

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 848 846**

51 Int. Cl.:

**H04L 12/861** (2013.01)

**H04W 72/12** (2009.01)

**H04L 12/54** (2013.01)

**G06F 9/50** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **19.10.2011 PCT/US2011/056829**

87 Fecha y número de publicación internacional: **18.05.2012 WO12064471**

96 Fecha de presentación y número de la solicitud europea: **19.10.2011 E 11785167 (5)**

97 Fecha y número de publicación de la concesión europea: **02.12.2020 EP 2638467**

54 Título: **Esquema de mensajería de no bloqueo o copia cero para aplicaciones en redes de telecomunicaciones**

30 Prioridad:

**12.11.2010 US 945146**

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

**12.08.2021**

73 Titular/es:

**ALCATEL LUCENT (100.0%)  
Site Nokia Paris Saclay, Route de Villejust  
91620 Nozay, FR**

72 Inventor/es:

**KHAWER, MOHAMMAD R. y  
SO, LINA**

74 Agente/Representante:

**ISERN JARA, Jorge**

ES 2 848 846 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

Esquema de mensajería de no bloqueo o copia cero para aplicaciones en redes de telecomunicaciones

## 5 Antecedentes de la invención

10 Las realizaciones a modo de ejemplo de la invención se refieren a un esquema de mensajería de copia cero, de no bloqueo, y sin bloqueo para soporte de múltiples celdas en un tablero moderno individual usando un procesador multi-núcleo. Mientras que la invención se refiere particularmente a la técnica de telecomunicaciones inalámbricas, y se describirá, además, con referencia específica a éstas, se apreciará que la invención puede tener utilidad en otros campos y otras aplicaciones.

15 A modo de antecedente, LTE (Evolución a Largo Plazo) es un proyecto 3GPP que evoluciona rápido que se orienta a mejorar el estándar de telefonía móvil UMTS (Sistema Universal de Telefonía Móvil) para abarcar las demandas futuras en las redes de comunicaciones. LTE mejora la eficiencia y el ancho de banda de las redes inalámbricas, reduce los costos y mejora la experiencia del servicio. Específicamente, la LTE hace uso de nuevas oportunidades de espectro y ofrece mejor integración con otros estándares abiertos. La LTE generalmente incluye RAN LTE (Red de Acceso por Radio) (también conocida como EUTRAN) junto con un EPS (Sistema de Paquete Evolucionado, también llamado Núcleo de Paquete Evolucionado).

20 Los sistemas de comunicaciones generalmente se dividen en dos funciones primarias: las funciones del plano de datos y las funciones del plano de control. En los productos de LTE anteriores, se usan al menos dos procesadores en el tablero moderno: uno para soportar las funciones del plano de control (tiempo no real, por ej., Operaciones, Administración y Manejo (u OA&M), y funcionalidades relacionadas con el manejo de procesamiento de llamadas), y otro para terminar y soportar las funciones del plano de datos (tiempo real, por ej., procesamiento de la capa de LTE 2). Tanto los planos de control como de datos utilizan diferentes instancias del sistema operativo (OS), tales como Linux para el plano de control y un OS en tiempo real tal como vXWorks (fabricado y vendido por Wind River Systems de Alameda, California) para el núcleo en plano de datos. Normalmente, un tablero moderno soporta un sector o celda. Así para soportar configuraciones de múltiples celdas (por ej., 3 celdas o 6 celdas) sería necesario proporcionar tantos tableros modernos como número de celdas.

35 Un procesador multi-núcleo puede usarse en una estación base LTE inalámbrica en un tablero moderno. Un sistema operativo tal como SMP Linux con un parche PREEMPT\_RT puede ejecutarse en una partición SMP que contiene los ocho núcleos. En esta configuración el plano de control (hilos/procesos en tiempo no real) y el plano de datos (hilos/procesos en tiempo real) comparten las mismas instancias del sistema operativo aunque se unen para ejecutarse en diferentes núcleos. El uso del paquete de protocolos Linux para enviar/recibir paquetes TCP/UDP de los procesos/hilos en tiempo real, tales como el planificador LTE L2 ejecutado en el núcleo de plano de datos, puede producir picos de latencia ilimitados que pueden discontinuar el procesamiento del planificador LTE L2. Esto se debe al hecho de que el paquete de protocolos Linux no está diseñado para ser sin bloqueo, y dado que también es usado por los procesos/hilos en tiempo no real tales como OAM, y CALLP ejecutados en el núcleo del plano de control.

45 En un entorno SMP, un bloqueo tomado por un proceso o hilo en tiempo no real en un núcleo (tal como se encuentra en el plano de control) puede hacer que el pico de latencia para un hilo o proceso en tiempo real espere que el bloqueo se libere en un núcleo diferente (tal como se encuentra en el plano de datos). De este modo, existe una necesidad de lograr un esquema de mensajería de no bloqueo, copia cero, sin bloqueo que no use el paquete de Protocolos Linux para la trayectoria de datos rápidos en el núcleo del plano de datos para satisfacer las necesidades de rendimiento del tablero moderno basado en un procesador multi-núcleo que usa una partición. Sin dicho esquema, el sistema puede estar sujeto a picos de latencia ilimitados que podrían descomponer el sistema entero, especialmente con respecto al procesamiento del planificador LTE L2.

50 El documento US 2008/276056 A1 divulga un sistema de comunicaciones de poner en cola y eliminar de cola de punto a punto que usa un método de gestión de buffer para un procesador multi-núcleo en el que un buffer se adquiere en la memoria compartida entre el núcleo y el espacio de usuario. El buffer puede contener una cola de marco que puede guardar los datos recibidos de la red y al que se accede usando dos punteros - un puntero delantero para apuntar a la ubicación de la memoria más vieja donde los datos están escritos, y un puntero de cola para apuntar a la ubicación de la memoria más vieja donde los datos se leyeron.

60 El documento WO 2005/098623 A2 divulga un planificador basado en requisitos previos que tiene en cuenta los requisitos previos de recursos del sistema para ejecución.

## Sumario de la invención

La invención se define en las reivindicaciones adjuntas. En la siguiente descripción, el término "realización" debe comprenderse como que significa un "ejemplo" siempre que se refiere a materia no cubierta por las reivindicaciones.

- 5 En un ejemplo, se proporciona un método implementado por ordenador para mensajería con copia de datos cero, de no bloqueo en un procesador multi-núcleo. El método incluye, para cada uno de una pluralidad de núcleos, adquirir un núcleo para el buffer de mapeo de usuario-espacio (K-U) y el correspondiente descriptor del buffer, insertar un paquete de datos en el buffer; e insertar el descriptor del buffer en un buffer circular.
- 10 En un ejemplo, se proporciona un aparato para proporcionar acceso de copia cero, de no bloqueo a buffers entre procesos usuario-espacio y procesos de espacio del núcleo. El aparato incluye un tablero moderno, un procesador multi-núcleo que tiene una pluralidad de núcleos de procesador unidos al tablero moderno. El tablero moderno incluye el espacio del núcleo, espacio de usuario y un buffer circular para cada núcleo del procesador adaptado para almacenar el espacio del núcleo a descriptores del buffer mapeado en espacio de usuario (K-U).
- 15 En un ejemplo, se proporciona un portador de datos permanentes usable por computadora que almacena instrucciones que hacen que una computadora adquiera, para cada uno de una pluralidad de núcleos de procesamiento, un buffer circular para almacenar descriptores de buffer. Los descriptores del buffer almacenados en un buffer circular contienen información sobre los buffers que se mapean del espacio de núcleo al espacio de usuario (K-U). Las instrucciones también hacen que el ordenador realice lo siguiente usando uno o más hilos del espacio del núcleo en conjunción con uno o más componentes de hardware: recibir un paquete de datos en una interfaz de Ethernet, adquirir un buffer mapeado K-U y el puntero correspondiente, insertar el paquete de datos recibido en un buffer mapeado K-U, poner en cola un descriptor de marco que comprende el puntero del buffer mapeado K-U que contiene el paquete de datos en una cola de marco, y después de que el paquete de datos llega delante de la cola de datos, generar una señal de interrupción que invoca un proceso que crea un descriptor de buffer del descriptor de marco. El descriptor de buffer asociado con el paquete de datos se inserta luego en el buffer circular especificado de acuerdo con la cola del marco para el mapeo del buffer circular.
- 20
- 25

30 El alcance adicional de aplicabilidad de la presente realización a modo de ejemplo resultará evidente a partir de la descripción detallada proporcionada a continuación. Debe entenderse, sin embargo, que la descripción detallada y los ejemplos preferidos, mientras que indican formas de realización específicas de la invención, se dan únicamente a modo ilustrativo, ya que varios cambios y varias modificaciones dentro del alcance de la invención resultarán evidentes para aquellas personas con experiencia en la técnica.

