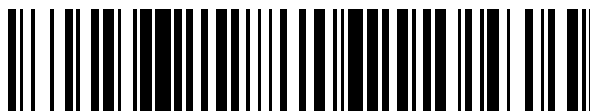


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 703 186**

51 Int. Cl.:

H04L 29/08 (2006.01)

G06F 17/30 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **13.05.2016** **E 16169624 (0)**

97 Fecha y número de publicación de la concesión europea: **26.09.2018** **EP 3244590**

54 Título: **Nodo de red, nodo terminal y procedimiento de recepción de un mensaje de interés**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
07.03.2019

73 Titular/es:

KONINKLIJKE KPN N.V. (50.0%)
Wilhelminakade 123
3072 AP Rotterdam, NL y
NEDERLANDSE ORGANISATIE VOOR
TOEGEPAST- NATUURWETENSCHAPPELIJK
ONDERZOEK TNO (50.0%)

72 Inventor/es:

GILL, AMRITPAL SINGH;
D'ACUNTO, LUCIA;
TRICHIAS, KONSTANTINOS y
VAN BRANDENBURG, RAY

74 Agente/Representante:

LEHMANN NOVO, María Isabel

ES 2 703 186 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

Nodo de red, nodo terminal y procedimiento de recepción de un mensaje de interés

Sector técnico de la invención

La invención se refiere a un nodo de red.

- 5 La invención se refiere además a un procedimiento de recepción de un mensaje de interés, en particular a un procedimiento recepción de un mensaje de interés y un mensaje de datos.

La invención se refiere asimismo un producto de programa informático que permite a un sistema informático llevar a cabo dicho procedimiento.

La invención se refiere además a un nodo terminal para transmitir un mensaje de interés.

10 Antecedentes de la invención

Redes centradas en información (ICN, Information-Centric Networking) es un nuevo paradigma de red que tiene el objetivo de evolucionar la infraestructura de internet actual, de un sistema orientado al anfitrión hacia un sistema orientado a los datos. En lugar de direccionar terminales por medio de direcciones IP, los propios datos se direccionan en ICN. Dividiendo los datos en trozos, y proporcionando a cada uno estos trozos un nombre único y jerárquico, ICN permite que los clientes soliciten a la red un determinado objeto de datos con nombre, sin tener que preocuparse sobre dónde están situados/almacenados los datos. Uno de los beneficios de poner nombre a los datos es que cada nodo de red intermedio, tal como un conmutador o un encaminador, tiene la capacidad de almacenar en memoria caché paquetes de datos que fluyen a su través. De este modo, si un nuevo usuario solicita datos que están ya almacenados en memoria caché en la red, esta solicitud no tiene que viajar a través de la red hacia el generador original de los datos, sino que puede en cambio ser servida por uno de los nodos de red intermedios que mantienen una copia de dichos datos.

Las actuales arquitecturas de ICN tienen dos tipos elementales de mensajes en su núcleo: mensajes de interés y mensajes de datos. Cuando una aplicación desea recuperar un objeto de datos particular, emite un mensaje de interés para dicho objeto de datos. La parte más importante de este mensaje de interés es el nombre del objeto o, en casos en los que no se conoce el nombre completo del objeto, un prefijo.

Un ejemplo de dicha arquitectura ICN es la Red de datos con nombre (NDN, Named Data Networking) descrita en "Named data networking", Zhang et al., ACM SIGCOMM Computer Communication Review 44 Número 3, julio de 2014, páginas 66 a 73. Cuando un nodo de red recibe un mensaje de interés en NDN, verifica en primer lugar si se da el caso de que su memoria caché local contenga un objeto de datos con nombre que se corresponda con el nombre en el mensaje de interés. Si se encuentra un correspondiente objeto de datos con nombre, se envuelve en un mensaje de datos y se devuelve sobre la interfaz sobre la que se recibió el mensaje de interés.

En caso de que el objeto no esté almacenado en memoria caché, la siguiente etapa consiste en comprobar si el nombre solicitado está ya en la denominada tabla de intereses pendientes (PIT, Pending Interest Table). La PIT mantiene una lista de solicitudes existentes de objetos de datos, pendientes, y los interfaces correspondientes a cuyo través fueron solicitadas. Cuando entra un mensaje de datos, éste se reenvía a todas las interfaces que, de acuerdo con la PIT, tienen un interés insatisfecho por el nombre correspondiente. A continuación, la entrada en la PIT es eliminada. Cuando un nombre en un mensaje de interés entrante no está listado en la PIT, se añade junto con la interfaz sobre la que fue recibido. A continuación, el mensaje de interés se reenvía sobre una o varias interfaces de salida de acuerdo con la base de información de reenvío (FIB, Forwarding Information Base) y la capa de estrategia, en búsqueda de los datos solicitados en otro nodo.

El almacenamiento en caché en ICN tiene el potencial de reducir la latencia experimentada por el usuario, reduciendo al mismo tiempo el tráfico de red. Sin embargo, dado que cualquier nodo en el trayecto desde el origen del contenido (que puede ser un nodo de almacenamiento caché intermedio o el generador original) hasta el usuario puede almacenar el contenido en caché, gestionar las decisiones de almacenamiento en caché de los nodos de almacenamiento caché se vuelve crucial para evitar un uso ineficiente de dicha capacidad de almacenamiento caché en la red y un rendimiento por debajo del óptimo en términos de tráfico y de latencia. Decisiones importantes de almacenamiento en caché que tienen que ser gestionadas cuidadosamente son: 1. Qué elemento de datos debería almacenar una caché específica, y 2. Si el almacenamiento caché está lleno, qué elemento de datos debería ser sustituido.

50 Parece evidente que es más probable que se produzcan decisiones ineficientes de almacenamiento en caché cuando los nodos adoptan dichas decisiones de manera independiente entre sí, basándose exclusivamente en información "local" (es decir, información disponible en cada propio nodo de almacenamiento caché, tal como los elementos que se solicitan, los elementos que se almacenan en caché, la frecuencia de las solicitudes, etc.), que es el enfoque estándar en ICN. Sin embargo, se hace necesario considerar algún nivel de visión "global" (por ejemplo, ¿está el elemento recién recibido almacenado ya en una caché vecina?, ¿son los datos que almacena actualmente

una caché más populares que un elemento recién recibido?) para adoptar decisiones eficientes de almacenamiento en caché en una red centrada en información.

El documento "Age-based cooperative caching in information-centric networking", de Zhongxing Ming, Mingwei Xu, y Dan Wang., 23rd International Conference on Computer Communication and Networks, 2014, publicado por IEEE, da a conocer un procedimiento en el que se almacenan elementos en caché junto con un campo de antigüedad que indica un tiempo de vencimiento. Cuando transcurre el tiempo de vencimiento, el elemento es sustituido por otro elemento. El valor del campo de antigüedad almacenado junto con el elemento en la caché se basa en el campo de antigüedad del mensaje de datos recibido que comprende el elemento. La antigüedad de un elemento es ajustada por el primer nodo junto con el trayecto, en base a la popularidad del elemento. Cada subsiguiente nodo a lo largo del trayecto aumenta (por ejemplo, duplica) la antigüedad en el mensaje de datos antes de reenviar el mensaje de datos. El resultado es que las caché en el borde comprenden los elementos más populares, mientras que las caché más alejadas del borde comprenden asimismo elementos menos populares. Una ventaja de este procedimiento es que no requiere mensajes de señalización (independientes) entre encaminadores.

Un inconveniente de este procedimiento es que las caché no son utilizadas eficientemente, debido a que es probable que determinados elementos populares se almacenen en caché tanto en el borde como en uno o varios nodos más alejados del borde, mientras que es probable que determinados elementos menos populares no se almacenen en caché en absoluto. Ejemplos adicionales de técnicas de almacenamiento en caché en ICN son: "Caching in information centric networking: A survey", XP028744689, de Zhang et al, y "Distributed Cache Management in Information-Centric networks", XP011526912, de Surlas Vasilis et al.

Resumen de la invención

La invención se define en la reivindicación independiente 1 relacionada con un nodo de red, la reivindicación independiente 11 relacionada con un nodo terminal, la reivindicación independiente de procedimiento 12 y la reivindicación independiente 13 relacionada con un programa informático o un conjunto de programas informáticos.

Un primer objetivo de la invención es dar a conocer un nodo de red, que facilite un uso más eficiente de las cachés en una red centrada en información.

Un segundo objetivo de la invención es dar a conocer un procedimiento de recepción de un mensaje de interés, que facilite una utilización más eficiente de las cachés en una red centrada en información.

Un tercer objetivo de la invención es dar a conocer un nodo terminal, que facilite una utilización más eficiente de las cachés en una red centrada en información.

De acuerdo con la invención, se consigue el primer objetivo en que el nodo de red comprende por lo menos una interfaz de comunicación, por lo menos una memoria, y por lo menos un procesador configurado para recibir un mensaje de interés sobre dicha por lo menos una interfaz de comunicación procedente de un nodo de más abajo, comprendiendo dicho mensaje de interés un nombre de un objeto de datos solicitado, determinar si pujar para almacenar dicho objeto de datos solicitado en dicha por lo menos una memoria, calcular un valor de puja, reenviar dicho mensaje de interés a un nodo de más arriba en dicha por lo menos una interfaz de comunicación, incluyendo dicho mensaje de interés dicho valor de puja calculado en función de, por lo menos, dicha determinación sobre si pujar, recibir un mensaje de datos que comprende dicho objeto de datos solicitado, desde dicho nodo de más arriba sobre dicha por lo menos una interfaz de comunicación, determinar a partir de por lo menos dicho mensaje de datos si dicho nodo de red ha ganado dicha puja, almacenar dicho objeto de datos en dicha por lo menos una memoria si dicho nodo de red ha ganado dicha puja, y reenviar dicho mensaje de datos a dicho nodo de más abajo sobre dicha por lo menos una interfaz de comunicación.

Los inventores han detectado que limitando la cantidad de nodos de red en el trayecto de un mensaje de datos que almacenan en caché un elemento recibido en el mensaje de datos, se consigue una utilización más eficiente de las cachés en una red centrada en información. Esto se consigue permitiendo que los nodos de red pujen para almacenar el elemento en su caché, compitiendo de ese modo contra otros nodos de red a lo largo del trayecto o contra otros nodos de red en el mismo grupo de nodos de red a lo largo del trayecto. En otras palabras, se subasta el derecho para almacenar en caché un elemento. Esto reduce la probabilidad de que los elementos menos populares no se almacenen en caché en absoluto. Esto puede conseguir asimismo una difusión optimizada del contenido a través de la red (menos peticiones al servidor), aumentando de ese modo su robustez con respecto a generadores de contenido que quedan fuera de línea. Además, esto puede reducir la latencia experimentada por los nodos solicitantes, sin aumentar el tráfico de red. Más ventajas pueden ser, por ejemplo: almacenamiento en caché optimizado en base a componentes geográficos y/o de temporización de las solicitudes, toma en consideración de las propiedades individuales de los nodos, coordinación entre nodos en almacenamiento en caché con sobrecarga mínima.

Si dicha por lo menos una interfaz de comunicación comprende múltiples interfaces de comunicación, el mensaje de interés es recibido preferentemente sobre otra interfaz de comunicación diferente a aquella en la que se reenvía. El mensaje de interés es reenviado de acuerdo con las reglas de reenvío del nodo. El mensaje de datos es recibido normalmente desde el nodo de más arriba sobre la misma interfaz de comunicación sobre la que el mensaje de

interés fue reenviado al nodo de más arriba. Si dicha por lo menos una interfaz de comunicación comprende una única interfaz, por ejemplo una interfaz inalámbrica, el nodo de red puede reenviar el mensaje de interés sobre la misma interfaz de comunicación desde la que recibió el mensaje de interés.

5 El por lo menos un procesador puede calcular el valor de puja antes de determinar si pujar para almacenar el objeto de datos solicitado en dicha por lo menos una memoria. Por ejemplo, el por lo menos un procesador puede determinar pujar solamente cuando el valor de puja calculado es mayor o menor que un determinado umbral o queda dentro de un determinado intervalo. Alternativamente, el por lo menos un procesador puede calcular el valor de puja después de decidir pujar para almacenar el objeto de datos solicitado en dicha por lo menos una memoria. Por ejemplo, el por lo menos un procesador puede calcular el valor de puja solamente cuando dicha por lo menos una memoria tiene una posición vacía o por lo menos uno de los objetos de datos ya almacenados en caché es más antiguo que una determinada antigüedad.

15 El valor de puja se puede calcular en base a uno o varios parámetros. Un parámetro puede estar relacionado con la situación local de un nodo, tal como el tamaño de caché o el ancho de banda del enlace hacia arriba. Adicionalmente, un parámetro puede capturar el "contexto" de dicha solicitud particular, que está relacionado con el contenido y/o la red. Por ejemplo, un parámetro puede estar relacionado con el contenido, por ejemplo la popularidad de un elemento o la latencia deseada para un elemento, o relacionado con la red, por ejemplo la cantidad de saltos desde el originador del mensaje de interés hasta el nodo de red o la latencia de red experimentada por el mensaje de interés. Un nodo de más abajo es un nodo que transmite un mensaje de interés al nodo de red y recibe un mensaje de datos desde el nodo de red, es decir, un nodo en el trayecto de interés/datos que está situado más próximo al solicitante de datos original que el nodo actual. Un nodo de más arriba es un nodo que recibe un mensaje de interés desde el nodo de red y transmite un mensaje de datos al nodo de red, es decir, un nodo en el trayecto de interés/datos que está situado más lejos del solicitante de datos original que el nodo actual.

20 Dicho mensaje de interés recibido puede comprender además un valor de puja recibido y dicho por lo menos un procesador puede estar configurado para incluir dicho valor de puja calculado en dicho mensaje de interés si dicha determinación sobre si pujar fue positiva y dicho valor de puja calculado representa una puja superior a dicho valor de puja recibido. En una primera realización, un valor de puja superior representa una puja superior, y en una segunda realización, un valor de puja inferior representa una puja superior. Si el nodo de red no es el primer nodo según la invención que recibe el mensaje de interés, entonces el mensaje de interés comprenderá ya un valor de puja cuando el mensaje de interés es recibido por el nodo de red. En una realización, si el valor de puja calculado no representa una puja superior al valor de puja recibido, el nodo de red pierde la puja. En una realización alternativa, si el valor de puja calculado no representa una puja superior y no es el mismo que el valor de puja recibido, el nodo de red pierde la puja. Si el nodo de red pierde la puja, no modifica el valor de puja recibido en el mensaje de interés recibido antes de reenviar el mensaje de interés al nodo de más arriba.

