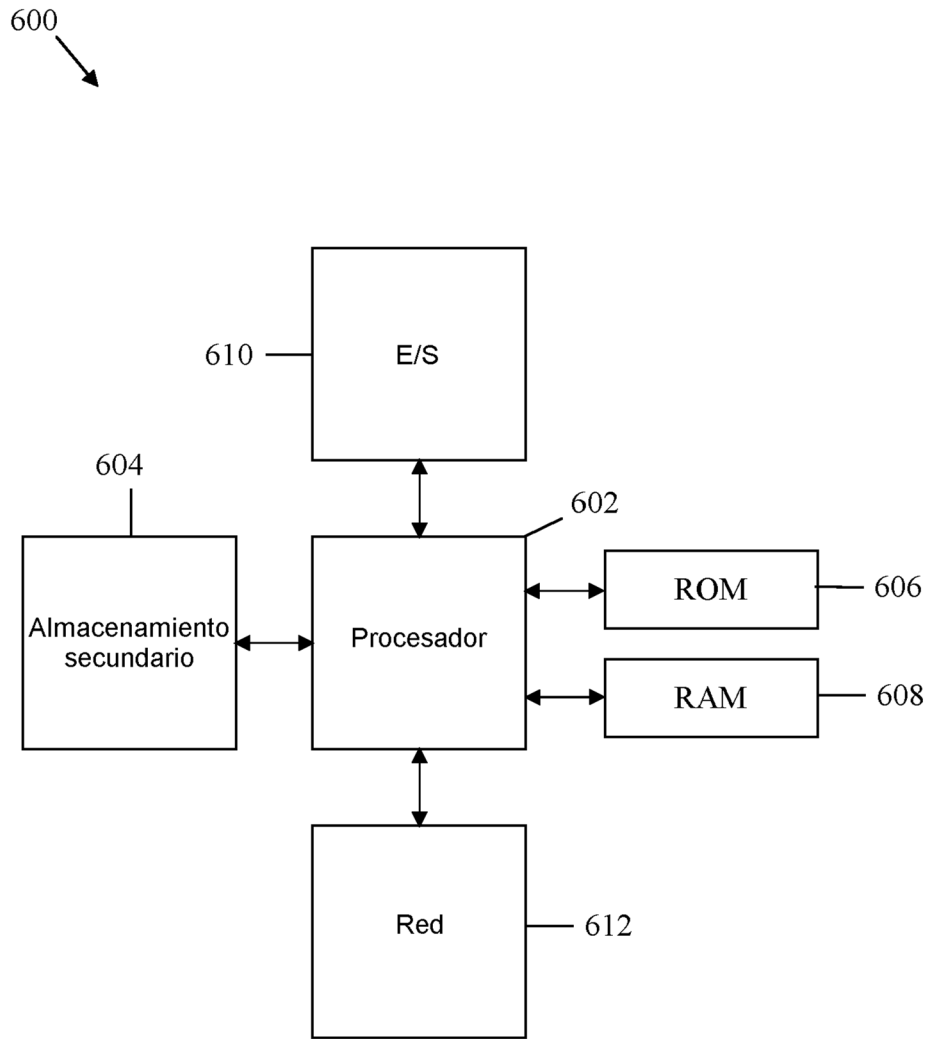


FIG. 6