## DESCRIPCIÓN DE LAS FIGURAS

- 35 La presente invención existe en la construcción, la gestión y la combinación de las varias partes del dispositivo, y los pasos del método, por lo cual los objetivos contemplados se logran como se establece más completamente de aquí en adelante, como se menciona específicamente en las reivindicaciones, y se ilustra en los dibujos que se incluyen, en los que:
- 40 La FIG. 1 ilustra una realización de una arquitectura de plataforma de acuerdo con los aspectos de la presente invención; La FIG. 2 ilustra una arquitectura a modo de ejemplo para implementar la capa de abstracción de núcleo que se muestra en la FIG. 1; La FIG. 3 ilustra un sistema basado en BED DPAA para proporcionar el flujo de datos de ingreso de los paquetes de datos; La FIG. 4 representa una descripción de alto nivel del flujo de proceso con respecto a un paquete de datos entrantes;
- 45 La FIG. 5 ilustra un sistema basado en BED DPAA para proporcionar el flujo de datos de egreso de los paquetes de datos; y La FIG. 6 representa una descripción de alto nivel del flujo de proceso con respecto a un paquete de datos salientes (de egreso).

## DESCRIPCIÓN DETALLADA

50 Ahora con referencia a los dibujos en los que lo que se muestra es para fines ilustrativos de las reivindicaciones solamente y no para fines de limitación del asunto reivindicado, la FIG. 1 proporciona una vista de un sistema en el que pueden incorporarse las realizaciones actualmente descritas.

55 Con referencia ahora a la FIG. 1, se muestra una arquitectura de plataforma 100 a modo de ejemplo. La arquitectura generalmente se usa en un tablero moderno, pero debe comprenderse que puede usarse en otras aplicaciones. En esta realización, una partición se define con los ocho núcleos presentes en ella. Se apreciará, sin embargo, que el procesador multi-núcleo 100 puede tener cualquier número de núcleos. Con esta realización, es posible de este modo usar una instancia del sistema operativo (OS) multiprocesamiento simétrico (SMP) individual 102 que corre en la totalidad de los

60

núcleos (por ej., ocho núcleos). Dado que los planes de control y datos están bajo una instancia del OS, generalmente se requiere cuidado para asegurar que un problema con los datos de control no degrade también el plano de control.

5 En este ejemplo, el procesador multi-núcleo 100 sirve a tres celdas (mostradas como 104, 106 y 108 en la figura). Cada celda requiere un planificador de enlace ascendente (UL) (mostrado como 110, 112, y 114 en la figura) y un planificador de enlace descendente (DL) (mostrado como 116, 118, y 120 en la FIG. 1).

10 Se sabe que la capa de Control de Enlace de Radio (RLC) se usa para segmentar, concatenar y corregir errores en marcos de paquetes enviados y recibidos a través de la interfaz de aire LTE. El software de Control de Enlace de Radio y el Control de Acceso a Medios (RLC/MAC) se usa en el grupo inalámbrico GPRS (2.5G). Proporciona transferencia de datos reconocidos y no reconocidos entre la estación móvil y el controlador de la estación base (BSC). De este modo, también incluye un bloque de RLC/MAC 122, que es la unidad de transporte básico en la interfaz de aire que se usa entre el móvil y la red. El bloque RLC/MAC 122 se usa para transportar datos y señalización de RLC/MAC.

15 El procesador multi-núcleo 100 también proporciona Operaciones, Administración y Gestión (OA&M) 124 y un módulo CALLLP 126 que proporciona servicios de gestión de llamados en tiempo no real.

20 Además, el procesador multi-núcleo 100 incluye una capa de abstracción (CAL) 128, que esconde los detalles específicos del núcleo del software de aplicación de la Capa 2 (L2). La Capa 2 es la Capa de Enlace de Datos del modelo OSI de siete capas de las redes informáticas. La Capa de Enlace de Datos es la capa del protocolo que transfiere datos entre nodos adyacentes de la red en una red de área amplia o entre nodos en el mismo segmento de la red de área local. La Capa de Enlace de Datos proporciona medios funcionales y de procedimientos para transferir datos entre entidades de la red y podría proporcionar los medios para detectar y posiblemente corregir errores que pueden ocurrir en la Capa Física. Los ejemplos de protocolos de enlaces de datos son Ethernet para redes de área local (multi-nodo), el Protocolo Punto a Punto (PPP), HDLC y ADCCP para conexiones punto a punto (nodo dual). En este caso, L2 generalmente se refiere al procesamiento del planificador L2 que se necesita para la interfaz de aire LTE, que tiene requerimientos en tiempo real muy ajustados.

30 Para cumplir con las necesidades de rendimiento en tiempo real de la estación base, que es responsable de manejar el tráfico y la señalización entre el dispositivo de comunicación móvil y el sub-sistema de conmutación de red, se puede usar un OS tal como SMP Linux con parche PREEMPT\_RT. Por supuesto, se comprenderá que pueden usarse otros sistemas operativos. Se elige un OS de fuente abierta tal como SMP Linux con Preempt\_RT sobre un OS en tiempo real de núcleo de avanzada tal como vxWorks dado que vxWorks tiene aranceles de licencia muy elevados para su uso. Para lograr el comportamiento determinista en dicha configuración de SMP, el sistema se implementa preferentemente de una manera que emplea construcciones de reserva del núcleo y afinidad del núcleo para lograr un comportamiento de sistema AMP. Esto también es deseable para lograr el mejor rendimiento de SMP Linux con OS PREEMPT\_RO, por ejemplo. El uso de servicios de copia cero de no bloqueo, tales como servicios de gestión del buffer y mensajería, también puede ayudar a abordar los temas de latencia que pueden aparecer por el uso de SMP Linux con OS PREEMPT\_RT.

40 Una de las funciones principales de la capa de abstracción del núcleo (128) como se muestra en la FIG. 1 es proporcionar aplicaciones del más alto nivel tales como procesamiento L2, con diversos servicios que utilizan capacidades completas de la plataforma multi-núcleo. La capa de abstracción del núcleo está diseñada de este modo para lograr varios objetivos. Primero, debe soportar una Interfaz BED (Unidad de Ethernet de Plano Posterior) basada en DPAA, mientras que esconde la implementación de DPAA y específica de multi-núcleo a partir de un software de aplicación a nivel superior (es decir, software L2). Segundo, debe utilizar componentes de hardware DPAA P4080 para proporcionar una trayectoria de datos acelerados para datos en el plano de usuario en ambas direcciones de ingreso y egreso. Tercero, debe proporcionar tanta flexibilidad como sea posible para adaptarse fácilmente a los cambios en configuración (es decir, sin requerir cambios en el código). Un ejemplo de la configuración CAL es una configuración de recursos DPAA para grupos de buffer, colas del margo de ingreso, y colas del marco de egreso.

50 Para resumir, una implementación de la presente invención usa la totalidad de los núcleos del procesador en una partición. Esto requerirá el uso de solamente una instancia del OS, y no necesita usar un hipervisor. Dado que puede ser difícil para el Linux simple cumplir con todas las necesidades de procesamiento en tiempo real, se prefiere un OS tal como SMP Linux con parche PREEMPT\_RT. Un OS de recurso abierto también se usa para reducir el costo. El sistema incorpora además capacidades de afinidad del núcleo y reserva de la CPU para SMP Linux para definir un comportamiento del sistema similar a AMP dentro de la configuración SMP, lo que permite configuraciones de 6 celdas o incluso 9 celdas. Debido a que la instancia del OS se comparte entre núcleos en tiempo no real (tales como el plano de control) y núcleos en tiempo real (tales como los planos de datos), surgen los problemas cuando se toma un bloqueo de parte de un hilo/proceso en tiempo no real que causa una demora para un hilo/proceso en tiempo real ya que el hilo/proceso en tiempo real tiene que esperar para la liberación del bloqueo para el(los) núcleos en el plano de datos. Dado que el paquete de protocolos Linux no garantiza una implementación sin bloqueos, esta invención define un esquema de mensajería de no bloqueo para un proceso en tiempo real (planificador LTE L2) que se ejecuta en núcleos en el plano de datos, para enviar/recibir paquetes

de IP TCP/UDP mientras que se evita el uso del paquete de protocolos. El proceso en tiempo no real, tal como la ejecución de OA&M en el núcleo del plano de control, continuará usando el paquete de protocolos Linux para su operación normal. Debido a temas de Linux GPL nuestro planificador LTE L2 de marca registrada es operado por el espacio de usuario. Para enviar y recibir datos de IP de TCP/UDP del planificador LTE L2, los datos tienen que cruzar el límite de espacio usuario-núcleo que normalmente requiere una copia de datos. De este modo, la potencia de consumo del procesador para copiar los datos de una ubicación de memoria a otra gasta recursos valiosos. De esta manera, es aconsejable proporcionar un método y un sistema que permite un servicio de mensajería eficiente sin bloqueo, de copia cero y no bloqueo para los hilos/procesos en tiempo real que se ejecutan en los núcleos en el plano de datos mientras permiten que el plano de control opere en su manera normal (tal como mediante el uso del paquete de protocolos Linux tradicional).