25 Dicho por lo menos un procesador puede estar configurado además para registrar información de puja en dicha por lo menos una memoria, representando dicha información de puja si dicho nodo de red incluyó dicho valor de puja calculado, en dicho mensaje de interés. Esta información de puja, por ejemplo un valor de verdadero o falso, se almacena antes de que el nodo de red reenvíe el mensaje de interés y se puede utilizar posteriormente cuando el nodo de red recibe el mensaje de datos, por ejemplo, tal como se describe en los siguientes dos párrafos.

30 Dicho por lo menos un procesador puede estar configurado para determinar a partir de por lo menos dicho mensaje de datos y dicha información de puja si dicho nodo de red ganó dicha puja. La información de puja registrada es beneficiosa cuando el nodo de red no puede determinar a partir de solamente el mensaje de datos si ha ganado la puja. Por ejemplo, el nodo de red puede ser capaz de determinar a partir del mensaje de datos si ha ganado la puja, solamente si se cumple la condición de que tenía la puja más alta en el momento de reenviar el mensaje de interés, que se indica mediante la información de puja y no puede ser determinada a partir del mensaje de datos. Esto se explica en mayor detalle en el párrafo siguiente.

35 En una realización alternativa, el nodo de red puede determinar a partir de solamente el mensaje de datos si ha ganado la puja, por ejemplo cuando el mensaje de datos comprende un identificador único (por ejemplo, dirección IP o dirección MAC) del nodo de red que ganó la puja, u opcionalmente múltiples identificadores únicos si el algoritmo de subasta permite que tengan lugar múltiples subastas dentro de un trayecto de interés. En otra realización alternativa, cuando no es probable que los valores de puja sean idénticos, el propio valor de puja calculado se almacena en dicha por lo menos una memoria si el nodo de red incluyó su valor de puja calculado en el mensaje de interés y el valor o valores de puja ganadores están incluidos en el mensaje de datos. Esto permite que el nodo de red determine si ha ganado la puja comparando el valor de puja almacenado con el valor o valores de puja ganadores en el mensaje de datos recibido.

40 Dicho mensaje de datos puede comprender un contador de ignorar, representando dicho contador de ignorar la cantidad de nodos de más abajo que han perdido dicha puja, determinada por dicho nodo de más arriba o por otro nodo de más arriba, y dicho por lo menos un procesador puede estar configurado para determinar a partir de dicho contador de ignorar y dicha información de puja si dicho nodo de red ha ganado dicha puja. Un nodo de red puede determinar si tiene la puja máxima hasta ahora, pero no si cualesquiera nodos de red que todavía esperan recibir el mensaje de interés van a calcular un valor de puja superior. Si un nodo de red más adelante en el trayecto del

mensaje de interés calcula un valor de puja superior, este sabe que los nodos anteriores a lo largo del trayecto o los nodos anteriores en el mismo grupo de nodos de red a lo largo del trayecto no ganaron la puja y puede ajustar el contador de ignorar del mensaje de datos de vuelta a la cantidad de nodos anteriores que no ganaron la puja. Un nodo de red que ha incluido un valor de puja calculado en el mensaje de interés puede a continuación determinar que no ha ganado la puja cuando el contador de ignorar no es cero.

Dicho por lo menos un procesador puede estar configurado además para incrementar un contador de nodos en dicho mensaje de interés antes de reenviar dicho mensaje de interés a dicho nodo de más arriba, representando dicho contador de nodos una cantidad de nodos de más abajo. Esto permite que un nodo de red más adelante en el trayecto del mensaje de interés que calcula un valor de puja superior determine la cantidad de nodos anteriores que no ganaron la puja y almacene esta cantidad en su por lo menos una memoria si incluye su valor de puja en el mensaje de interés que reenvía. El contador de nodos se puede incrementar aumentando un valor con respecto al número de referencia, por ejemplo 0, 1, 2, 3, etc., o disminuyendo un valor con respecto a un número de referencia, por ejemplo 1000, 999, 998, etc., por ejemplo. El nodo de red puede restablecer el contador de nodos cuando es el primero o el último nodo en un grupo de nodos de red que participan en una subasta, por ejemplo cuando el contador de nodos alcanza o excede un umbral.

Dicho por lo menos un procesador puede estar configurado además para registrar un cómputo de pujadores perdedores en dicha por lo menos una memoria si dicho nodo de red incluyó dicho valor de puja calculado en dicho mensaje de interés, basándose dicho cómputo de pujadores perdedores en dicho contador de nodos incluido en dicho mensaje de interés recibido y representando la cantidad de nodos de más abajo que han perdido dicha puja, determinada por dicho nodo de red. El cómputo de pujadores perdedores se puede utilizar para ajustar el contador de ignorar cuando el nodo de red recibe el mensaje de datos y determina que ha ganado la puja, por ejemplo. Dicho nodo de red puede aumentar dicho contador de nodos incluido en dicho mensaje de interés recibido, almacenar dicho contador de nodos aumentado como dicho cómputo de pujadores perdedores y reenviar dicho mensaje de interés incluyendo dicho contador de nodos aumentado, por ejemplo. Alternativamente, dicho nodo de red puede almacenar dicho contador de nodos recibido como dicho cómputo de pujadores perdedores, aumentar dicho nodo incluido en dicho mensaje de interés recibido y reenviar dicho mensaje de interés que incluye dicho contador de nodos aumentado, por ejemplo.

Dicho por lo menos un procesador puede estar configurado para ajustar dicho contador de ignorar en dicho mensaje de datos en base a dicho cómputo de pujadores perdedores antes de reenviar dicho mensaje de datos a dicho nodo de más abajo si dicho nodo de red ganó dicha puja. El contador de ignorar puede ajustar el contador de nodos que fue almacenado en el cómputo de pujadores perdedores cuando se recibió el mensaje de interés, por ejemplo. Esto permite que el nodo de red utilice el mensaje de datos para informar a los nodos anteriores a lo largo del trayecto del mensaje de interés o a los nodos anteriores en el mismo grupo de nodos de red a lo largo del trayecto del mensaje de interés, que todavía esperan recibir el mensaje de datos, de que no han ganado la puja, por ejemplo.

De acuerdo con la invención, el segundo objetivo se consigue porque el procedimiento de recepción de un mensaje de interés comprende las etapas de recibir un mensaje de interés sobre por lo menos una interfaz de comunicación desde un nodo de más abajo, comprendiendo dicho mensaje de interés un nombre de un objeto de datos solicitado, determinar si pujar para almacenar dicho objeto de datos solicitado, en por lo menos una memoria, calcular un valor de puja, reenviar dicho mensaje de interés a un nodo de más arriba sobre dicha por lo menos una interfaz de comunicación (o sobre una interfaz de comunicación diferente, si la hay, de acuerdo con las reglas de reenvío del nodo), incluyendo dicho mensaje de interés dicho valor de puja calculado en función de, por lo menos, dicha determinación de si pujar, recibir un mensaje de datos que comprende dicho objeto de datos solicitado desde dicho nodo de más arriba sobre dicha por lo menos una interfaz de comunicación, determinar a partir de dicho mensaje de datos si dicho nodo de red ganó dicha puja, almacenar dicho objeto de datos en dicha por lo menos una memoria si dicho nodo de red ganó dicha puja, y reenviar dicho mensaje de datos a dicho nodo de más abajo sobre dicha por lo menos una interfaz de comunicación. Dicho método puede ser realizado mediante software ejecutado en un dispositivo programable. Este software se puede proporcionar como un producto de programa informático.

Dicho mensaje de interés recibido puede comprender además un valor de puja recibido y dicho procedimiento puede comprender además la etapa de incluir dicho valor de puja calculado en dicho mensaje de interés si dicha determinación de si pujar fue positiva y dicho valor de puja calculado representa una puja mayor que dicho valor de puja recibido.

Dicho procedimiento puede comprender además registrar información de puja en dicha por lo menos una memoria, representando dicha información de puja si dicho nodo de red ha incluido dicho valor de puja calculado, en dicho mensaje de interés. Dicha etapa de determinar a partir de por lo menos dicho mensaje de datos si dicho nodo de red ganó dicha puja puede comprender determinar a partir de por lo menos dicho mensaje de datos y dicha información de puja si dicho nodo de red ganó dicha puja. Dicho mensaje de datos recibido puede comprender un contador de ignorar, representando dicho contador de ignorar una cantidad de nodos de más abajo que perdieron dicha puja, determinada mediante dicho nodo de más arriba u otro nodo de más arriba, y dicha etapa de determinar a partir de dicho mensaje de datos si dicho nodo de red ganó dicha puja puede comprender determinar a partir de por lo menos dicho contador de ignorar y dicha información de puja si dicho nodo de red ganó dicha puja.

- Dicho procedimiento puede comprender además incrementar un contador de nodos en dicho mensaje de interés antes de reenviar dicho mensaje de interés a dicho nodo de más arriba, representando dicho contador de nodos una cantidad de nodos de más abajo. Dicho procedimiento puede comprender además registrar un cómputo de pujadores perdedores en dicha por lo menos una memoria si dicho nodo de red incluyó dicho valor de puja calculado, en dicho mensaje de interés, estando basado dicho cómputo de pujadores perdedores en dicho contador de nodos incluido en dicho mensaje de interés recibido y representando la cantidad de nodos de más abajo que perdieron dicha puja, determinada por dicho nodo de red. Dicho procedimiento puede comprender además ajustar dicho contador de ignorar en dicho mensaje de datos en base a dicho cómputo de pujadores perdedores antes de reenviar dicho mensaje de datos a dicho nodo de más abajo si dicho nodo de red ganó dicha puja.
- De acuerdo con la invención, el tercer objetivo se consigue porque el nodo terminal comprende por lo menos una interfaz de comunicación y por lo menos un procesador configurado para determinar un nombre de un objeto de datos solicitado, crear un mensaje de interés que comprende dicho nombre de dicho objeto de datos solicitado y un campo de puja para introducir un valor de puja, dejándose sin definir dicho valor de puja para indicar que debería comenzar una subasta del derecho a almacenar en caché dicho objeto de datos solicitado, y transmitir dicho mensaje de interés a un nodo de más arriba sobre dicha por lo menos una interfaz de comunicación.
- Además, se da a conocer un programa informático para llevar a cabo los procedimientos descritos en la presente memoria, así como un medio de almacenamiento no transitorio legible por ordenador que almacena el programa informático. Un programa informático puede, por ejemplo, descargarse o subirse a un dispositivo existente o almacenarse con la fabricación de estos sistemas.
- Un medio de almacenamiento no transitorio legible por ordenador almacena por lo menos una parte de código de software, estando configurada la parte de código de software, cuando es ejecutada o procesada por un ordenador, para realizar operaciones ejecutables que comprenden: recibir un mensaje de interés sobre por lo menos una interfaz de comunicación desde un nodo de más abajo, comprendiendo dicho mensaje de interés un nombre de un objeto de datos solicitado, determinar si pujar para almacenar dicho objeto de datos solicitado, en por lo menos una memoria, calcular un valor de puja, reenviar dicho mensaje de interés a un nodo de más arriba sobre dicha por lo menos una interfaz de comunicación, incluyendo dicho mensaje de interés dicho valor de puja calculado en función de, por lo menos, dicha determinación de si pujar, recibir un mensaje de datos que comprende dicho objeto de datos solicitado desde dicho nodo de más arriba sobre dicha por lo menos una interfaz de comunicación, determinar a partir de por lo menos dicho mensaje de datos si dicho nodo de red ganó dicha puja, almacenar dicho objeto de datos en dicha por lo menos una memoria si dicho nodo de red ganó dicha puja, y reenviar dicho mensaje de datos a dicho nodo de más abajo sobre dicha por lo menos una interfaz de comunicación.
- Tal como apreciará un experto en la materia, los aspectos de la presente invención se pueden realizar como un dispositivo, un procedimiento o un producto de programa informático. Por consiguiente, los aspectos de la presente invención pueden adoptar la forma de una realización íntegramente de hardware, una realización íntegramente de software (incluyendo software inalterable, software residente, microcódigo, etc.) o una realización que combina aspectos de software y de hardware que pueden todos denominarse en general en la presente memoria un "circuito", "módulo" o "sistema". Las funciones descritas en esta invención se pueden implementar como un algoritmo ejecutado por un procesador/microprocesador de un ordenador. Además, los aspectos de la presente invención pueden adoptar la forma de un producto de programa informático realizado en uno o varios medios legibles por ordenador que tienen incorporado código de programa legible por ordenador, por ejemplo, almacenado, en los mismos.
- Se puede utilizar cualquier combinación de uno o varios medios legibles por ordenador. El medio legible por ordenador puede ser un medio de señal legible por ordenador o un medio de almacenamiento legible por ordenador. Un medio de almacenamiento legible por ordenador puede ser, por ejemplo, de forma no limitativa, un sistema, aparato, o dispositivo, electrónico, magnético, óptico, electromagnético, de infrarrojos o semiconductor, o cualquier combinación adecuada de los anteriores. Más ejemplos específicos de un medio de almacenamiento legible por ordenador pueden incluir, de forma no limitativa, los siguientes: una conexión eléctrica que tiene uno o varios cables, un disquete informático portátil, un disco duro, una memoria de acceso aleatorio (RAM, random access memory), una memoria de sólo lectura (ROM, read-only memory), una memoria de sólo lectura programable borrable (EPROM o memoria flash), una fibra óptica, una memoria de sólo lectura de disco compacto (CD-ROM) portátil, un dispositivo de almacenamiento óptico, un dispositivo de almacenamiento magnético o cualquier combinación adecuada de los anteriores. En el contexto de la presente invención, un medio de almacenamiento legible por ordenador puede ser cualquier medio tangible que pueda contener, o almacenar, un programa para su utilización por, o en conexión con un sistema, aparato o dispositivo de ejecución de instrucciones.
- Un medio de señal legible por ordenador puede incluir una señal de datos propagada con código de programa legible por ordenador incorporado en la misma, por ejemplo, en banda base o como parte de una onda portadora. Dicha señal propagada puede adoptar cualquiera de diversas formas incluyendo, de forma no limitativa, electromagnética, óptica o cualquier combinación adecuada de las mismas. Un medio de señal legible por ordenador puede ser cualquier medio legible por ordenador que no sea un medio de almacenamiento legible por ordenador y que pueda comunicar, propagar o transportar un programa para ser utilizado por, o en conexión con un sistema, aparato o dispositivo de ejecución.

El código de programa incorporado en un medio legible por ordenador puede ser transmitido utilizando cualquier medio apropiado incluyendo, de forma no limitativa, inalámbrico, cableado, fibra óptica, cable, RF, etc., o cualquier combinación adecuada de los anteriores. El código de programa informático para llevar a cabo operaciones para aspectos de la presente invención puede escribirse en cualquier combinación de uno o varios lenguajes de programación, incluyendo un lenguaje de programación orientado a objetos, tal como Java(TM), Smalltalk, C++ o similar y lenguajes de programación por procedimientos, tales como el lenguaje de programación "C" o lenguajes de programación similares. El código de programa se puede ejecutar íntegramente en el ordenador del usuario, parcialmente en el ordenador del usuario, como un paquete de software independiente, parcialmente en el ordenador del usuario y parcialmente en un ordenador remoto, o íntegramente en el servidor o el ordenador remoto. En el último escenario, el ordenador remoto se puede conectar al ordenador del usuario a través de cualquier tipo de red, incluyendo una red de área local (LAN, local area network) o una red de área extensa (WAN, wide area network), o la conexión se puede realizar a un ordenador externo (por ejemplo, a través de internet utilizando un proveedor de servicios de internet).

A continuación se describen aspectos de la presente invención haciendo referencia a ilustraciones de diagramas de flujo y/o diagramas de bloques de procedimientos, aparatos (sistemas) y productos de programa informático de acuerdo con realizaciones de la presente invención. Se comprenderá que cada bloque de las ilustraciones de diagramas de flujo y/o diagramas de bloques, y combinaciones de bloques en las ilustraciones de diagramas de flujo y/o de diagramas de bloques, se pueden implementar mediante instrucciones de programa informático. Estas instrucciones de programa informático pueden ser proporcionadas a un procesador, en particular un microprocesador o una unidad central de proceso (CPU, central processing unit), de un ordenador de propósito general, un ordenador de propósito especial u otro aparato programable de procesamiento de datos, para producir una máquina, de tal modo que las instrucciones, que se ejecutan por medio del procesador del ordenador, de otro aparato de procesamiento de datos programable o de otros dispositivos, crean medios para implementar las funciones/acciones especificadas en el bloque o bloques del diagrama de flujo y/o del diagrama de bloques.

Estas instrucciones de programa informático pueden asimismo almacenarse en un medio legible por ordenador que puede dirigir un ordenador, otro aparato programable de procesamiento de datos u otros dispositivos, para que funcionen de manera particular, de tal modo que las instrucciones almacenadas en el medio legible por ordenador producen un artículo de fabricación que incluye instrucciones que implementan la función/acción especificada en el bloque o bloques del diagrama de flujo y/o del diagrama de bloques.

Las instrucciones de programa informático pueden asimismo cargarse en un ordenador, otro aparato programable de procesamiento de datos u otros dispositivos, para hacer que se lleven a cabo una serie de etapas operativas en el ordenador, otro aparato programable u otros dispositivos, con el fin de producir un proceso implementado por ordenador, de tal modo que las instrucciones que se ejecutan en el ordenador u otro aparato programable proporcionan procesos para implementar las funciones/acciones especificadas en el bloque o bloques del diagrama de flujo y/o del diagrama de bloques.

El diagrama de flujo y los diagramas de bloques de las figuras muestran la arquitectura, la funcionalidad y el funcionamiento de posibles implementaciones de dispositivos, procedimientos y productos de programa informático, de acuerdo con varias realizaciones de la presente invención. A este respecto, cada bloque en el diagrama de flujo o los diagramas de bloques puede representar un módulo, segmento, o una parte de código, que comprende una o varias instrucciones ejecutables para implementar la función o funciones lógicas especificadas. Se debe observar asimismo que, en algunas implementaciones alternativas, las funciones indicadas en los bloques se pueden producir fuera del orden indicado en las figuras. Por ejemplo, dos bloques mostrados en sucesión pueden, de hecho, ser ejecutados de manera sustancialmente simultánea, o los bloques pueden en ocasiones ejecutarse en el orden inverso, dependiendo de la funcionalidad involucrada. Se debe observar asimismo que cada bloque de las ilustraciones de diagramas de bloques y/o de diagramas de flujo, y combinaciones de bloques en las ilustraciones de diagramas de bloques y/o de diagramas de flujo, se pueden implementar mediante sistemas basados en hardware de propósito especial que llevan a cabo las funciones o acciones especificadas, o combinaciones de hardware de propósito especial e instrucciones informáticas.

Breve descripción de los dibujos

Estos y otros aspectos de la invención son evidentes a partir de los dibujos, y se aclararán en mayor medida, a modo de ejemplo, haciendo referencia a los mismos, en los que:

la figura 1 es un diagrama de bloques de una realización del nodo de red de la invención;

la figura 2 es un diagrama de flujo de una realización del procedimiento de la invención;

la figura 3 muestra un ejemplo de un trayecto de red que incluye múltiples nodos;

la figura 4 es un diagrama de flujo que muestra un proceso realizado a continuación de la recepción de un mensaje de interés en una realización del procedimiento de la invención;

la figura 5 es un diagrama de flujo que muestra detalles de una de las etapas de la figura 4;

la figura 6 es un diagrama de flujo que muestra detalles de una de las etapas de la figura 4;

la figura 7 es un diagrama de flujo que muestra detalles de una de las etapas de la figura 4;

la figura 8 es un diagrama de flujo que muestra un proceso realizado a continuación de la recepción de un mensaje de datos en una realización del procedimiento de la invención;

5 la figura 9 es un diagrama de flujo que muestra detalles de una de las etapas de la figura 8;

la figura 10 es un diagrama de flujo que muestra detalles de una de las etapas de la figura 8;

la figura 11 muestra un ejemplo de campos y valores introducidos en el mensaje de interés, el mensaje de datos y la PIT de la invención;

10 la figura 12 es un diagrama de bloques de un sistema de procesamiento de datos a modo de ejemplo, para realizar el procedimiento de la invención; y

la figura 13 es un diagrama de bloques del nodo terminal de la invención.

Los elementos correspondientes en los dibujos se indican mediante el mismo numeral de referencia.

Descripción detallada de los dibujos

15 La invención se refiere a un nodo de red que comprende por lo menos una interfaz de comunicación, por lo menos una memoria y por lo menos un procesador. En la realización mostrada en la figura 1, el nodo de red 1₅ comprende dos interfaces de comunicaciones 3₁₋₂, una memoria 5 y un procesador 7. El procesador 7 está configurado para recibir un mensaje de interés sobre la interfaz de comunicación 3₁ desde un nodo de más abajo 1₄. El mensaje de interés comprende un nombre de un objeto de datos solicitado. El procesador 7 está configurado además para determinar si pujar para almacenar el objeto de datos solicitado en la memoria 5. El procesador 7 está configurado además para calcular un valor de puja. El procesador 7 está configurado además para reenviar el mensaje de interés a un nodo de más arriba 1₆ sobre la interfaz de comunicación 3₂. El mensaje de interés incluye el valor de puja calculado en función de, por lo menos, la determinación de si pujar. El procesador 7 está configurado además para recibir un mensaje de datos que comprende el objeto de datos solicitado, desde el nodo de más arriba 1₆ sobre la interfaz de comunicación 3₂. El procesador 7 está configurado además para determinar a partir de por lo menos el mensaje de datos si el nodo de red ganó la puja. El procesador 7 está configurado además para almacenar el objeto de datos en la memoria 5 si el nodo de red ganó la puja. El procesador 7 está configurado además para reenviar el mensaje de datos al nodo de más abajo 1₄ sobre la interfaz de comunicación 3₁. El nodo de red 1₅ se denomina asimismo un "nodo de almacenamiento caché" en esta descripción (para distinguirlo de otros nodos en el trayecto de interés/datos que pueden no estar configurados para almacenar elementos de datos y pueden no estar configurados para cambiar mensajes de interés y de datos).

20 El nodo de red 1₅ puede ser un encaminador de red dedicado, por ejemplo, pero puede ser asimismo, por ejemplo, un dispositivo de usuario (con o sin una interfaz de usuario), por ejemplo, un teléfono móvil o un dispositivo incorporado en un vehículo. El por lo menos un procesador puede comprender uno o varios procesadores de propósito general y/o uno o varios procesadores específicos de aplicación. El procesador 7 puede ser un procesador ARM, por ejemplo. Dicha por lo menos una memoria puede comprender uno o varios discos duros magnéticos, una o varias unidades de disco y/o una o varias memorias de estado sólido, por ejemplo. El objeto u objetos de datos pueden almacenarse en la misma memoria o en una memoria diferente a la de la otra información, tal como una tabla de intereses pendientes (PIT) o una configuración del nodo de red. Dicha por lo menos una interfaz de comunicación puede comprender uno o varios puertos ópticos y/o uno o varios puertos de Ethernet, por ejemplo.

35 El procedimiento de recibir un mensaje de interés de la invención comprende por lo menos ocho etapas, ver la figura 2. Una etapa 11 comprende recibir un mensaje de interés sobre por lo menos una interfaz de comunicación desde un nodo de más abajo, comprendiendo el mensaje de interés un nombre de un objeto de datos solicitado. Una etapa 13 comprende determinar si pujar para almacenar el objeto de datos solicitado en por lo menos una memoria. Una etapa 15 comprende calcular un valor de puja. La etapa 15 se puede llevar a cabo después o antes de la etapa 13, por ejemplo. Cuando se determina no pujar en la etapa 13, se puede saltar la etapa 15. El valor de puja calculado en la etapa 15 puede ser utilizado en la etapa 13 para determinar si pujar o no.

40 Una etapa 17 comprende reenviar el mensaje de interés a un nodo de más arriba sobre dicha por lo menos una interfaz de comunicación, incluyendo el mensaje de interés el valor de puja calculado en función de, por lo menos, la determinación sobre si pujar. Una etapa 19 comprende recibir un mensaje de datos que comprende el objeto de datos solicitado procedente del nodo de más arriba sobre dicha por lo menos una interfaz de comunicación. Una etapa 21 comprende determinar a partir de por lo menos el mensaje de datos si el nodo de red ganó la puja. Una etapa 23 comprende almacenar el objeto de datos en la por lo menos una memoria si el nodo de red ganó la puja. Una etapa 25 comprende reenviar el mensaje de datos al nodo de más abajo sobre dicha por lo menos una interfaz de comunicación.

En otras palabras, los nodos de almacenamiento caché situados en el trayecto a cuyo través viaja el mensaje de interés participan en una subasta para ganar el derecho a almacenar en caché los datos que se devolverán en respuesta a dicho mensaje de interés (mediante un mensaje de datos). Realizar una subasta sobre el derecho para almacenar en caché permite que los nodos de almacenamiento caché sobre el trayecto de la solicitud se coordinen acerca de cuál de ellos va a almacenar una copia del objeto de datos, evitando de ese modo un almacenamiento en caché redundante debido a decisiones individuales. Para limitar la sobrecarga de elegir y señalar al ganador de la subasta, la subasta no se realiza centralmente sino por el contrario de manera distribuida: el valor de puja más alto actualmente se incorpora al mensaje de interés y cada nodo de almacenamiento caché decide si pujar o no en base al valor de puja en el mensaje de interés y a la propia puja que la caché del mismo puede emitir (el valor de la puja que un nodo de almacenamiento caché puede emitir puede estar basado en parámetros locales del nodo, así como en el contexto de la solicitud).

Si un nodo de almacenamiento caché decide pujar, sustituirá la actual máxima puja en el mensaje de interés con su propio valor de puja antes de reenviar el mensaje de interés al siguiente nodo. La decisión final sobre qué nodo ha ganado la subasta se comunica a los nodos de almacenamiento caché participantes incorporándola al mensaje de datos de vuelta. Este enfoque permite que los nodos de almacenamiento caché colaboren en la decisión acerca de cuál de ellos va a almacenar en caché el elemento, así como que (opcionalmente) enriquezcan la información que conduce a esta decisión (con el contexto de la solicitud), sin la necesidad de señalización adicional o sustancial.

Para la selección del valor para pujar, un nodo de almacenamiento caché puede utilizar tanto información local como el contexto de la solicitud. Los parámetros que describen el contexto de la solicitud se pueden incorporar al mensaje de interés para minimizar la sobrecarga o se pueden incluir en mensajes especiales, por ejemplo.

Para controlar la redundancia de los datos, múltiples nodos de almacenamiento caché pueden almacenar en caché el mismo objeto de datos. Esto se puede conseguir agrupando los nodos de almacenamiento caché sobre el trayecto de la solicitud en grupos diferentes y realizando una subasta independiente para cada grupo. En la figura 3, se representan once nodos de los que nueve son nodos de almacenamiento caché 1₁-1₉, uno es un dispositivo de usuario 31 en el que se origina a una solicitud/interés y uno es un servidor 35 en el que se origina un objeto de datos. Los nodos de almacenamiento caché 1₁-1₉ están agrupados en tres grupos 33₁₋₃. Un determinado objeto de datos se almacena en caché en el nodo de almacenamiento caché 1₅ del grupo 33₂ y el nodo de almacenamiento caché 1₉ del grupo 33₃. Las decisiones relativas a redundancia de datos se pueden basar en una combinación de la información local y el contexto de la solicitud, o pueden ser decididas por el administrador de red en base a estadísticas de la red, por ejemplo.

El objeto de datos determinado no se almacena en caché en el grupo 33₁, debido a que los nodos de almacenamiento caché de este grupo contienen ya contenido muy popular (por ejemplo, debido a que estos nodos están situados cerca de otro dispositivo de usuario) y como resultado ninguno de los nodos de almacenamiento caché del grupo 33₁ ha participado en la subasta. Estos nodos de almacenamiento caché han utilizado un umbral de subasta para determinar si participar o no en la subasta: si un nodo de almacenamiento caché tiene datos que son más antiguos que el umbral de subasta, se cualifica para tomar parte en una subasta. El valor para el umbral de subasta puede estar basado en una combinación de información (si está disponible) local y global o puede ser decidido por el administrador de red en base a estadísticas de la red. El trayecto de encaminamiento representado en la figura 3 se ha determinado en base al nombre del objeto de datos solicitado.

Un nodo de almacenamiento caché puede tener en cuenta uno o varios parámetros cuando determina si pujar, y cuando calcula un valor de puja. Siguen ejemplos de aspectos locales que un nodo de almacenamiento caché podría tener en cuenta.