Con referencia ahora a la FIG. 2, se muestra una arquitectura 200 a modo de ejemplo que logra estos y otros objetivos. Se sabe que un sistema operativo de ordenador convencional usualmente separa memoria virtual en el espacio del núcleo y el espacio de usuario. El espacio del núcleo se reserva estrictamente para ejecutar el núcleo, extensiones del núcleo y la mayoría de las unidades del dispositivo. En contraste, el espacio de usuario es un área de memoria en la que todas las aplicaciones en modo usuario trabajan y esta memoria puede intercambiarse cuando sea necesario. En este aspecto, una capa de abstracción del núcleo (CAL) 201 incluye varios módulos en espacio de usuario, incluyendo, pero sin limitación: un módulo de inicialización de la capa de abstracción (CALInit) 202 que carga la configuración de la red LTE y cualquier regla de PCD estática a los administradores de marco (FMAN) 230 y 232 y configura el marco CAL en base a un conjunto de archivos de configuración, un módulo del buffer de la capa de abstracción del núcleo (CALBuf) 204, un módulo de mensajería de la capa de abstracción del núcleo (CALMsg) 206 que proporciona servicios de mensajería al software de L2 para enviar y recibir datos del plano de usuario a o de otro tablero (es decir, eCCM) sin el uso del paquete de protocolos Linux, un análisis de la capa de abstracción del núcleo, módulo de reclasificación y distribución (CALPcdFmc) 208 que proporciona reglas de Análisis, Clasificación y Distribución (PDC) y configuraciones que serán usadas por cada FMAN (230, 232) para marcos de ingreso a núcleos apropiados, y un rastro DPAA de la capa de abstracción del núcleo (CALDpaaTrace) 210 que proporciona capacidades de rastreo para habilitar y deshabilitar rastros en la unidad DPAA de la capa de abstracción del núcleo (CALDPAADriver) 212, que es un módulo de espacio del núcleo.

La arquitectura 200 incluye además un sistema operativo adecuado 214 tal como SMP Linux con parche Preempt RT. El sistema operativo 214, a su vez, soporta varias unidades, tales como la mencionada anteriormente CALDPAADriver 212, al menos una unidad administradora de marco (FMAN) 216, al menos una unidad administradora del buffer (BMAN) 218, y al menos una unidad administradora de cola (QMAN) 220.

Según lo mostrado en la FIG. 2, la arquitectura 200 puede incluir adecuadamente una tela P4080 CoreNet 222, que es una arquitectura de interconexión adecuada para que la red en chip susceptible de actualizarse se conecte a múltiples núcleos de procesamiento de arquitectura con cachés, cachés individuales y subsistemas de memoria.

El procesador P4080 incluye una implementación de la nueva Arquitectura de Aceleración de la Trayectoria de Datos (DPAA). De este modo, la arquitectura 200 puede incluir, además una DPAA P4080 224. La DPAA 224 está diseñada para optimizar el procesamiento de red multi-núcleo tal como la dispersión de la carga y compartimiento de recurso, incluyendo interfaces de red y aceleradores de hardware. Como se muestra, la DPAA 224 generalmente incluye varios administradores tales como una BMAN 226, una QMAN 228, y una primera y segunda FMAN 230 y 232, respectivamente.

El módulo CALInit 202 normalmente carga la configuración de la red LTE y cualquier regla de PCD estática a los administradores de marco 230 y 232 y configura la red CAL en base a un conjunto de archivos de configuración. El módulo CALInit 202 forma interfaz con una herramienta FMC (Herramienta de Configuración FMan) (no se muestra) o cualquier número de FMan API(s) (no se muestra) para configurar las PCD FMan, y la unidad CALDPAADriver 212 para cargar y ajustar la configuración de CAL (por ej., recursos de DPA del plano de usuario).

El módulo CALPcdFmc 208 proporciona reglas de Análisis, Clasificación y Distribución (PDC) y configuraciones que serán usadas por cada FMan (230, 232) para direccionar marcos de ingreso a los núcleos apropiados.

El módulo CALInit 202 es responsable de proporcionar varias funcionalidades. Para el núcleo principal en el arranque, el módulo CALInit 202 configura un marco CAL para soportar el procesamiento de "trayectoria rápida". Este paso puede incluir inicializar la unidad CALDPAADriver 212, que a su vez (a) crearía varios recursos de DPAA necesarios para procesar datos del plano de usuario (por ej., grupos de buffer, FQs (o colas de marco) y (b) crear la infraestructura CAL necesaria para soportar la gestión del buffer y servicios de mensajería a través de DPAA (por ej., tablas internas que mantienen la configuración del grupo de buffer, FQs, y la asociación entre las FQs de ingreso y las direcciones de IP del Planificador DL, etc.). El módulo CALInit 202 también carga reglas PCD de FMC de LTE (estáticas) y configuraciones de la red.

Se sabe que en un sistema de comunicación inalámbrico de múltiple acceso, los transmisores y receptores pueden comunicarse usando un paquete de comunicación de múltiples capas. Las capas pueden incluir, por ejemplo, una capa

física, una capa de control de acceso medio (MAC), una capa de control de enlace de radio (RLC), una capa de protocolo (por ej., capa de protocolo de convergencia de datos del paquete (PDCP)), una capa de aplicación y así sucesivamente. La capa de RLC recibe unidades de datos de servicio (SDU) de la capa de PDCP, y concatena o segmenta las SDU en unidades de datos del protocolo de RLC (PDU) para transmisión a la capa MAC.

5

De esta manera, el módulo CALBuf 204 facilita servicios de manejo del buffer de no bloqueo para aplicaciones L2 para usar en el procesamiento de SDU del RLC. Como se conoce en la técnica, un algoritmo de no bloqueo asegura que los hilos que compiten por un recurso compartido no tienen su ejecución infinitamente pospuesta por exclusión mutua. Un algoritmo de no bloqueo es sin bloqueo (o libre de bloqueo) si se garantiza el progreso a nivel del sistema. El módulo CALBuf 204 también soporta realizar cola para los datos estáticos del grupo de buffer (por ej., estado de agotamiento de grupo, recuento de agotamiento, estado de disponibilidad del grupo, recuento de error de asignación de grupo, etc.). El módulo CALBuf 204 forma una interfaz con CALDPAADriver 212 para implementar los servicios. El módulo CALBuf 204 proporciona un esquema de administración del buffer sin bloqueo que es extremadamente crítico para la operación del sistema apropiado en un entorno multi-núcleo, donde un bloqueo tomado por un proceso en tiempo no real puede causar problemas de latencia para un proceso en tiempo real que espera para la liberación de ese bloqueo.

10

15

La unidad CALDPAADriver 212 es el componente del espacio del núcleo de CAL 201 y ayuda a implementar y proporcionar servicios de administración del buffer y servicios de mensajería usando APIs BMAN y QMAN. Como se usa en la presente memoria, el término API (o interfaz de programación de aplicaciones) se refiere a una interfaz implementada por un programa de software que le permite interactuar con otro software. Facilita la interacción entre diferentes programas de software de manera similar al modo en que la interfaz de usuario facilita la interacción entre los usuarios y los ordenadores. Una API es implementada por aplicaciones, bibliotecas y sistemas operativos para determinar sus vocabularios y convenciones de llamada, y se usa para acceder a sus servicios. Puede incluir especificaciones para rutinas, estructuras de datos, clases de objetos y protocolos usados para comunicarse entre el consumidor y el implementador de la API.

20

25

La unidad CALDPAADriver 212 es responsable de administrar los recursos de DPAA (grupos de buffer y colas de marco) que se usarán para distribuir los datos del plano de usuario; proporcionar interfaz de espacio de usuario a otros módulos CAL a través de varias operaciones de archivo tales como abrir, liberar, control i-o (ioctl) para inicialización, administración del buffer, y servicios de mensajería; realizar mapeo del buffer del espacio del núcleo a usuario (K-U); proporcionar grupo del buffer DPAA y datos estadísticos de receptor y transmisor; e implementar servicios para administrar buffers de anillo. Se debe tener en cuenta que los buffers de anillo representan la cola de la L2 del software de CAL y generalmente se usa en para almacenar FDs destinados a un DLT L2 específico. El módulo CALMsg 206 proporciona APIs para que L2 recupere los descriptores del buffer de un anillo.

30

35

Todos los componentes CAL descritos con anterioridad generalmente son middleware de plataforma (ejecutados en el espacio de usuario), con excepción del CALDPAADriver 212. La unidad CALDPAADriver 212 es una unidad personalizada que se ejecuta en el espacio del núcleo, y está diseñada para implementar y prestar los servicios que necesita el middleware del espacio de usuario CAL, en particular aquellos servicios que dependen de los componentes de hardware P4080 DPAA.