LRU: la marca de tiempo del elemento utilizado menos recientemente. El valor LRU debería ser ponderado adecuadamente para evitar casos en los que las solicitudes de elementos podrían ser el resultado de aplicaciones en tiempo real (tal una salida de sensor) cuyos valores antiguos no son de utilidad en el futuro, y podrían tener como resultado un desperdicio de los recursos de almacenamiento en caché:

- **LFU**: el cómputo de solicitudes del elemento utilizado menos frecuentemente. LRU y LFU reflejan la popularidad de los datos almacenados en caché, según la percibe el nodo.
- **BWIN**: el ancho de banda del enlace de datos entrante. En caso de que el ancho de banda del enlace de datos entrantes sea pequeño, un nodo de almacenamiento caché desearía pujar más alto para evitar más tráfico en enlaces menores. BWIN refleja las condiciones actuales de la red en torno al nodo de almacenamiento caché. Adicional o alternativamente, se podría tener en cuenta el ancho de banda del enlace de datos salientes.
- **CS**: el tamaño de caché del nodo. CS refleja la capacidad de almacenamiento en caché de la caché del nodo.
- **RC**: la velocidad de almacenamiento en caché (es decir, el número de objetos almacenados en caché por unidad de tiempo). La fracción de datos entrantes frente al tamaño de la caché proporcionaría una estimación del rendimiento del nodo de almacenamiento caché. RC refleja cómo de rápido cambian actualmente los datos en la caché del nodo. Por lo tanto, se podría asignar una ponderación a CS y/o RC que ayudaría a los nodos de almacenamiento caché más importantes a pujar más fuerte.

- **INC:** el cómputo de las veces que el mismo mensaje de interés ha llegado al nodo pero no se ha almacenado en caché. Se podría mantener un cómputo temporal de los intereses que se han solicitado anteriormente pero nunca se han almacenado en caché. Este cómputo ayuda en la búsqueda de elementos de datos muy populares, a los que se asignarían entonces ponderaciones mayores durante el proceso de puja. INC refleja la popularidad de los datos recién recibidos.

Además, existen varios parámetros no locales ("contexto de la solicitud") que un nodo de almacenamiento caché podría tener en cuenta:

- **HOP:** distancia (en saltos) desde el solicitante. La información sobre el número de saltos que separan un nodo de almacenamiento caché del usuario (dispositivo) solicitante podría ayudar a asignar una baja probabilidad de almacenamiento en caché para nodos que están más lejos del usuario (**dispositivo**), y **viceversa**.
- **L:** latencia del mensaje de interés.

Se podría calcular un valor de puja, por ejemplo, determinando una suma ponderada de uno o varios de los anteriores parámetros locales y de contexto, donde cada ponderación proporciona más o menos énfasis a cada uno de los aspectos. Las ponderaciones podrían, por ejemplo, depender de la red, o ser ajustadas a conveniencia por el administrador de la red.

Por ejemplo, una puja podría ser calculada por tres nodos en competencia (R1-R3) con la ecuación siguiente:

$$Bid(R_x) = \left[w_1 * \arctan\left(\frac{LRU}{LRU_{MAX}/\alpha_1}\right) * \frac{2}{\pi} \right] + \left[w_2 * \arctan\left(\frac{1/LFU}{1/LFU_{MAX}/\alpha_2}\right) * \frac{2}{\pi} \right] + \left[w_3 * \arctan\left(\frac{CS}{CS_{MAX}/\alpha_3}\right) * \frac{2}{\pi} \right] \dots \quad \text{(Ecuación 1)}$$

Para este ejemplo, se supone que los tres nodos decidirán su puja en base a tres parámetros, en concreto el valor utilizado menos recientemente (LRU), el valor utilizado menos frecuentemente (LFU) y el tamaño de caché (CS) de cada nodo. La utilización de la ecuación para cualquier otra clase de parámetro se puede extrapolar a partir de este ejemplo. Los nodos no intercambian información sobre su visión de la red y por lo tanto pueden tener diferentes ajustes en relación, por ejemplo, con el rango observable de cada uno de los parámetros. Por ejemplo, desde el punto de vista de R1, el rango máximo observado para el valor de LRU puede ser de 1 segundo, mientras que desde el punto de vista de R3, el rango máximo observado para el valor de LRU puede ser de 10 segundos.

Para este ejemplo, se supone que los nodos de almacenamiento caché tienen los siguientes valores observados/estimados para cada uno de los parámetros utilizados en la ecuación de puja.

Tabla 1

	Nodo R1	Nodo R2	Nodo R3
LRU (ms)	10	200	1000
LRU Máx. del rango	1000	1000	1000
LFU (veces/segundo)	50	10	10
LFU Máx. del rango	100	100	20
CS (MB)	200	500	500
CS Máx. del rango	1000	1000	1000
α₁	2	2	2
α₂	2	2	2
α₃	2	2	10
w₁	0,4	0,3	0,4
w₂	0,4	0,4	0,3
w₃	0,2	0,3	0,3

A continuación se proporcionan algunas aclaraciones sobre el significado de los valores presentados anteriormente:

- 5 • **LRU:** el objeto de datos menos utilizado recientemente almacenado por R1 fue el accedido más recientemente mientras que el objeto de datos menos utilizado recientemente almacenado por R3 fue accedido aproximadamente un segundo antes. Esto debería conducir a R3 a calcular una puja mayor que R1 y R2 con respecto a este parámetro (los objetos de datos a los que no se ha accedido durante un tiempo prolongado son preferentemente los primeros candidatos para sustitución). Resulta que todos los nodos tienen la misma perspectiva sobre el valor máximo de este parámetro.
- 10 • **LFU:** el objeto de datos utilizado menos frecuentemente almacenado por R1 es accedido con mucha mayor frecuencia que el objeto de datos utilizado menos frecuentemente almacenado por los otros nodos. Esto significa que R1 debería ofertar una puja baja con respecto al valor LFU. Dado que un valor elevado de este parámetro se debería traducir preferentemente en un bajo valor de puja, la inversa de este parámetro se utiliza en la ecuación 1, es decir $\left(\frac{1}{LFU}\right)$ o LFU^{-1} , tanto en el numerador como en el denominador (para el valor del rango). El nodo R3 tiene una visión diferente sobre cuál es el valor máximo de este parámetro, que conducirá a una puja diferente del nodo R2 para este parámetro, aunque de hecho tengan el mismo valor LFU.
- 15 • **CS:** el nodo R1 tiene una caché disponible menor comparada con los otros nodos, lo que conduce a valores de puja menores en relación con este parámetro respecto de R1 en la ecuación 1 (puede ser preferible un almacenamiento en caché menos frecuente en aquellas caché con una memoria pequeña). Dado que esto se puede ver asimismo como el punto débil de cada nodo, también es preferentemente el aspecto más importante que determinará si R1 almacenará o no en caché el objeto de datos. Por esta razón, la ponderación que se proporciona a este parámetro (w_3 de R1) es mucho menor que para los otros dos parámetros (LRU y LFU) con el fin de afectar negativamente a las oportunidades de que el objeto de datos sea almacenado en caché en el nodo con el mínimo tamaño de caché.
- 20 • **Parámetros de sensibilidad ($\alpha_1 - \alpha_3$):** estos valores determinan la sensibilidad de cada parámetro dentro de su rango, dado que definen la 'pendiente' de la función arcotangente. Se puede utilizar un valor por defecto en la mayor parte de los casos (por ejemplo, 2 en este ejemplo), pero si un parámetro se considera lo suficientemente importante para un nodo específico y se requiere una mayor granularidad de los valores de puja que quedan dentro del rango de dicho parámetro, entonces se puede aumentar el valor de sensibilidad de este parámetro (por ejemplo, α_3 para R3 aumenta a 10 en este ejemplo).
- 25 • **Ponderaciones de parámetros ($w_1 - w_3$):** estas ponderaciones determinan la contribución de cada uno de los parámetros al valor de puja final del nodo. Dependiendo de las ventajas o desventajas de un nodo, diferentes parámetros pueden ser más o menos importantes. Estos parámetros pueden ser ajustados por la aplicación o por el administrador de red, y pueden ser ajustables de manera semiestática. La suma de todas las ponderaciones para un nodo vale 1 en este ejemplo (en esta realización, el valor de puja es un valor entre 0 y 1).

35 Utilizando la información anterior, se puede resolver la ecuación para cada uno de los nodos para encontrar el valor de puja que utilizarán. Sustituyendo los valores mencionados en la tabla 1, se puede determinar el valor de puja de cada nodo.

Valor de puja de R1:

$$Bid(R_1) = \left[0.4 * \arctan\left(\frac{10}{1000/2}\right) * \frac{2}{\pi}\right] + \left[0.4 * \arctan\left(\frac{1/50}{1/(100/2)}\right) * \frac{2}{\pi}\right] + \left[0.2 * \arctan\left(\frac{200}{1000/2}\right) * \frac{2}{\pi}\right]$$

(Ecuación 2)

$$Bid(R_1) = 0.0051 + 0.2 + 0.0484$$

40 $Bid(R_1) = 0.2535$

Valor de puja de R2:

$$Bid(R_2) = \left[0.3 * \arctan\left(\frac{200}{1000/2}\right) * \frac{2}{\pi}\right] + \left[0.4 * \arctan\left(\frac{1/10}{1/(100/2)}\right) * \frac{2}{\pi}\right] + \left[0.3 * \arctan\left(\frac{500}{1000/2}\right) * \frac{2}{\pi}\right]$$

(Ecuación 3)

$$Bid(R_2) = 0.0727 + 0.3497 + 0.15$$

$$Bid(R_2) = 0.5724$$

Valor de puja de R3:

$$Bid(R_3) = \left[0.4 * \arctan\left(\frac{1000}{1000/2}\right) * \frac{2}{\pi} \right] + \left[0.3 * \arctan\left(\frac{1/10}{1/(20/2)}\right) * \frac{2}{\pi} \right] + \left[0.3 * \arctan\left(\frac{500}{1000/10}\right) * \frac{2}{\pi} \right]$$

(Ecuación 4)

$$Bid(R_3) = 0.2819 + 0.15 + 0.2623$$

$$Bid(R_3) = 0.6942$$

- 5 Dado que Bid(R3) > Bid(R2) > Bid(R1), el nodo R3 es el ganador de la puja/subasta, y va a almacenar en caché el objeto de datos cuando éste se devuelva.

10 Para cada grupo de subasta, puede haber 0, 1 o más ganadores de la puja/subasta, es decir, nodos de almacenamiento caché que ganan el derecho a almacenar en caché. Por ejemplo, si todos los nodos de almacenamiento caché en el grupo de subasta tienen un umbral de subasta que es un valor mayor que su valor LRU (lo que significa que los nodos ya están almacenando en caché datos muy populares/recientes) y no existen posiciones disponibles de almacenamiento en caché, ninguno de estos nodos va a pujar y por lo tanto no existe ningún ganador en este grupo de subasta, y ninguno de estos nodos va a almacenar en caché el objeto de datos solicitado en el mensaje de interés. Aunque preferentemente el mecanismo de subasta se implementa de manera que se permite que solamente uno de los nodos de almacenamiento caché en un grupo de subasta almacene finalmente en caché el elemento, el mecanismo de subasta podría alternativamente conceder el derecho a almacenar en caché a todos los nodos de almacenamiento caché que pujan con el mismo valor máximo en un grupo de subasta.

15 Para soportar la realización de una subasta, los mensajes de interés se pueden extender con los siguientes dos campos:

- 20 • **bid_win (valor de puja ganador):** este campo contiene la puja ganadora actual. Siempre que la puja del nodo actual sea mayor que este valor, el nodo actualizará este campo. Si la puja del nodo actual es menor o igual que el valor de este campo, entonces el campo permanece sin cambios. En una realización alternativa, el campo se actualiza asimismo si la puja del nodo actual es igual al valor de este campo.
- 25 • **bid_num (contador de nodos):** este campo mantiene un seguimiento del número de nodos de almacenamiento caché participantes. Por lo tanto, cada nodo de almacenamiento caché aumentará este campo en uno. Si se ha definido bid_exp (ver la información de configuración más abajo) y bid_num supera a bid_exp, entonces bid_num se restablecerá a cero y bid_win se ajustará a vacío. Esto marcará asimismo el final de la subasta anterior y el comienzo de una nueva, con un nuevo grupo de nodos de almacenamiento caché.

30 Adicionalmente, los mensajes de interés pueden transportar una lista de parámetros de "contexto de la solicitud", tales como los mencionados a continuación:

- **hop_count:** este campo mantiene un seguimiento de la distancia desde el solicitante. Cada encaminador aumentará este campo en uno.
- **latencia:** éste campo mantiene un seguimiento de la latencia experimentada por el mensaje de interés. Cada encaminador lo incrementa con la latencia medida/estimada desde el salto anterior.

35 Para soportar la realización de una subasta, los mensajes de datos se pueden extender con el campo siguiente:

- 40 • **ignorar (contador de ignorar):** este campo representa el número de nodos de almacenamiento caché próximos que no ganaron la subasta y por lo tanto no almacenarán en caché el objeto de datos. Esencialmente, este campo mantiene un seguimiento de cuántos nodos de almacenamiento caché tiene que atravesar el mensaje de datos antes de que sea almacenado en caché, y por lo tanto su valor disminuye en cada salto. Cuando alcanza el valor 0, los datos se pueden almacenar en caché (dependiendo de si el campo "caché" de la PIT -ver más abajo- está ajustado a verdadero o a falso) y el campo ignorar se ajustaría en consecuencia al valor de bid_ig (ver PIT a continuación) de dicho nodo de almacenamiento caché particular: es decir, se restablece el cómputo de los subsiguientes nodos de almacenamiento caché que tienen que ser ignorados.

45 Para soportar la realización de una subasta, la PIT almacenada en la por lo menos una memoria del nodo se puede extender con los siguientes dos campos:

- 50 • **caché (información de puja):** este campo se ajusta a verdadero si el nodo de almacenamiento caché ha actualizado el campo bid_win del mensaje de interés, de lo contrario se ajusta a falso. Verdadero significa que un nodo de almacenamiento caché ha recibido su capacidad de almacenar el objeto de datos entrante (es decir, ha pujado para el almacenamiento en caché del objeto de datos). Falso significa que el nodo de almacenamiento caché ignorará el objeto de datos entrante con respecto a almacenamiento en caché.

- **bid_ig (cómputo de pujadores perdedores):** si el campo caché es verdadero, este campo se ajusta al valor del campo bid_num del mensaje de interés, de lo contrario permanece vacío.

Además, el nodo de almacenamiento caché puede almacenar parámetros adicionales para soportar la subasta, tal como se describe a continuación:

- 5
- **bid_exp:** este campo representa el número de nodos de almacenamiento caché que pueden participar en un grupo de subasta. Si está ausente, todas las cachés en el trayecto de interés participarán en la misma subasta.
 - **auction_thresh (umbral de subasta):** los nodos de almacenamiento caché con datos más recientes que este umbral no participarán en la subasta. Si está ausente, el nodo de almacenamiento caché tomará partido en la subasta, independientemente de la novedad de los datos que almacene.