40

El módulo CALBuf 204 presta servicios de administración del buffer que se usarán exclusivamente para procesamiento de datos de "trayectoria rápida". El módulo CALBuf 204 proporciona API del espacio de usuario a la aplicación de L2. El módulo CALBuf 204 colabora con el CALDpaaDriver 212 para proporcionar el servicio de administración de buffer de copia cero y no bloqueo para buffers que la unidad CALDpaa 212 crea pero son administrados por el BMAN 226.

45

El módulo CALMsg 206 proporciona servicios para recibir (ingresar) SDU de RLC y enviar (egresar) SDU de RLC a través de DPAA. El módulo CALMsg 206 también soporta la cola para los datos estadísticos de la interfaz de Ethernet Tx/Rx (por ej., número de FDs recibidos o transmitidos, número de FDs descartados, varios tipos de FDs malos, etc.). El módulo CALMsg 206 forma una interfaz con CALDPAADriver 212 para implementar los servicios. El módulo CALMsg 206 proporciona servicio de mensajería de copia cero de no bloqueo a la aplicación L2 de LTE para enviar o recibir paquetes de IP TCP/UDP sin el uso del paquete de protocolos. Esto asegura que el software de aplicación no encuentre picos de latencia no ligada que pueden descomponer el comportamiento propio del sistema LTE, que tiene requerimientos de procesamiento muy estrictos en tiempo real.

50

El módulo CALMsg 206 proporciona servicios de mensajería al software L2 para enviar y recibir datos del plano de usuario a o de otro tablero (es decir, eCCM). El módulo CALMsg 206 generalmente hace interfaz con el CALDPAADriver 212 para proporcionar servicios de mensajería de copia cero de no bloqueo a través de DPAA. Esta particularidad permite que el software de la aplicación de L2 envíe y reciba paquetes de IP TCP/UDP sin el uso del paquete de protocolo para evitar demoras de latencia no ligada.

60

El módulo CALMsg 206 implementa y proporciona APIs que soportan varios servicios, tales como los descritos en los siguientes párrafos.

5 Un servicio posible es el registro de las entidades de la aplicación (de L2) con el servicio CALMsg, en donde una entidad puede recibir paquetes entrantes a través de "trayectoria rápida." Durante este proceso de registro, se crea una cola de la L2 del software de CAL (es decir, un anillo de los descriptores del buffer, o en otras palabras, un buffer circular para los descriptores del buffer) para mantener los descriptores del buffer recibidos destinados a la entidad. También durante este registro, el módulo CALMsg 206 crea una asociación entre las FQ de ingreso a la dirección de IP y el anillo de los ID del descriptor del buffer para posterior referencia en otro procesamiento (por ej., determinando qué anillo de los descriptores del buffer empujan un descriptor del buffer a cuando el marco llega a una FQ). El módulo CALMsg 206 también realiza mapeo de núcleo a espacio de usuario de los anillos relevantes de los descriptores del buffer, y configura la regla PCD para la entidad de aplicación (si no fue realizado aún por reglas estáticas). Además, al comienzo del proceso de registro, el CAL 201 implementa una estrategia de defensa para asegurar que todos los buffers adquiridos por la aplicación sean liberados apropiadamente cuando un hilo se cuelga.

15 Un segundo servicio proporcionado por el módulo CALMsg 206 es recuperar un marco destinado a la entidad de aplicación. La dirección del buffer devuelto apunta al inicio del área de datos comenzando con el encabezado de Ethernet.

20 Un tercer servicio proporcionado por el módulo CALMsg 206 es enviar un mensaje a una entidad externa a través de DPAA en la interfaz de Ethernet configurada para procesar los datos del plano de Usuario (por ej., eth0). Se espera que la L2 ingrese todos los encabezados (Ethernet, IP, UDP) necesarios y, que el hardware esté configurado apropiadamente para generar e ingresar una suma de verificación de IP y una suma de verificación de UDP.

25 Un cuarto servicio es hacer cola para datos estadísticos del puerto receptor y transmisor.

Un quinto servicio es eliminar el registro de una entidad de aplicación del módulo CALMsg 206. Una vez que se ha eliminado el registro de la entidad de aplicación, ya no será capaz de recibir paquetes a través de la "trayectoria rápida". Como parte del proceso de eliminación del registro, el CAL liberará todos los buffers adquiridos por el software de aplicación. Para el caso en el que el módulo CALMsg 206 se use para recibir marcos a través de la trayectoria rápida, el anillo asociado de los descriptores del buffer y la regla PCD también serán retirados.

35 Con referencia a la FIG. 3, se muestra una visión general de alto nivel de un sistema de Unidad de Ethernet de Plano posterior (BED) basado en DPA 300 adaptado para proporcionar el flujo de datos de ingreso de los paquetes. El lado izquierdo de la FIG. 3 ilustra la trayectoria tomada por los paquetes destinados a ir al núcleo X del plano de control usando un paquete de protocolo tradicional. El lado derecho del diagrama ilustra el paquete transversal para el núcleo Y del plano de datos donde el Planificador L2 de LTE se encuentra en ejecución.

40 Para fines ilustrativos, el planificador L2 de LTE consiste en dos hilos en tiempo real. Los dos hilos incluyen el Planificador DL que se muestra como PLANIFICADOR DL 324 en la FIG. 3 que entra en acción para flujo de paquetes de ingreso, y un planificador de enlace ascendente mostrado como PLANIFICADOR UL que entra en acción para el flujo de paquetes de egreso.

45 Desde la perspectiva del hardware, cada núcleo tiene un portal de software 316. Cada portal tiene un número especificado de canales. Cada canal tiene ocho colas de trabajo (WQ) basadas en prioridades que comprenden la pluralidad de colas de marco (FQ). A un alto nivel, cuando FMAN 302 recibe un paquete de datos 304, adquiere un buffer mapeado en espacio de Núcleo-Usuario de tamaño adecuado de BMAN 226, y copia el paquete recibido en el buffer, el FMAN 302 determina (de acuerdo con las reglas de PCD) qué FQ ingresa 306 para sacar de cola el paquete, y usa la QMAN API 308 para sacar de cola el paquete de datos 304 sobre la FQ apropiada FQ 306. Cuando el paquete de datos 304 hace su recorrido al registro de hardware DQRR 328, se genera una interrupción de portal para el núcleo especificado (tal como el Núcleo Y 312). En otras palabras, QMAN maneja todas las FQ para los diversos canales. Las WQ están allí para implementar la calidad de servicio (QoS). De modo que cada paquete que llega habiendo estado puesto en cola en una FQ atraviesa colas de hardware internas dentro de QMAN. Las FQ que llegan delante de la cola, y están listas para procesamiento, se colocan en el registro de DQRR por el QMAN. El registro de DQRR tiene una profundidad de 15 entradas. Como cada entrada (que es un descriptor de cola de marco que contiene información sobre la FQ, tal como el puntero al buffer del paquete, una función de llamada asociada, etc.), es mantenida por la función de llamada registrada 330 en la rutina de servicio de Interrupción (ISR), el QMAN 308 automáticamente pone en cola el siguiente descriptor de FQ en el DQRR.

60 Luego, se ejecuta la función de llamada registrada "UplaneDispatcher" 330, que es una parte de la Unidad DPA CAL 314 para la FQ 306, en el espacio del núcleo y realiza el procesamiento adecuado del paquete. El anillo de los descriptores del buffer 320 para cada instancia del Planificador DL tiene un ID de Anillo. El anillo del descriptor del buffer 320 también está mapeado del núcleo al espacio de usuario (K-U) para permitir que los componentes del espacio de usuario (como por