10 Se explica un ejemplo de un proceso de transmisión de un mensaje de interés y el proceso de recepción de un mensaje de datos, con la ayuda de las figuras 4 a 10. Para este ejemplo, el mensaje de interés que es enviado por el solicitante de contenido, es decir el nodo en el que se define el mensaje de interés, se parecerá a un mensaje de interés estándar sin ningún campo adicional. Esto permite que el solicitante de contenido sea neutral en cuanto a subastas.

15 En una realización alternativa, el solicitante de contenido puede estar al corriente del mecanismo de subasta y puede incluir campos adicionales en el mensaje de interés para facilitar la subasta, por ejemplo, bid_win y bid_num. Los valores de estos campos de dejarán vacíos o indefinidos, o bien contendrán valores que no están en el intervalo de los valores potenciales reales, para señalar al primer nodo del trayecto que la puja debe comenzar allí.

20 La figura 4 es un diagrama de flujo que muestra un proceso realizado a continuación de la recepción de un mensaje de interés en una realización del procedimiento de la invención. Después de recibir el mensaje de interés entrante en la etapa 41, el nodo de almacenamiento caché determinará si el objeto de datos solicitado está almacenado en su memoria caché, en la etapa 42. Si el objeto solicitado ha sido almacenado en caché por el nodo de almacenamiento caché, este determina en la etapa 43 que, por lo tanto, la subasta ha finalizado, prepara un mensaje de datos que comprende el objeto de datos en la etapa 44 y envía de vuelta el mensaje de datos sobre el trayecto de interés en la etapa 45.

25 Si el objeto solicitado no ha sido almacenado en caché por el nodo de almacenamiento caché, éste comprueba en la etapa 46 si el mensaje de interés comprende campos específicos de subasta, tales como bid_win o bid_num. Si no, el nodo de almacenamiento caché determina entonces que es el primer nodo en el trayecto de interés que participa en la subasta, y lleva a cabo la etapa 50. Los detalles de la etapa 50 se proporcionan en la figura 5.

30 Si el mensaje de interés comprende campos específicos de subasta, el nodo de almacenamiento caché determina en la etapa 47 que no es el primer nodo en el trayecto de interés que participa en la subasta. A continuación comprueba si bid_num es igual a bid_exp - 1 en la etapa 48 y realiza la etapa 49 si bid_num es igual a bid_exp - 1, es decir, si el nodo de almacenamiento caché es el primer nodo de un nuevo grupo de subasta (y por lo tanto, debería ignorar el campo bid_win transportado en el mensaje de interés). El nodo de almacenamiento caché realiza la etapa 35 51 si bid_num no es igual a bid_exp - 1, es decir, el nodo de almacenamiento caché forma parte del mismo grupo de subasta que su predecesor. Los detalles de las etapas 49 y 51 se proporcionan en las figuras 6 y 7, respectivamente.

40 Puede ser posible que un nodo de almacenamiento caché anule el valor bid_exp para finalizar la subasta actual antes de su tiempo de vencimiento real. Esto puede ser beneficioso cuando un nodo de almacenamiento caché tiene un enlace ascendente de poco ancho de banda o cuando un nodo de almacenamiento caché está situado en la unión de diferentes tipos de redes, por ejemplo, en el límite de redes fijas y redes ad hoc vehiculares (VANET, vehicular ad hoc networks) o en el límite entre enlaces continentales.

45 El primer nodo de almacenamiento caché que participa en la subasta comprobará primero su umbral de subasta (si está definido) en la etapa 81, ver la figura 5, y si los datos almacenados en caché en el nodo son más antiguos que el umbral de subasta, o si siguen existiendo posiciones disponibles de almacenamiento en caché, decidirá pujar. Para ello, calculará un valor de puja en la etapa 82, por ejemplo del modo descrito anteriormente. A continuación ajustará la entrada de caché a verdadero en la PIT en la etapa 83 y la entrada bid_ig a 0 en la PIT en la etapa 84. Después de la etapa 84, el nodo de almacenamiento caché añadirá el contexto de la solicitud al mensaje de interés en la etapa 53, añadirá un campo bid_num al mensaje de interés con un valor 0 en la etapa 54 y añadirá un campo 50 bid_win al mensaje de interés, con el valor de puja calculado como su valor, en la etapa 55 antes de reenviar el mensaje de interés al siguiente salto definido según la estrategia regular de reenvío de ICN, en la etapa 88.

55 Si los datos almacenados en caché en el nodo son más jóvenes que el umbral de subasta y no existen posiciones disponibles de almacenamiento en caché, el nodo de almacenamiento caché decidirá no pujar en la etapa 85. A continuación ajustará la entrada de caché a falso en la PIT en la etapa 86 y dejará la entrada bid_ig indefinida en la PIT en la etapa 87. Después de la etapa 87, el nodo de almacenamiento caché añadirá el contexto de la solicitud al mensaje de interés en la etapa 56, añadirá un campo bid_num al mensaje de interés con un valor 0 en la etapa 57 y

añadirá un campo `bid_win` al mensaje de interés sin ningún valor en la etapa 58 antes de reenviar el mensaje de interés al siguiente salto en la etapa 88.

5 El primer nodo de almacenamiento caché de un nuevo grupo de subasta comprobará en primer lugar su umbral de subasta (si está definido) en la etapa 81, ver la figura 6) y si los datos almacenados en caché en el nodo son más antiguos que el umbral de subasta, o si existen posiciones disponibles de almacenamiento en caché, decidirá pujar. Para ello, calculará un valor de puja en la etapa 82, por ejemplo del modo descrito anteriormente. A continuación ajustará la entrada de caché a verdadero en la PIT en la etapa 83 y la entrada `bid_ig` a 0 en la PIT en la etapa 84. Después de la etapa 84, el nodo de almacenamiento caché actualizará el contexto de la solicitud en el mensaje de interés en la etapa 63, ajustará el campo `bid_num` del mensaje de interés a 0 en la etapa 64 y ajustará el campo `bid_win` al valor de puja calculado, en la etapa 65, antes de reenviar el mensaje de interés al siguiente salto en la etapa 88.

15 Si los datos almacenados en caché en el nodo son más jóvenes que el umbral de subasta y no existen posiciones disponibles de almacenamiento en caché, el nodo de almacenamiento caché decidirá no pujar en la etapa 85. A continuación ajustará la entrada de caché a falso en la PIT en la etapa 86 y dejará la entrada `bid_ig` indefinida en la PIT en la etapa 87. Después de la etapa 87, el nodo de almacenamiento caché actualizará el contexto de la solicitud en el mensaje de interés en la etapa 66, ajustará el campo `bid_num` del mensaje de interés a 0 en la etapa 67 y ajustará el campo `bid_win` del mensaje de interés a ningún valor en la etapa 68, antes de reenviar el mensaje de interés al siguiente salto en la etapa 88.

20 Si el nodo de almacenamiento caché está en el mismo grupo de subasta que su predecesor, comprobará en primer lugar su umbral de subasta (si está definido) en la etapa 81, ver la figura 7, y si los datos almacenados en caché en el nodo son más antiguos que el umbral de subasta, o si siguen existiendo posiciones disponibles de almacenamiento en caché, decidirá pujar. Para ello, calculará un valor de puja en la etapa 82, por ejemplo del modo descrito anteriormente. A continuación comprobará en la etapa 71 si la puja calculada es superior que el valor `bid_win` del mensaje de interés recibido. En caso afirmativo, ajustará la entrada de caché a verdadero en la PIT en la etapa 83 y la entrada `bid_ig` a `bid_num + 1` en la PIT en la etapa 72. Después de la etapa 72, el nodo de almacenamiento caché actualizará el contexto de la solicitud en el mensaje de interés en la etapa 73, aumentará el valor del campo `bid_num` del mensaje de interés en 1 en la etapa 74 y ajustará el campo `bid_win` al valor de puja calculado, en la etapa 75, antes de reenviar el mensaje de interés al siguiente salto en la etapa 88.

30 Si la puja calculada no es mayor que el valor `bid_win` del mensaje de interés recibido, en la etapa 71, o si los datos almacenados en caché en el nodo son más jóvenes que el umbral de subasta y no existen posiciones disponibles de almacenamiento en caché, el nodo de almacenamiento caché decidirá no pujar en la etapa 85. A continuación ajustará la entrada de caché a falso en la PIT en la etapa 86 y dejará la entrada `bid_ig` indefinida en la PIT en la etapa 87. Después de la etapa 87, el nodo de almacenamiento caché actualizará el contexto de la solicitud en el mensaje de interés en la etapa 76, aumentará el valor del campo `bid_num` del mensaje de interés en 1 en la etapa 77 y dejará sin cambios el valor del campo `bid_win` del mensaje de interés en la etapa 78, antes de reenviar el mensaje de interés al siguiente salto en la etapa 88.

40 Una vez que el mensaje de interés llega al nodo que contiene una copia del objeto de datos (ya sea un nodo intermediario que tiene almacenada en caché una copia del objeto de datos o el propio nodo generador, por ejemplo el servidor 35 de la figura 3), dicho nodo empaquetará el objeto de datos en un mensaje de datos y lo enviara de vuelta sobre el trayecto de interés. El mensaje de datos enviado por el nodo que contiene una copia del objeto de datos puede tener el formato de un mensaje de datos estándar, que permite que el proveedor de contenidos sea neutral en cuanto a subastas. En una realización alternativa, el proveedor de contenidos puede estar al corriente del mecanismo de subastas y puede incluir campos adicionales en el mensaje de datos para facilitar la subasta, por ejemplo, un campo ignorar. Los diagramas de flujo de las figuras 4 a 7 describen un nodo de almacenamiento caché que recibe un mensaje de interés para un objeto de datos solicitado para el que no está registrado actualmente un interés en la PIT. Cuando un nodo de almacenamiento caché recibe un mensaje de interés para el que ya se ha registrado un interés en la PIT, el nodo de más abajo se asociará con el ya registrado interés/nombre de objeto de datos solicitado, pero no se reenviará ningún mensaje de interés. En lo que se refiere a este nuevo interés, la subasta finaliza por lo tanto en este nodo.

50 La figura 8 es un diagrama de flujo que muestra un proceso realizado a continuación de la recepción de un mensaje de datos en una realización del procedimiento de la invención. Una vez que un nodo recibe el mensaje de datos en la etapa 120, comprueba en primer lugar si existe alguna entrada para el objeto de datos en su PIT, en la etapa 121. Si no existe ninguna entrada para el objeto de datos en la PIT, el nodo determina entonces en la etapa 127 que el mensaje de datos ha llegado a su destino final, es decir, el originador del mensaje de interés, y no se adopta ninguna acción. Si existe una entrada para el objeto de datos en la PIT, el nodo determina en la etapa 122 que el nodo es un nodo intermedio con respecto al mensaje de datos recibido. Si el nodo determina en la etapa 123 que soporta almacenamiento en caché, gestiona el primer interés registrado en la PIT, en la etapa 124. Se proporcionan detalles de la etapa 124 en la figura 9. Si el nodo determina en la etapa 123 que no soporta almacenamiento en caché, reenvía el mensaje de datos (sin cambios) en la etapa 128 sobre la interfaz o interfaces de comunicación asociadas con dichos uno o varios intereses registrados en la PIT para este objeto de datos.

Después de que el primer interés en la PIT ha sido gestionado en la etapa 124, el nodo determina en la etapa 125 si existen algunos otros intereses para este objeto de datos en la PIT. Si no existen otros intereses registrados en la PIT, el nodo detendrá el proceso en la etapa 129. Si existen otros intereses registrados en la PIT, la etapa 126 se realizará para cada uno de estos otros intereses. Se proporcionan detalles de la etapa 126 en la figura 10.

5 Para el primer interés en la PIT, el nodo de almacenamiento caché comprobará si el valor del campo caché en su PIT está ajustado a "verdadero" en la etapa 141, de la figura 9. En caso afirmativo, comprobará en la etapa 91 si el mensaje de datos contiene un campo ignorar y, en caso afirmativo, si el campo ignorar está ajustado a 0. Si el campo ignorar está ausente, significa que no hay ningún otro nodo de más arriba (que participe en la subasta) almacenado en caché en el objeto de datos. Si el campo ignorar está ausente o ajustado a 0, el nodo procederá
10 entonces con el almacenamiento en caché del elemento en la etapa 142, y ajustará a continuación el campo ignorar del mensaje de datos al valor del campo bid_ig de su PIT, en la etapa 143. Después de la etapa 143, se prepara en la etapa 144 un mensaje de datos con el nuevo valor ignorar, y se reenvía hacia abajo sobre el trayecto de interés en la etapa 145.

15 Sin embargo, si se determina que el valor del campo ignorar es mayor que 0 en la etapa 91, el nodo decidirá en la etapa 93 no almacenar en caché el elemento de datos. El nodo disminuirá a continuación el valor del campo ignorar en 1 en la etapa 95 antes de preparar el mensaje de datos con un nuevo valor ignorar en la etapa 144, y reenvía el mensaje de datos más abajo sobre el trayecto de interés en la etapa 145.

20 Por el contrario, si se determina en la etapa 141 que el valor del campo caché en su PIT está ajustado a "falso", el nodo decidirá no almacenar en caché el objeto de datos en la etapa 92. A continuación, el nodo reenviará intacto el mensaje de datos más abajo en el trayecto de interés en la etapa 145, es decir, al nodo de más abajo desde el que recibió el mensaje de interés.

25 Para un interés diferente al primer interés registrado en la PIT, el nodo de almacenamiento caché comprobará en la etapa 141, ver la figura 10, si el valor del campo caché en su PIT está ajustado a "verdadero". En caso afirmativo, el nodo comprobará en la etapa 106 si el objeto de datos ha sido ya almacenado en caché, por ejemplo, cuando se procesan los intereses anteriores en la PIT para este objeto. En caso negativo, el nodo procederá con el almacenamiento en caché del objeto de datos en la etapa 142, y a continuación avanza a la etapa 143. Si el objeto de datos ha sido ya almacenado en caché, se omite la etapa 142 y se realiza directamente la etapa 143 después de la etapa 106. En la etapa 143, el nodo ajustará el campo ignorar del mensaje de datos al valor del campo bid_ig de su PIT. Después de la etapa 143, se prepara en la etapa 144 un mensaje de datos con el nuevo valor ignorar, y se
30 reenvía hacia abajo sobre el trayecto de interés en la etapa 145.