- ejemplo, CALMSG 326), y del espacio del núcleo (como por ejemplo, la Unidad DPA CAL 314) de CAL 128 accedan a éste sin ningún problema de direccionamiento y sin la necesidad de copia de datos. Durante la inicialización celular, el CALMSG 326 proporciona un servicio a la aplicación de usuario Planificador DL 324 para registrarse con CALMSG 326. En ese momento también se crea una asociación de mapeo entre la FQ de ingreso 306, y el ID de anillo 320. La función de llamada 330 determina a qué ID de anillo 320 pertenece este descriptor de cola de marco particular, y crea un descriptor de buffer mediante la copia del puntero de buffer mapeado en espacio de núcleo a usuario (junto con cualquier info de registro de contabilidad) y puesta en cola del descriptor del buffer sobre el anillo mapeado en espacio de núcleo-usuario apropiado de los descriptores de buffer 320 para procesamiento posterior mediante el hilo del planificador DL de espacio de usuario asociado 324 en un núcleo Y del plano de datos particular 312.
- El proceso del Planificador DL 324 se ejecuta en un modo de "Ejecutar hasta la Compleción" en el espacio de usuario. El hilo en tiempo real despierta cada momento Delta T, e invoca la API CALMSG 326 para consultar, si hay paquetes en cola para este consumo en el anillo de los descriptores del buffer 320. Si hay, el Planificador DL 326 procesa (o consume) los paquetes en el anillo 320. Dado que el paquete recibido no ha atravesado el paquete de protocolo estándar, éste contiene el encabezado completo del paquete (TCP/UDP/IP/Ethernet). El Planificador DL 324 ignora el encabezado del paquete y opera en el área de datos del paquete. Cuando se realiza, libera el buffer usando la API CALBUF 204. Una vez que ya no hay más paquetes para que el planificador DL 324 procese, el hilo del planificador DL 324 vuelve al estado de reposo. Dado que solo hay un productor (CAL DPADriver 314) y un consumidor (un hilo del Planificador DL específico 324) para cada anillo de descriptores del buffer 320, este esquema puede implementarse con operaciones atómicas que no requieren ningún bloqueo del software. Una ventaja adicional se logra por el uso de buffers mapeados del núcleo al espacio de usuario junto con un anillo de los descriptores del buffer tal como el anillo de los descriptores del buffer 320. El uso de los buffers mapeados del núcleo al espacio de usuario obvia cualquier necesidad de copia de datos porque el paquete atravesará desde las unidades del dispositivo de espacio del núcleo hasta su destino final en el espacio de usuario.
- En la realización a modo de ejemplo, el tráfico del plano de control y el tráfico del plano de datos se aíslan unos de otros y se dirigen a interfaces de Ethernet separadas. Un FMAN 302 puede soportar múltiples interfaces de Ethernet. El paquete de protocolo Linux se une a, y se ejecuta en, el núcleo del plano de control X 332. El módulo de PCD CAL establece reglas PCD estáticas y permite que el FMAN 302 ponga en cola todos los paquetes 304 que llegan a la interfaz de Ethernet del plano de control a una Cola de Marco (FQ) 334 que será manipulada por la unidad DPA FSL 336 que se ejecuta en el espacio de Núcleo. La unidad 336 traslada los paquetes al paquete de protocolos Linux 338. Una copia de datos también está involucrada cuando el paquete cruza el espacio de núcleo al límite del espacio de usuario. Esto se muestra en el lado izquierdo de la FIG. 3.
- Una vista de alto nivel del flujo de proceso con respecto a un paquete de datos entrante para el núcleo del plano de datos se ilustra en la FIG. 4. El método de la FIG. 4 puede implementarse por el aparato ilustrado en la FIG. 3.
- En la etapa 410, un Planificador DL 320 se registra con el CALMSG 326. El sistema también crea un anillo de los descriptores del buffer 320 que se mapean del espacio de núcleo al espacio de usuario (K-U) para permitir que tanto la entidad de espacio de núcleo (tal como CALDPAADriver 314), como la entidad del espacio de usuario (tal como la CALMSG 326) accedan al anillo. Este paso puede realizarse por CALMSG 326 en conjunción con el módulo CALDPAADriver 314 para cada celda configurada en el tablero moderno. De esta manera, si hay tres celdas soportadas en un tablero moderno, entonces hay tres anillos separados de descriptores de buffer 320 (uno para cada celda), y CALDPAADriver 314 tendrá un mapeo entre la FQ de ingreso FQ 306 a un anillo específico de los descriptores del buffer 320. En la realización de ejemplo, solo hay un anillo de los descriptores del buffer 320 para cada instancia del PLANIFICADOR DL 324. El anillo de los descriptores del buffer tiene un puntero de lectura y un puntero de escritura. Dado que solo hay un productor (tal como, por ejemplo, el uplaneDispatcher 330 de CALDPAADriver 314 en el espacio de núcleo), y un consumidor (tal como el Planificador DL 324 en el espacio de usuario), el anillo de los descriptores del buffer 320 se implementa en un modo de no bloqueo usando operación atómica.
- En el paso 420, la interfaz de Ethernet del plano de datos específico recibe un paquete de datos 304.
- En el paso 430, el FMAN 302 adquiere un buffer/token mapeado separado de núcleo a usuario del BMAN (FIG. 2, 226).
- En la etapa 440, el FMAN 302 inserta el paquete de datos recibidos 304 en el buffer adquirido.
- En el paso 450, según las reglas de PCD dinámicas establecidas por el módulo CAL (CALMSG 206) durante la inicialización celular (que proporciona un mapeo entre una dirección de IP y una FQ de ingreso 306), el FMAN 302, con ayuda de QMAN 308, pone en cola el paquete sobre la FQ apropiada 306.
- En el paso 460, cuando el paquete se hace camino sobre el registro de DQRR de hardware 328, el sistema, a través de un procesador multi-núcleo P4080, genera una interrupción para el portal del software 316 para el Núcleo Y 312 (el núcleo

- 5 en el plano de datos). La función de llamada que se registra para la FQ específica (en este caso el `uplaneDispatcher 330`) se ejecutará cuando la rutina ISR en el espacio del núcleo procese el paquete. En esencia, la función de llamada crea un descriptor del buffer de la información contenida en el Descriptor de Cola del Marco (FD) de la entrada del DQRR procesada. Un puntero del buffer de FD se copia en el descriptor del buffer, junto con cierta información adicional de contabilidad.
- En el paso 470, la unidad CALDPAA 314 pone en cola al descriptor de buffer generado en el paso 460 en el anillo de los descriptores de buffer 320 asociados con el núcleo Y 312.
- 10 Los pasos 410 a 470 se realizan por hardware (FMAN y QMAN) en conjunción con el hilo del núcleo de CALDPAADriver 314 (por ej., el `uplaneDispatcher 330`). En la realización de ejemplo, los pasos 480 a 495 se realizan en un proceso o hilo independiente del (de los) proceso(s) o hilo(s) que realizaron los pasos 410 a 470. De esta manera, los pasos 480-495 se realizan por hilos del espacio de usuario tal como el Planificador DL 324 en el núcleo Y del plano de datos 312.
- 15 Contemporáneo con los pasos 410 a 470, el Planificador DL 324 se ejecuta en un ciclo sin fin al modo completación, y se activa en un intervalo de TTI (Tiempo para Interrumpir). En el paso 480, una vez activo, el Planificador DL 324 invoca un CALMSG API 326 para consultar si hay algún paquete no procesado en el anillo de los descriptores del buffer 320 para consumo. Si no hay paquetes de datos no procesados en el anillo de los descriptores del buffer 320, el PLANIFICADOR DL 324 recibirá una respuesta de no bloqueo del módulo CALMSG 326 que indica que no hay paquetes no procesados.
- 20 El hilo del Planificador DL 324 luego queda en reposo hasta el siguiente TTI.
- En el paso 485, si el Planificador DL 324 determina que hay paquetes de datos no procesados, entonces el método avanza al paso 490, y así el método avanza a 495.
- 25 En el paso 490, el Planificador DL 324 procesa el siguiente paquete de datos en el anillo de los descriptores del buffer 320. El siguiente paquete se indica mediante el puntero de lectura del anillo. Después de que se procesa el paquete de datos, el Planificador DL 324 libera el buffer mapeado K-U usando API CALBUF 204. En la realización de ejemplo, el paquete de datos que será procesado contiene ambos de uno o más encabezados y un área de datos. El Planificador DL 324 ignora el encabezado y opera en el área de datos. El encabezado se usa para direccionar el paquete hasta su destino,
- 30 lo que se logra en la realización de ejemplo sin el uso de un paquete de protocolo. Una vez que el PLANIFICADOR DL 324 procesa todos los paquetes, el método avanza al paso 495 donde el PLANIFICADOR DL descansa.
- La realización de ejemplo soporta tres celdas (actualizables a seis celdas). Cada celda tendrá su propio Planificador DL (hilo del PLANIFICADOR DL 324) que se une a, y se ejecuta en, su propio núcleo del plano de datos específico. De esta manera, en la realización de ejemplo, hay un mínimo de tres hilos de PLANIFICADOR DL (FIG. 1, 116, 118, y 120) que se ejecutan en tres núcleos del plano de datos diferentes. De manera similar cada celda necesita un Planificador L2 UL (PLANIFICADOR UL). De esta manera, hay tres hilos del PLANIFICADOR UL (FIG. 1, 110, 112, y 114) que se ejecutan en tres núcleos del plano de datos diferentes.
- 40 Durante la configuración de la celda, las reglas de PDC dinámicas (es decir, el mapeo de las direcciones de IP a las FQ de ingreso) se ajustan por el módulo CALMSG 326. El módulo CALMSG 326 también puede realizar funciones de contabilidad, tal como mapeo de FQ de ingreso a los ID del anillo de los descriptores del buffer. Las reglas de PCD permiten que el FMAN examine la dirección de IP de cada paquete y determine en qué FQ poner en cola el paquete. La información de contabilidad adicional (tal como la FQ de ingreso 306 al mapeo de ID del anillo de los descriptores del buffer 320)
- 45 permite que el hilo del núcleo `uplaneDispatcher 330` decida en qué anillo de los descriptores del buffer debe poner un descriptor del buffer dado con un paquete de datos entrantes. De esta manera, se logra un esquema de mensajería de copia cero y sin bloqueo que permite el ingreso de paquetes para alcanzar un hilo del planificador DL 324 que se ejecuta sobre el núcleo del plano de datos Y 312 sin el uso de un paquete de protocolo convencional.
- 50 Con referencia a la FIG. 5, se muestra una visión general de alto nivel de un sistema de Unidad de Ethernet de Plano posterior (BED) basado en DPA 500 adaptado para proporcionar el flujo de datos de egreso de los paquetes. La realización de ejemplo 500 proporciona una solución que no usa el paquete de protocolos Linux para transmitir el tráfico del plano de datos de egreso. El lado izquierdo de la FIG. 5 ilustra la trayectoria tomada por los paquetes enviados del núcleo X del plano de control 532 usando un paquete de protocolo tradicional 538. La aplicación al espacio de usuario en tiempo no real tal como OAM&M 124 que se ejecuta en el núcleo del plano de control X 532 envía paquetes de IP TCP/UDP IP
- 55 usando el paquete de protocolo 538. El paquete de protocolo 538 envía el paquete a la unidad DPA FSL 536 en el espacio de núcleo. El proceso de enviar el paquete del núcleo X 532 en el espacio de usuario a la unidad DPA FSL 536 en el espacio de núcleo requiere una copia de los datos. La unidad DPA FSL 536 pone en cola al paquete en una FQ 534 que finalmente recibe el servicio de FMAN 506 con ayuda de QMAN 508, y el paquete se transmite en la interfaz de Ethernet.
- 60 El lado derecho del diagrama ilustra el paquete transversal para paquetes que se originan en el núcleo Y 512 del plano de datos donde el Planificador UL de L2 de LTE 502 se encuentra en ejecución sin el uso del paquete de protocolo 538. De