Por el contrario, si se determina en la etapa 141 que el valor del campo caché en su PIT está ajustado a "falso", el nodo decidirá no almacenar en caché el objeto de datos en la etapa 102. A continuación, el nodo ajustará el campo ignorar del mensaje de datos a 0 en la etapa 105, preparará un mensaje de datos con el nuevo valor de ignorar en la etapa 144 y reenviará el mensaje de datos más abajo sobre el trayecto de interés en la etapa 145.

35 En una realización alternativa, puede ser deseable que los datos se almacenen en caché solamente una vez entre el solicitante y el origen de los datos. En este caso, todos los nodos en el trayecto pueden tomar parte en la misma subasta, es decir, solamente existirá un grupo de subasta, y el nodo de almacenamiento caché puede no almacenar (o puede ignorar el valor de) un campo bid_exp. Por ejemplo, cuando el bid_exp tiene un valor especial, por ejemplo 0, el nodo de almacenamiento caché puede saber que existe solamente un grupo de subasta.

40 El campo bid_exp puede asimismo estar ausente si el número de grupos de subasta es decidido por el origen del objeto de datos. Se podrían acoplar más campos al mensaje de interés, que ayudarían a comunicar información extra a los nodos de almacenamiento caché de enlace ascendente. Esta información sería entonces utilizada por el nodo de almacenamiento caché con el contenido solicitado, o por el servidor, para elegir a continuación el valor apropiado de bid_exp. Por lo tanto, en lugar de ajustar bid_exp en el camino desde el usuario hasta el origen, en este caso el valor de bid_exp se ajustará en el trayecto de retorno desde el servidor hacia el usuario.
45

La figura 11 muestra un ejemplo de campos y valores introducidos en el mensaje de interés, el mensaje de datos y la PIT de la invención. La figura 11 representa el mismo trayecto de red que la figura 3. En este ejemplo, una puja de un nodo está basada solamente en su valor LRU y se utiliza un bid_exp de 3 nodos y un auction_thresh de 3 segundos. En este ejemplo, un nodo de red puja solamente cuando su valor LRU es mayor o igual que el umbral de subasta auction_thresh. El valor LRU representa cuánto hace que se accedió al objeto de datos almacenado en caché utilizado menos recientemente.
50

El dispositivo de usuario 31 en el que se origina el mensaje de interés es neutral en cuanto a subastas, es decir, transmite un mensaje de interés convencional sin campos adicionales al nodo de almacenamiento caché 1₁. El nodo de almacenamiento caché 1₁ recibe el mensaje de interés del dispositivo de usuario 31, pero decide no pujar, debido a que su valor LRU (1 segundo) es menor que el umbral de subasta. Ajusta la información de puja (por ejemplo, el valor del campo caché) de su PIT a falso y deja sin definir el contador de pujadores perdedores (por ejemplo, el valor del campo bid_ig) de su PIT. Dado que es el primer nodo el que toma parte en la subasta, añade los campos de puja (denominado asimismo bid_win) y bid_num al mensaje de interés antes de reenviar el mensaje de interés al nodo de
55

almacenamiento caché 1₂. El valor de puja se deja sin definir y el valor bid_num (es decir, el contador de nodos) se ajusta a 0.

5 El nodo de almacenamiento caché 1₂ recibe el mensaje de interés desde el nodo de almacenamiento caché (de más abajo) 1₁, pero decide no pujar, debido a que su valor LRU (1 segundo) es menor que el umbral de subasta. Ajusta la información de puja (por ejemplo, el valor del campo caché) de su PIT a falso y deja sin definir el cómputo de pujadores perdedores (por ejemplo, el valor del campo bid_ig) de su PIT. En relación con el mensaje de interés, el nodo de almacenamiento caché 1₂ deja sin definir el valor de puja, aumenta el valor de bid_num (es decir, el contador de nodos) a 2, y reenvía a continuación el mensaje de interés al nodo de almacenamiento caché (de más arriba) 1₃.

10 El nodo de almacenamiento caché 1₃ recibe el mensaje de interés desde el nodo de almacenamiento caché 1₂, pero decide no pujar, debido a que su valor de LRU (2 segundos) es menor que el umbral de subasta. Ajusta la información de puja (por ejemplo, el valor del campo caché) de su PIT a falso y deja sin definir el cómputo de pujadores perdedores (por ejemplo, el valor del campo bid_ig) de su PIT. En relación con el mensaje de interés, el nodo de almacenamiento caché 1₃ deja sin definir el valor de puja, aumenta a 2 el valor de bid_num y reenvía a continuación el mensaje de interés al nodo de almacenamiento caché 1₄.

15 El nodo de almacenamiento caché 1₄ recibe el mensaje de interés del nodo de almacenamiento caché 1₃ y determina primero que bid_num es igual a bid_exp - 1, es decir, que el nodo de almacenamiento caché 1₄ es el primer nodo de un nuevo grupo de subasta. El nodo de almacenamiento caché 1₄ decide a continuación calcular un valor de puja, dado que su valor de LRU (5 segundos) es mayor que el umbral de subasta. Calcula un valor de puja de 5 (igual al valor de LRU). En una realización alternativa, los valores de puja se normalizan entre los valores de 0 y 1 (un valor de LRU de 5 se puede normalizar de acuerdo con la ecuación 1, y por ejemplo adoptar el valor de 0,5). El nodo de almacenamiento caché 1₄ ajusta a continuación la información de puja (por ejemplo, el valor del campo caché) de su PIT a verdadero y ajusta el cómputo de pujadores perdedores (por ejemplo, el valor del campo bid_ig) de su PIT a 0. A continuación ajusta el valor de puja del mensaje de interés a su valor de puja calculado y restablece bid_num (es decir, el contador de nodos) a cero, antes de reenviar el mensaje de interés al nodo de almacenamiento caché 1₅.

20 El nodo de almacenamiento caché 1₅ recibe el mensaje de interés del nodo de almacenamiento caché 1₄ y decide calcular un valor de puja, dado que su valor de LRU (6 segundos) es mayor que el umbral de subasta. Calcula un valor de puja de 6 (igual al valor de LRU). Dado que el valor de puja calculado (6) es mayor que el valor de puja recibido (5), el nodo de almacenamiento caché 1₅ ajusta el valor de puja del mensaje de interés al valor de puja calculado, ajusta la información de puja (por ejemplo, el valor del campo caché) de su PIT a verdadero y ajusta el cómputo de pujadores perdedores (por ejemplo, el valor del campo bid_ig) de su PIT a bid_num + 1, es decir a 1. Aumenta además el valor de bid_num (es decir, el contador de nodos) a 1 y reenvía a continuación el mensaje de interés al nodo de almacenamiento caché 1₆.

30 El nodo de almacenamiento caché 1₆ recibe el mensaje de interés del nodo de almacenamiento caché 1₅ y decide calcular un valor de puja, dado que su valor de LRU (4 segundos) es mayor que el umbral de subasta. Calcula un valor de puja de 4 (igual al valor de LRU). Dado que el valor de puja calculado (4) es menor que el valor de puja recibido (6), el nodo de almacenamiento caché 1₆ deja sin cambios el campo de puja en el mensaje de interés, ajusta la información de puja (por ejemplo, el valor del campo caché) de su PIT a falso y ajusta a indefinido el cómputo de pujadores perdedores de su PIT. Aumenta además el valor de bid_num (es decir, el contador de nodos) a 2 y reenvía a continuación el mensaje de interés al nodo de almacenamiento caché 1₇.

35 El nodo de almacenamiento caché 1₇ recibe el mensaje de interés del nodo de almacenamiento caché 1₆ y determina primero que bid_num es igual a bid_exp - 1, es decir, que el nodo de almacenamiento caché 1₇ es el primer nodo de un nuevo grupo de subasta. El nodo de almacenamiento caché 1₇ decide a continuación calcular un valor de puja, dado que su valor de LRU (4 segundos) es mayor que el umbral de subasta. Calcula un valor de puja de 4 (igual al valor de LRU). El nodo de almacenamiento caché 1₇ ajusta a continuación la información de puja (por ejemplo, el valor del campo caché) de su PIT a verdadero y ajusta el cómputo de pujadores perdedores (por ejemplo, el valor del campo bid_ig) de su PIT a 0. A continuación ajusta el valor de puja del mensaje de interés a su valor de puja calculado y restablece bid_num (es decir, el contador de nodos) a cero, antes de reenviar el mensaje de interés al nodo de almacenamiento caché 1₈.

40 El nodo de almacenamiento caché 1₈ recibe el mensaje de interés del nodo de almacenamiento caché 1₇ y decide calcular un valor de puja, dado que su valor de LRU (5 segundos) es mayor que el umbral de subasta. Calcula un valor de puja de 5 (igual al valor de LRU). Dado que el valor de puja calculado (5) es mayor que el valor de puja recibido (4), el nodo de almacenamiento caché 1₈ ajusta el valor de puja del mensaje de interés al valor de puja calculado, ajusta la información de puja (por ejemplo, el valor del campo caché) de su PIT a verdadero y ajusta el cómputo de pujadores perdedores (por ejemplo, el valor del campo bid_ig) de su PIT a bid_num + 1, es decir a 1. Aumenta además el valor de bid_num (es decir, el contador de nodos) a 1 y reenvía a continuación el mensaje de interés al nodo de almacenamiento caché 1₉.

45 El nodo de almacenamiento caché 1₉ recibe el mensaje de interés del nodo de almacenamiento caché 1₈ y decide calcular un valor de puja, dado que su valor de LRU (6 segundos) es mayor que el umbral de subasta. Calcula un

valor de puja de 6 (igual al valor de LRU). Dado que el valor de puja calculado (6) es mayor que el valor de puja recibido (5), el nodo de almacenamiento caché 1₉ ajusta el valor de puja del mensaje de interés al valor de puja calculado, ajusta la información de puja (por ejemplo, el valor del campo caché) de su PIT a verdadero y ajusta el cómputo de pujadores perdedores (por ejemplo, el valor del campo bid_ig) de su PIT a bid_num +1, es decir a 2. Aumenta además el valor de bid_num (es decir, el contador de nodos) a 2, y reenvía a continuación el mensaje de interés al servidor 35.

Cuando el servidor 35 recibe el mensaje de interés del nodo de almacenamiento caché 1₉, no reconoce los campos de puja y bid_num, pero determina que ha almacenado el objeto de datos solicitado y transmite el objeto de datos solicitado al nodo de almacenamiento caché 1₉ en un mensaje de datos. El servidor 35 es neutral en cuanto a subastas, es decir, transmite un mensaje de datos convencional sin campos adicionales al nodo de almacenamiento caché 1₉.

El nodo de almacenamiento caché 1₉ recibe el mensaje de datos del servidor 35 y determina a partir de su PIT que la información de puja (por ejemplo, el valor del campo caché) está ajustado a verdadero y que el contador de ignorar está ausente en el mensaje de datos recibido. Concluye que ha ganado la puja en su grupo de subasta y almacena en caché el objeto de datos. A continuación, añade un contador de ignorar al mensaje de datos y lo ajusta al valor del cómputo de pujadores perdedores (por ejemplo, el valor del campo bid_ig), es decir a 2, antes de reenviar el mensaje de datos al nodo de almacenamiento caché 1₈.

El nodo de almacenamiento caché 1₈ recibe el mensaje de datos del nodo de almacenamiento caché (de más arriba) 1₉ y determina a partir de su PIT que la información de puja (por ejemplo, el valor del campo caché) está ajustado a verdadero y, a partir del mensaje de datos recibido, que el contador de ignorar no está ajustado a 0. Concluye que ha perdido la puja en su grupo de subasta y no almacena en caché el objeto de datos. A continuación, disminuye el valor del contador de ignorar en 1, es decir a 1, antes de reenviar el mensaje de datos al nodo de almacenamiento caché (de más abajo) 1₇.

El nodo de almacenamiento caché 1₇ recibe el mensaje de datos del nodo de almacenamiento caché 1₈ y determina a partir de su PIT que la información de puja (por ejemplo, el valor del campo caché) está ajustado a verdadero y, a partir del mensaje de datos recibido, que el contador de ignorar no está ajustado a 0. Concluye que ha perdido la puja en su grupo de subasta y no almacena en caché el objeto de datos. A continuación, disminuye el valor del contador de ignorar en 1, es decir a 0, antes de reenviar el mensaje de datos al nodo de almacenamiento caché 1₆.

El nodo de almacenamiento caché 1₆ recibe el mensaje de datos del nodo de almacenamiento caché 1₇ y determina a partir de su PIT que la información de puja (por ejemplo, el valor del campo caché) está ajustado a falso. Concluye que no pujó para almacenar el objeto de datos y no almacena en caché el objeto de datos. Deja sin cambios a 0 el valor del contador de ignorar y reenvía el mensaje de datos al nodo de almacenamiento caché 1₅.

El nodo de almacenamiento caché 1₅ recibe el mensaje de datos del nodo de almacenamiento caché 1₆ y determina a partir de su PIT que la información de puja (por ejemplo, el valor del campo caché) está ajustado a verdadero y, a partir del mensaje de datos recibido, que el contador de ignorar está ajustado a 0. Concluye que ha ganado la puja en su grupo de subasta y almacena en caché el objeto de datos. A continuación ajusta el contador de ignorar del mensaje de datos al valor del cómputo de pujadores perdedores (por ejemplo, el valor del campo bid_ig), es decir a 1, antes de reenviar el mensaje de datos al nodo de almacenamiento caché 1₄.

El nodo de almacenamiento caché 1₄ recibe el mensaje de datos del nodo de almacenamiento caché 1₅ y determina a partir de su PIT que la información de puja (por ejemplo, el valor del campo caché) está ajustado a verdadero y, a partir del mensaje de datos recibido, que el contador de ignorar no está ajustado a 0. Concluye que ha perdido la puja en su grupo de subasta y no almacena en caché el objeto de datos. A continuación, disminuye el valor del contador de ignorar en 1, es decir a 0, antes de reenviar el mensaje de datos al nodo de almacenamiento caché 1₃.

El nodo de almacenamiento caché 1₃ recibe el mensaje de datos del nodo de almacenamiento caché 1₄ y determina a partir de su PIT que la información de puja (por ejemplo, el valor del campo caché) está ajustado a falso. Concluye que no pujó para almacenar el objeto de datos y no almacena en caché el objeto de datos. Deja sin cambios a 0 el valor del contador de ignorar y reenvía el mensaje de datos al nodo de almacenamiento caché 1₂.

El nodo de almacenamiento caché 1₂ recibe el mensaje de datos del nodo de almacenamiento caché 1₃ y determina a partir de su PIT que la información de puja (por ejemplo, el valor del campo caché) está ajustado a falso. Concluye que no pujó para almacenar el objeto de datos y no almacena en caché el objeto de datos. Deja sin cambios a 0 el valor del contador de ignorar y reenvía el mensaje de datos al nodo de almacenamiento caché 1₁.