este modo, el flujo de datos de egreso también usa un esquema de mensajería de copia cero, sin bloqueo que no incurre en picos de latencia ilimitados.

5 Una vista de alto nivel del flujo de proceso con respecto a un paquete de datos saliente (es decir, flujo de datos de egreso para el núcleo del plano de datos) se ilustra en la FIG. 6. El método de la FIG. 6 puede implementarse por el aparato ilustrado en la FIG. 5.

En el paso 610, el planificador UL 502 adquiere un buffer mapeado separado de núcleo a usuario del CALBUF 522.

10 En el paso 620, cuando el planificador UL 502 está listo para transmitir un paquete de IP TCP/UDP, el planificador UL 502 prepara el paquete completo (con encabezado y carga), y luego invoca la API CALMSG 522.

15 En el paso 630, el CALMSG 522, con ayuda de CALDPAADriver 524, pone en cola al paquete preparado en la FQ de egreso apropiada 504 del canal CHB asociado con la interfaz de Ethernet del plano de datos. Esta eliminación de cola se realiza sin el uso de la pila de protocolo y confía en su lugar en el esquema de mensajería cero, de no bloqueo para evitar picos de latencia ilimitada.

20 En el paso opcional 640, el QMAN 508 determina un estado de si la operación de poner en cola fue exitosa o no, y esta respuesta se envía de regreso al Planificador UL de la aplicación de usuario 502 si así desea recibir el estado.

25 En el paso 650, el FMAN 506, con ayuda de QMAN 508, procesa todos los paquetes de las varias FQ 504. Aunque hay un cambio de contexto involucrado siempre que el paquete atraviese el límite de espacio de usuario a espacio de núcleo, no hay copia de datos involucrada. De este modo, el procesamiento de paquetes por FMAN 506 se realiza en un modo eficaz para el hardware de modo tal que cada núcleo individual tenga penalidad de rendimiento insignificante o ninguna. Cada Planificador UL 302 tiene una FQ 504 específica en el canal de destino (CH B 526) para transmitir paquetes a la interfaz de Ethernet. También hay un productor (una instancia del hilo de PLANIFICADOR UL 502 en el espacio de usuario), y una FQ de egreso específico 504 que es consumida por FMAN 506 en el hardware.

30 En el paso 660, una vez que se transmite el paquete, el FMAN 506 libera el buffer nuevamente a BMAN (FIG. 2, 226).

35 Una persona capacitada en la técnica reconocería ampliamente que los pasos de los diversos métodos descritos anteriormente pueden ser realizados por ordenadores programados. En la presente memoria, algunas realizaciones también pretenden cubrir dispositivos de almacenamiento, por ejemplo, medios de almacenamiento de datos digitales, que son legibles por máquinas o por ordenadores y codifican programas de instrucciones ejecutables por máquinas o ejecutables por ordenadores, en los que dichas instrucciones realizan algunos o todos los pasos de dichos métodos descritos anteriormente. Los dispositivos de almacenamiento de programas pueden ser, por ejemplo, memorias digitales, medios de almacenamiento magnéticos tales como discos magnéticos y cintas magnéticas, discos duros, o medios de almacenamiento de datos digitales legibles de manera óptica. Las realizaciones también pretenden cubrir ordenadores programados para realizar dichos pasos de los métodos descritos anteriormente.

40 La descripción anterior simplemente proporciona una divulgación de las realizaciones particulares de la invención y no está destinada para los fines de limitar la misma a ella. De esta manera, la invención no está limitada solamente a las realizaciones descritas anteriormente. Más bien, se reconoce que un experto en la técnica concebiría realizaciones alternativas que caen dentro del alcance de la invención.

45

**REIVINDICACIONES**

5 1. Un método implementado por ordenador para la administración del buffer para un procesador multi-núcleo para usarse en un módem en una red de telecomunicaciones, estando el método caracterizado por:

para cada uno de una pluralidad de núcleos de procesamiento unidos al módem:

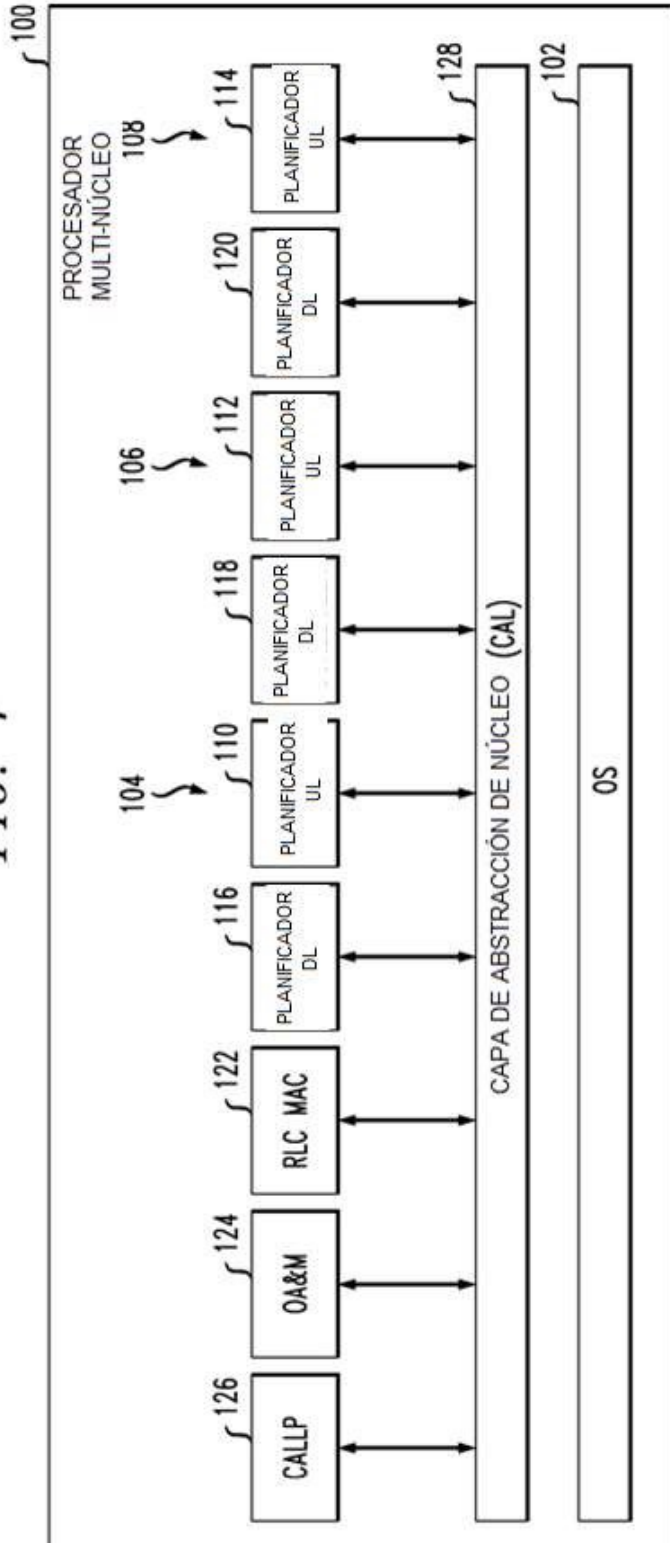
10 recibir (420) un paquete de datos de IP de TCP o UDP (304);  
adquirir (430) un buffer mapeado de núcleo a espacio de usuario (KU) y descriptor de buffer mapeado K-U correspondiente (320), en el que el descriptor está asociado con un paquete de datos y es indicativo del buffer mapeado K-U;  
insertar (440) el paquete de datos en el buffer; e  
insertar (470) el descriptor de buffer en un buffer circular mapeado K-U;  
15 en el que el descriptor del buffer se crea:

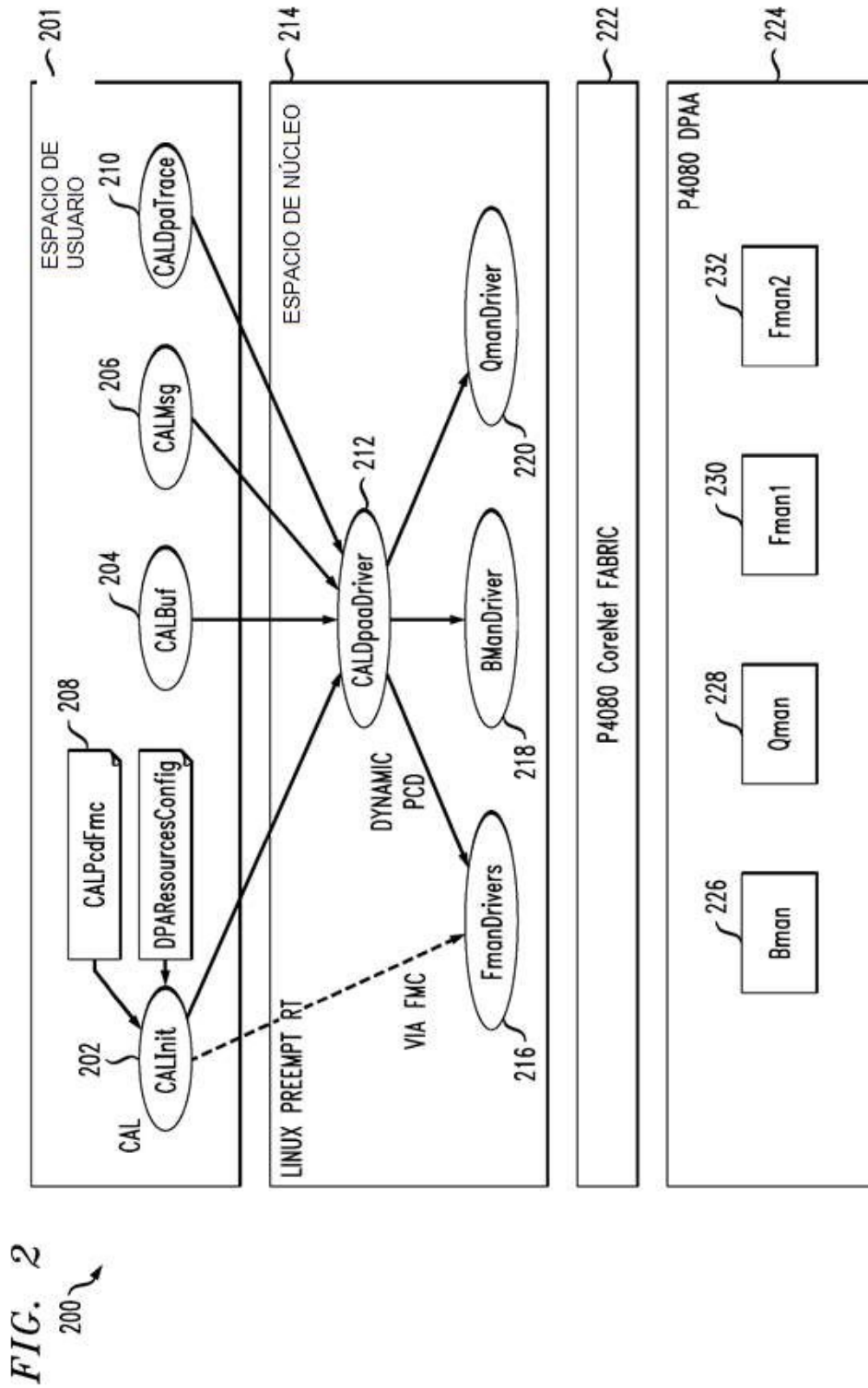
creando un descriptor del marco que contiene un puntero de buffer mapeado K-U adquirido correspondiente al buffer mapeado K-U;  
insertando el descriptor del marco en una cola de marco especificada por una regla dinámica que mapea direcciones de IP a las colas de marco; y  
20 creando el descriptor del buffer a partir del descriptor de marco.

2. El método de acuerdo con la Reivindicación 1, en el que la adquisición de un buffer mapeado K-U, la inserción del paquete de datos (304) en el buffer y la inserción del descriptor del buffer en el buffer circular se realizan por uno o más hilos del núcleo en conjunción con componentes de hardware, y que además comprende:

25 ejecutar un hilo del espacio de usuario (201) que se activa en intervalos periódicos;  
en el que el hilo del espacio de usuario, tras activarse, consulta al buffer circular para determinar si cualquier descriptor del buffer no procesado existe en el buffer circular; y  
tras determinar que un descriptor del buffer no procesado existe en el buffer circular, procesar el paquete de datos (304)  
30 dentro del buffer usando el mismo hilo de espacio de usuario o uno diferente.

FIG. 1





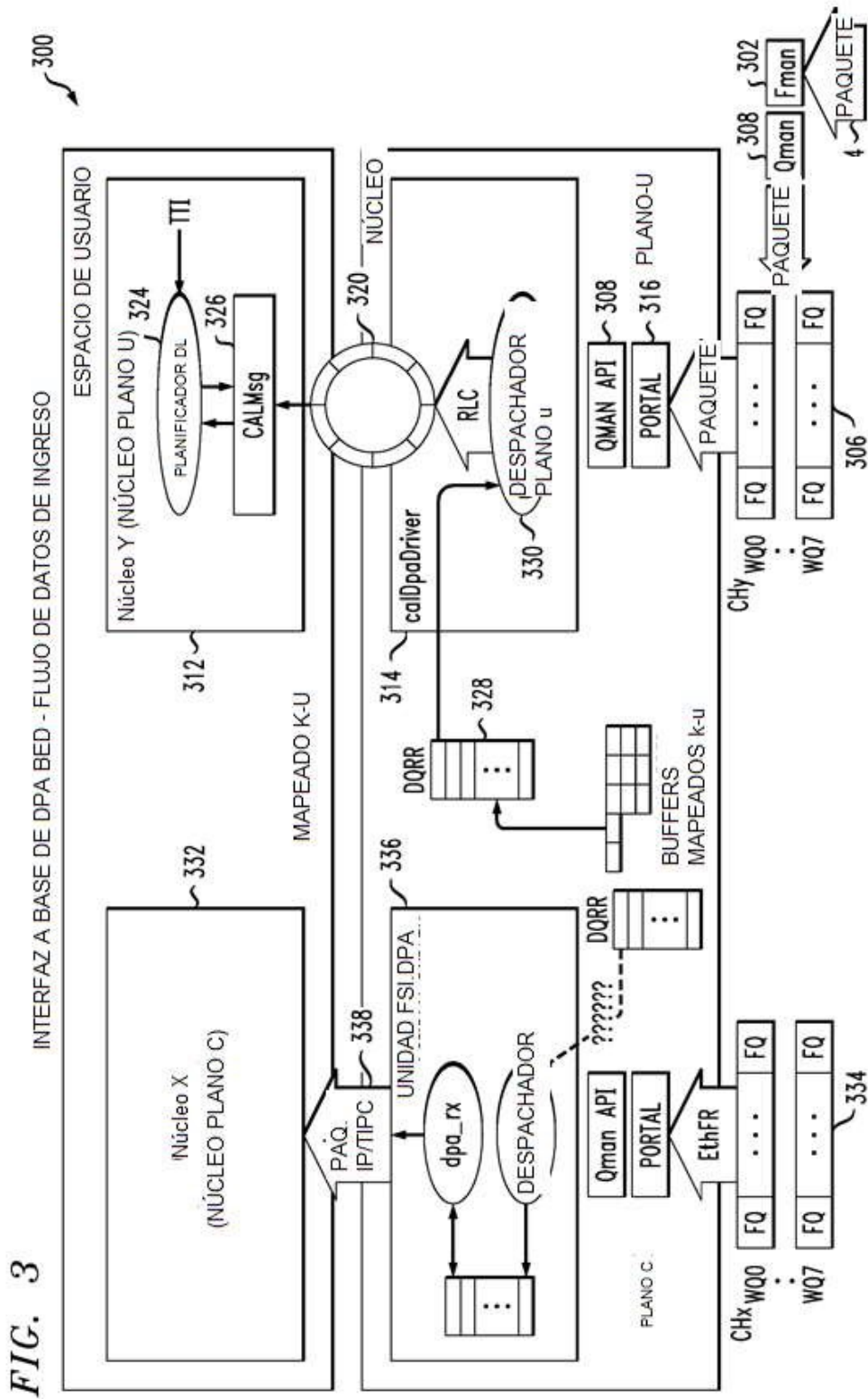
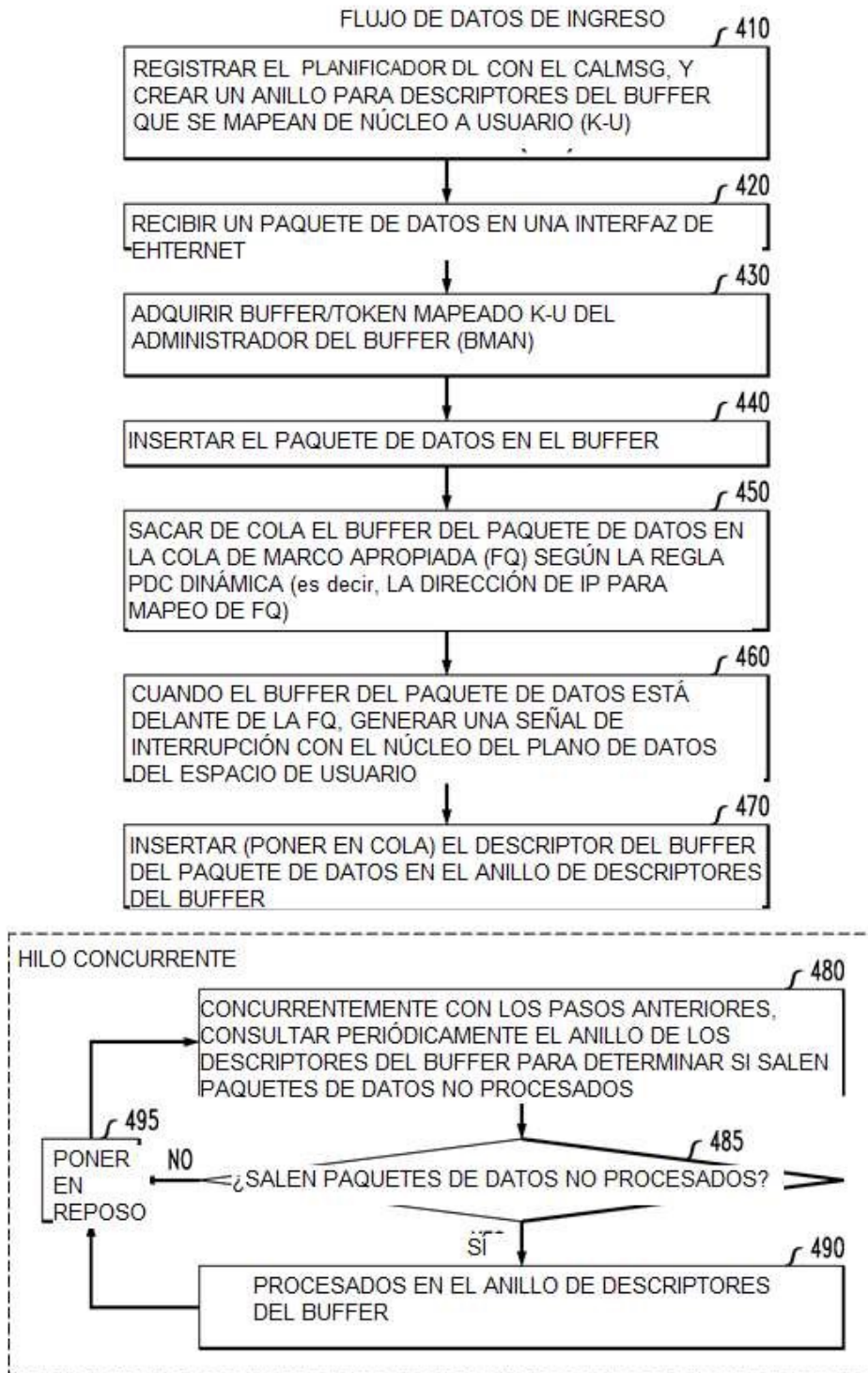
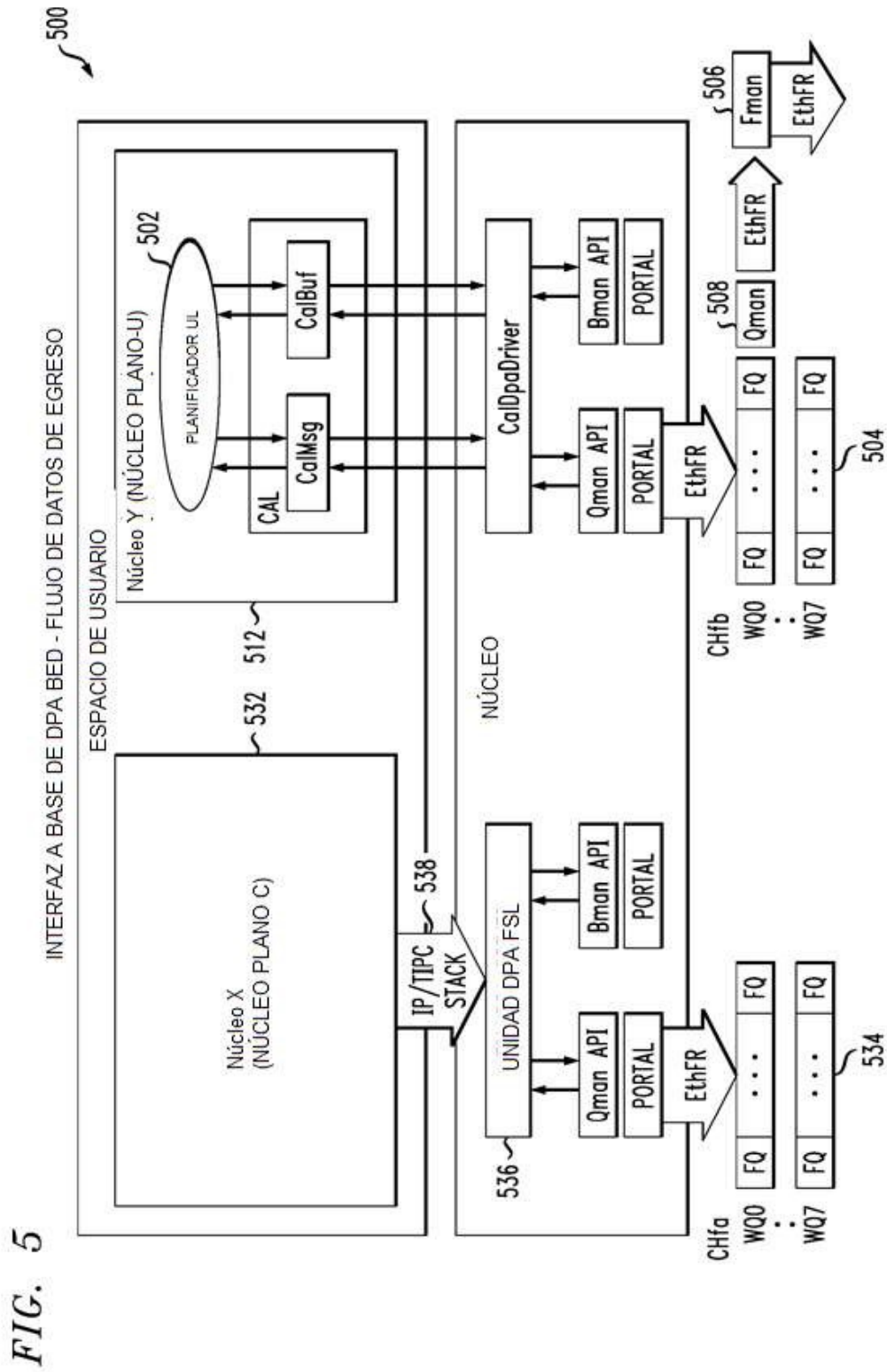


FIG. 4





*FIG. 6*