El nodo de almacenamiento caché 1₁ recibe el mensaje de datos del nodo de almacenamiento caché 1₂ y determina a partir de su PIT que la información de puja (por ejemplo, el valor del campo caché) está ajustado a falso. Concluye que no pujó para almacenar el objeto de datos y no almacena en caché el objeto de datos. Deja sin cambios a 0 el valor del contador de ignorar y reenvía el mensaje de datos al dispositivo de usuario 31. El dispositivo de usuario 31 reconoce el contador de ignorar, pero determina que no existen entradas para este objeto de datos en la PIT y que el mensaje de datos ha llegado por lo tanto al solicitante de contenido.

La figura 12 representa un diagrama de bloques que muestra un sistema de procesamiento de datos a modo de ejemplo que puede realizar los procedimientos que se han descrito haciendo referencia a la figura 2 y a las figuras 4 a 10.

Tal como se muestra en la figura 12, el sistema de procesamiento de datos 300 puede incluir por lo menos un procesador 302 acoplado a elementos de memoria 304 a través de un bus del sistema 306. De este modo, el sistema de procesamiento de datos puede almacenar código de programa dentro de los elementos de memoria 304. Además, el procesador 302 puede ejecutar código de programa al que se accede en los elementos de memoria 304 por medio de un bus del sistema 306. En un aspecto, el sistema de procesamiento de datos se puede implementar en un ordenador que es adecuado para almacenar y/o ejecutar código de programa. Sin embargo, se debe apreciar que el sistema de procesamiento de datos 300 se puede implementar en forma de cualquier sistema que incluya un procesador y una memoria que puedan llevar a cabo las funciones descritas en esta memoria descriptiva.

Los elementos de memoria 304 pueden incluir uno o varios dispositivos físicos de memoria tales como, por ejemplo, memoria local 308 y uno o varios dispositivos de almacenamiento de gran capacidad 310. La memoria local se puede referir a memoria de acceso aleatorio o a otro dispositivo o dispositivos de memoria no persistente, utilizados generalmente durante la ejecución real del código de programa. Un dispositivo de almacenamiento de gran capacidad se puede implementar como un disco duro u otro dispositivo de almacenamiento de datos persistente. El sistema de procesamiento 300 puede incluir asimismo una o varias memorias caché (no mostradas) que proporcionan almacenamiento temporal de por lo menos algún código de programa con el fin de reducir el número de veces que es necesario recuperar código de programa desde el dispositivo de almacenamiento de gran capacidad 310 durante la ejecución.

Los dispositivos de entrada/salida (I/O) representados como un dispositivo de entrada 312 y un dispositivo de salida 314 pueden opcionalmente estar acoplados al sistema de procesamiento de datos. Ejemplos de dispositivos de entrada pueden incluir, de forma no limitativa, un teclado, un dispositivo de puntero tal como un ratón, o similar. Ejemplos de dispositivos de salida pueden incluir, de forma no limitativa, un monitor o una pantalla, altavoces o similar. Los dispositivos de entrada y/o de salida pueden estar acoplados al sistema de procesamiento de datos, ya sea directamente o mediante controladores I/O intermedios.

En una realización, los dispositivos de entrada y de salida se pueden implementar como un dispositivo de entrada/salida combinado (mostrado en la figura 12 con una línea de trazos que rodea el dispositivo de entrada 312 y el dispositivo de salida 314). Un ejemplo de dicho dispositivo combinado es un visualizador sensible al tacto, denominado en ocasiones "visualizador de pantalla táctil" o simplemente "pantalla táctil". En una realización de este tipo, la entrada al dispositivo se puede proporcionar mediante un movimiento de un objeto físico, tal como por ejemplo un lápiz o un dedo de un usuario, sobre o cerca del visualizador de pantalla táctil.

Un adaptador de red 316 puede estar acoplado asimismo al sistema de procesamiento de datos para permitir que este se acople a otros sistemas, sistemas informáticos, dispositivos de red remotos y/o dispositivos de almacenamiento remotos por medio de redes privadas o públicas intermedias. El adaptador de red puede comprender un receptor de datos para recibir datos que son transmitidos por dichos sistemas, dispositivos y/o redes al sistema de procesamiento de datos 300, y un transmisor de datos para transmitir datos desde el sistema de procesamiento de datos 300 a dichos sistemas, dispositivos y/o redes. Los módems, módems de cable y tarjetas Ethernet son ejemplos de diferentes tipos de adaptador de red que pueden ser utilizados con el sistema de procesamiento de datos 300.

Tal como se representa en la figura 12, los elementos de memoria 304 pueden almacenar una aplicación 318. En varias realizaciones, la aplicación 318 se puede almacenar en la memoria local 308, en dichos uno o varios dispositivos de almacenamiento de gran capacidad 310, o separados de la memoria local y de los dispositivos de almacenamiento de gran capacidad. Se deberá apreciar que el sistema de procesamiento de datos 300 puede ejecutar además un sistema operativo (no mostrado en la figura 17) que puede facilitar la ejecución de la aplicación 318. La aplicación 318, implementándose en forma de código de programa ejecutable, puede ser ejecutada por el sistema de procesamiento de datos 300, por ejemplo, por el procesador 302. En respuesta a la ejecución de la aplicación, el sistema de procesamiento de datos 300 puede estar configurado para realizar una o varias operaciones o etapas de procedimiento descritas en la presente memoria.

La invención se refiere asimismo a un nodo terminal 330 que comprende por lo menos una interfaz de comunicación 331 y por lo menos un procesador 335, ver la figura 13. Dicho por lo menos un procesador 335 está configurado para determinar un nombre de un objeto de datos solicitado (por ejemplo, permitiendo a un usuario especificar este nombre), crear un mensaje de interés que comprende el nombre del objeto de datos solicitado y un campo de puja para introducir un valor de puja, dejándose sin definir el valor de puja (por ejemplo, ningún valor o un valor '0') para indicar que debería comenzar una subasta del derecho para almacenar en caché el objeto de datos solicitado, y transmitir el mensaje de interés a un nodo de más arriba 1_1 sobre la por lo menos una interfaz de comunicación 331. El nodo terminal 330 puede ser, por ejemplo, un dispositivo de usuario 31 y puede comprender, por ejemplo, una interfaz de usuario, por ejemplo una pantalla táctil. El nodo terminal 330 puede ser un ordenador portátil, una tableta, un teléfono móvil o un dispositivo incorporado en un vehículo, por ejemplo. Dicho por lo menos un procesador 335 puede ser un procesador ARM o Qualcomm, por ejemplo. El nodo terminal 330 puede comprender además otros

componentes típicos para un nodo terminal, por ejemplo una memoria y una fuente de alimentación. Dicha por lo menos una interfaz de comunicación 331 puede comprender un transceptor (por ejemplo, celular y/o de Wi-Fi) de datos inalámbricos, por ejemplo.

5 Varias realizaciones de la invención se pueden implementar como un producto de programa para utilizar con un sistema informático, donde el programa o programas del producto de programa definen funciones de las realizaciones (que incluyen los procedimientos descritos en la presente memoria). En una realización, el programa o programas pueden estar contenidos en una serie de medios de almacenamiento no transitorio legibles por ordenador donde, tal como se utiliza en la presente memoria, la expresión "medios de almacenamiento no transitorio legibles por ordenador" comprende todos los medios legibles por ordenador, con la única excepción de una señal de propagación, transitoria. En otra realización, el programa o programas pueden estar contenidos en varios medios de almacenamiento transitorio legibles por ordenador. Los medios de almacenamiento legibles por ordenador a modo ilustrativo incluyen, de forma no limitativa: (i) medios de almacenamiento no grabables (por ejemplo, dispositivos de memoria de sólo lectura dentro de un ordenador, tal como discos CD-ROM legibles mediante una unidad de CD-COM, chips ROM o cualquier tipo de memoria semiconductora de estado sólido no volátil) en los que la información se almacena permanentemente; y (ii) medios de almacenamiento grabables (por ejemplo, memoria flash, discos flexibles dentro de una unidad de disquete o una unidad de disco duro, o cualquier tipo de memoria de semiconductor de acceso aleatorio de estado sólido) en los que se almacena información modificable. El programa informático se puede ejecutar en un procesador 302 descrito en la presente memoria.

10

15

REIVINDICACIONES

1. Un nodo de red (1₅), que comprende:
 por lo menos una interfaz de comunicación (3₁₋₂);
 por lo menos una memoria (5); y
- 5 por lo menos un procesador (7) configurado:
- para recibir un mensaje de interés sobre dicha por lo menos una interfaz de comunicación (3₁) desde un nodo de más abajo (1₄), comprendiendo dicho mensaje de interés un nombre de un objeto de datos solicitado,
 - para determinar si pujar para almacenar dicho objeto de datos solicitado en dicha por lo menos una memoria (5),
 - para calcular un valor de puja,
- 10 - para reenviar dicho mensaje de interés a un nodo de más arriba (1₆) sobre dicha por lo menos una interfaz de comunicación (3₂), incluyendo el mensaje de interés reenviado dicho valor de puja calculado en función de por lo menos dicha determinación de si pujar,
- para recibir un mensaje de datos que comprende dicho objeto de datos solicitado desde dicho nodo de más arriba (1₆) sobre dicha por lo menos una interfaz de comunicación (3₂),
- 15 - para determinar a partir de por lo menos dicho mensaje de datos si dicho nodo de red (1₅) ganó dicha puja,
- para almacenar dicho objeto de datos en dicha por lo menos una memoria (5) si dicho nodo de red (1₅) ganó dicha puja, y
 - para reenviar dicho mensaje de datos a dicho nodo de más abajo de (1₄) sobre dicha por lo menos una interfaz de comunicación (3₁).
- 20 2. Un nodo de red (1₅) según la reivindicación 1, en el que dicho mensaje de interés recibido comprende además un valor de puja recibido y dicho por lo menos un procesador (7) está configurado para incluir dicho valor de puja calculado en dicho mensaje de interés si dicha determinación de si pujar fue positiva y dicho valor de puja calculado representa una puja superior que dicho valor de puja recibido.
- 25 3. Un nodo de red (1₅) según la reivindicación 1 o 2, en el que dicho por lo menos un procesador (7) está configurado además para registrar información de puja en dicha por lo menos una memoria (5), representando dicha información de puja si dicho nodo de red (1₅) ha incluido dicho valor de puja calculado en dicho mensaje de interés.
4. Un nodo de red (1₅) según la reivindicación 3, en el que dicho por lo menos un procesador (7) está configurado para determinar a partir de dicho mensaje de datos y dicha información de puja si dicho nodo de red (1₅) ganó dicha puja.
- 30 5. Un nodo de red (1₅) según la reivindicación 4, en el que dicho mensaje de datos recibido comprende un contador de ignorar, representando dicho contador de ignorar una cantidad de nodos de más abajo que han perdido dicha puja, determinada por dicho nodo de más arriba (1₆) u otro nodo de más arriba, y dicho por lo menos un procesador (7) está configurado para determinar a partir de dicho contador de ignorar y dicha información de puja si dicho nodo de red (1₅) ganó dicha puja.
- 35 6. Un nodo de red (1₅) según la reivindicación 5, en el que dicho por lo menos un procesador (7) está configurado además para incrementar un contador de nodos en dicho mensaje de interés antes de reenviar dicho mensaje de interés a dicho nodo de más arriba (1₆), representando dicho contador de nodos una cantidad de nodos de más abajo.
- 40 7. Un nodo de red (1₅) según la reivindicación 6, en el que dicho por lo menos un procesador (7) está configurado además para registrar un cómputo de pujadores perdedores en dicha por lo menos una memoria (5) si dicho nodo de red (1₅) ha incluido dicho valor de puja calculado en dicho mensaje de interés, estando basado dicho cómputo de pujadores perdedores en dicho contador de nodos incluido en dicho mensaje de interés recibido y representando la cantidad de nodos de más abajo que han perdido dicha puja, determinada por dicho nodo de red (1₅).
- 45 8. Un nodo de red (1₅) según la reivindicación 7, en el que dicho por lo menos un procesador (7) está configurado para ajustar dicho contador de ignorar en dicho mensaje de datos en base a dicho cómputo de pujadores perdedores antes de reenviar dicho mensaje de datos a dicho nodo de más abajo (1₄) si dicho nodo de red (1₅) ganó dicha puja.
- 50 9. Un nodo de red (1₅) según cualquiera de las reivindicaciones anteriores, en el que dicho por lo menos un procesador (7) está configurado para determinar si pujar para almacenar dicho objeto de datos solicitado en dicha por lo menos una memoria (5) mediante comprobar si por lo menos está disponible una posición vacía en dicha por

lo menos una memoria (5) y/o si por lo menos un elemento almacenado en caché recientemente es más antiguo que un determinado umbral de antigüedad.

5 10. Un nodo de red (1₅) según cualquiera de las reivindicaciones anteriores, en el que dicho por lo menos un procesador (7) está configurado para determinar si pujar para almacenar dicho objeto de datos solicitado en dicha por lo menos una memoria (5) mediante comprobar si dicho valor de puja calculado representa una puja que supera un determinado umbral de subasta.

11. Un nodo terminal (330, 31), que comprende:

por lo menos una interfaz de comunicación (331); y

por lo menos un procesador (335) configurado:

10 - para determinar un nombre de un objeto de datos solicitado,

- para crear un mensaje de interés que comprende dicho nombre de dicho objeto de datos solicitado y un campo de puja para introducir un valor de puja, dejándose sin definir dicho valor de puja para indicar que debería comenzar una subasta del derecho almacenar en caché dicho objeto de datos solicitado, y

15 - para transmitir dicho mensaje de interés a un nodo de más arriba (1₁) sobre dicha por lo menos una interfaz de comunicación (331).

12. Un procedimiento de recepción de un mensaje de interés, que comprende las etapas de:

-recibir (11) un mensaje de interés sobre por lo menos una interfaz de comunicación desde un nodo de más abajo, comprendiendo dicho mensaje de interés un nombre de un objeto de datos solicitado;

- determinar (13) si pujar para almacenar dicho objeto de datos solicitado en por lo menos una memoria;

20 - calcular (15) un valor de puja;

- reenviar (17) dicho mensaje de interés a un nodo de más arriba sobre dicha por lo menos una interfaz de comunicación, incluyendo el mensaje de interés reenviado dicho valor de puja calculado en función de por lo menos dicha determinación de si pujar;

25 - recibir (19) un mensaje de datos que comprende dicho objeto de datos solicitado desde dicho nodo de más arriba sobre dicha por lo menos una interfaz de comunicación;

- determinar (21) a partir de dicho mensaje de datos si dicho nodo de red ganó dicha puja;

- almacenar (23) dicho objeto de datos en dicha por lo menos una memoria si dicho nodo de red ganó dicha puja; y

- reenviar (25) dicho mensaje de datos a dicho nodo de más abajo sobre dicha por lo menos una interfaz de comunicación.

30 13. Un programa informático o conjunto de programas informáticos que comprende por lo menos una parte de código de software o un producto de programa informático que almacena por lo menos una parte de código de software, estando configurada la parte de código de software para, cuando se ejecuta en un sistema informático, llevar a cabo el procedimiento de la reivindicación 12.

35

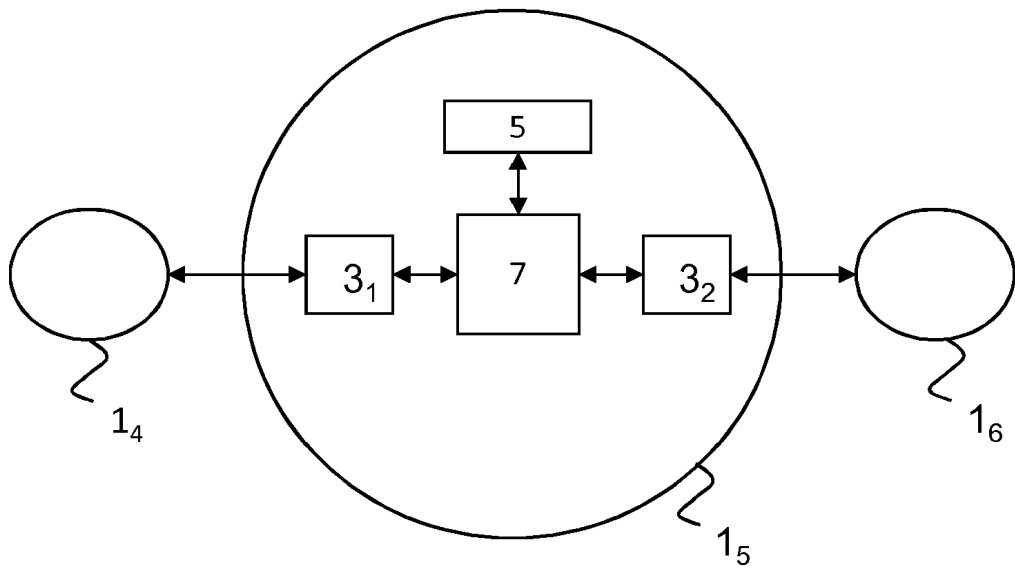


Fig. 1

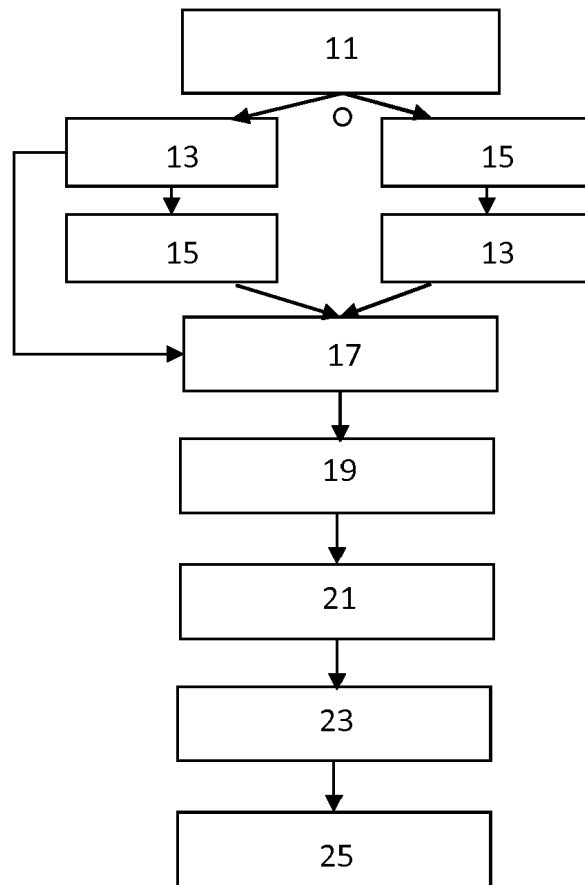


Fig. 2

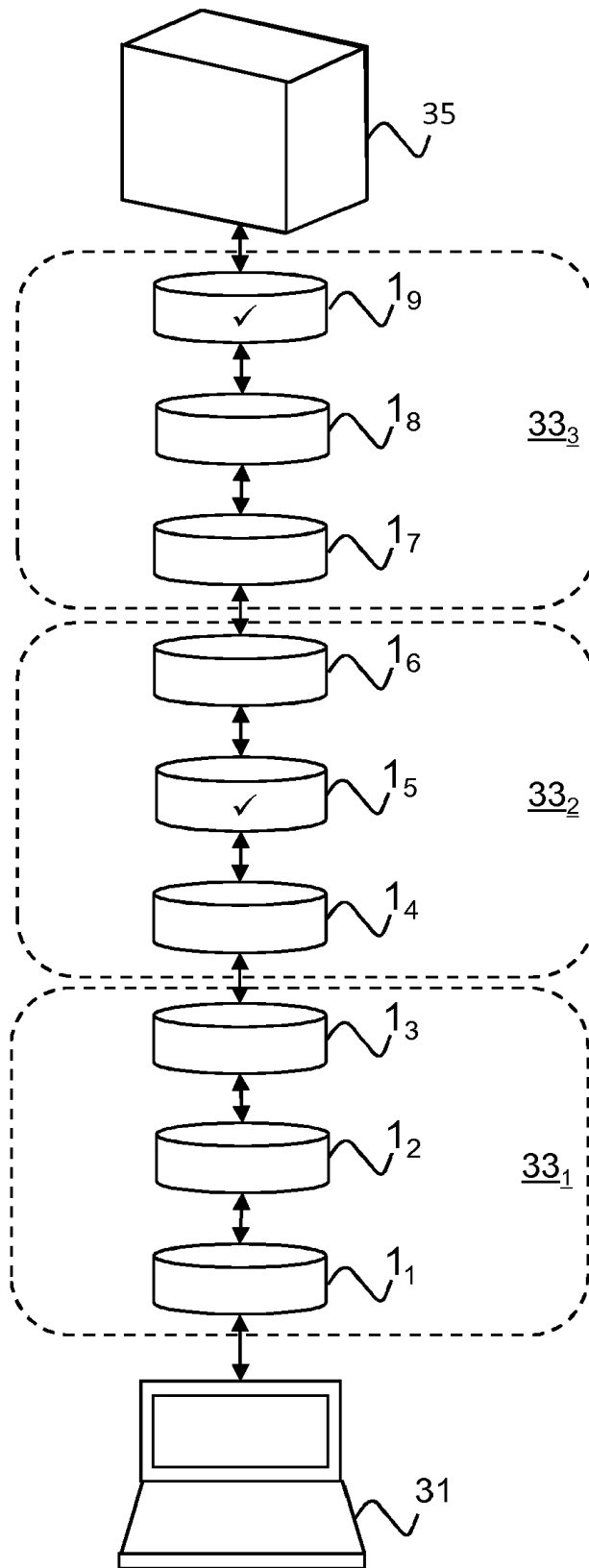


Fig. 3

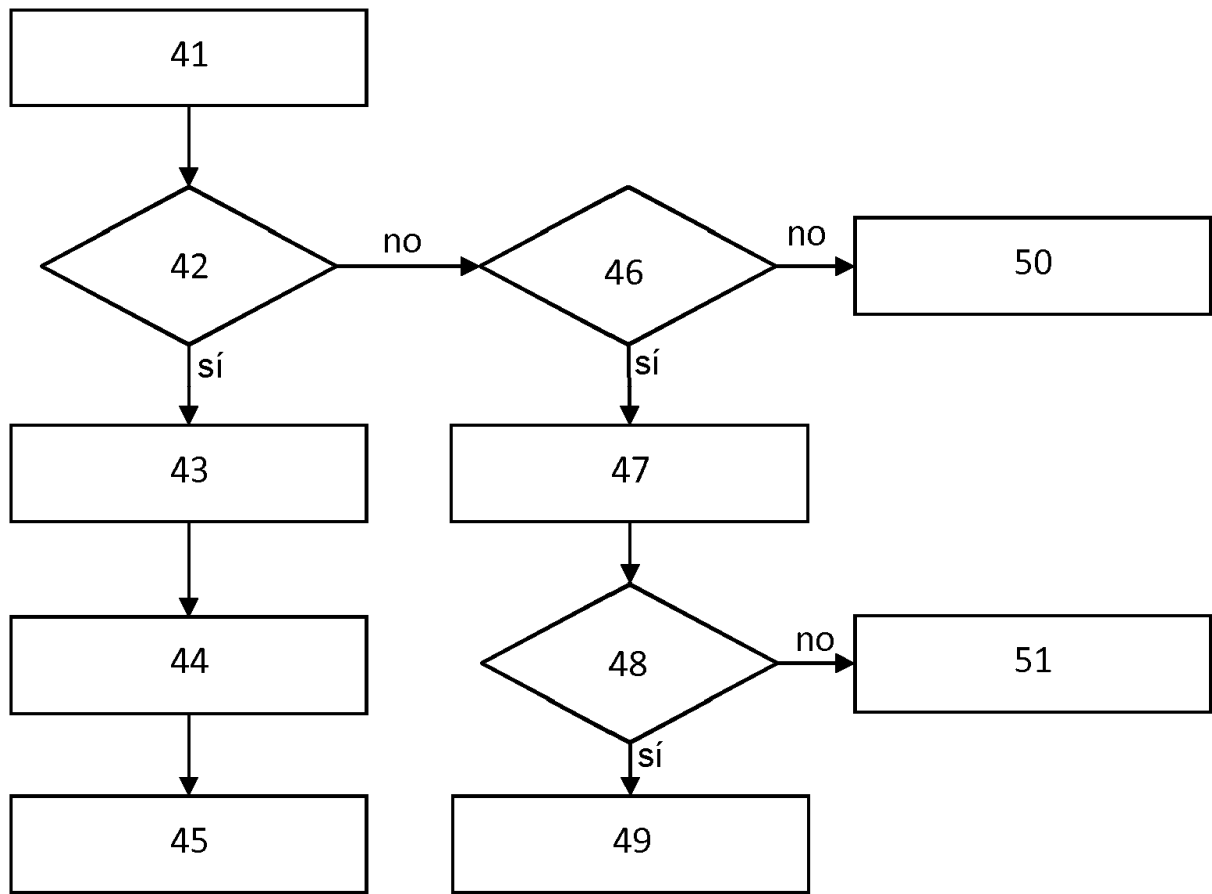


Fig. 4

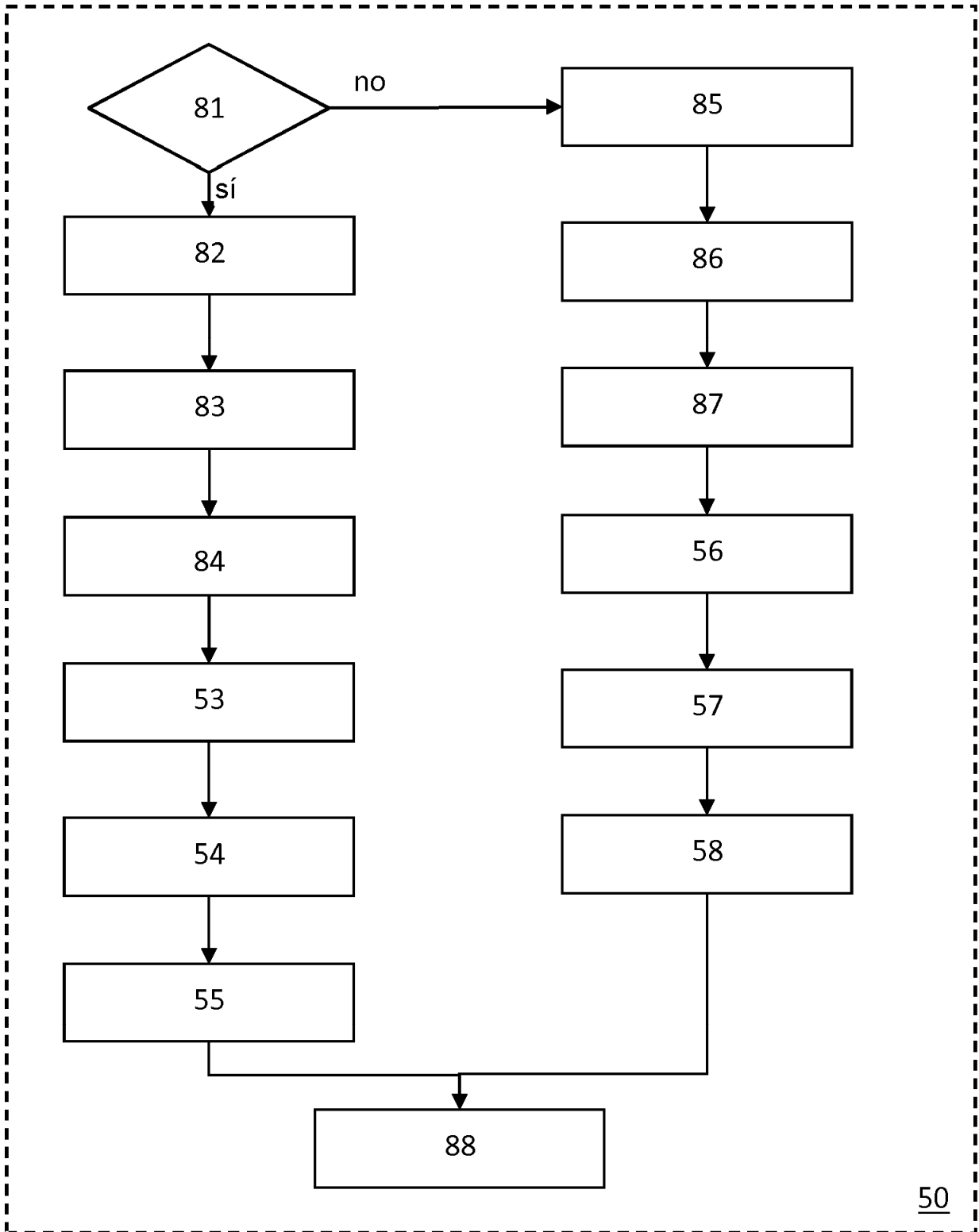
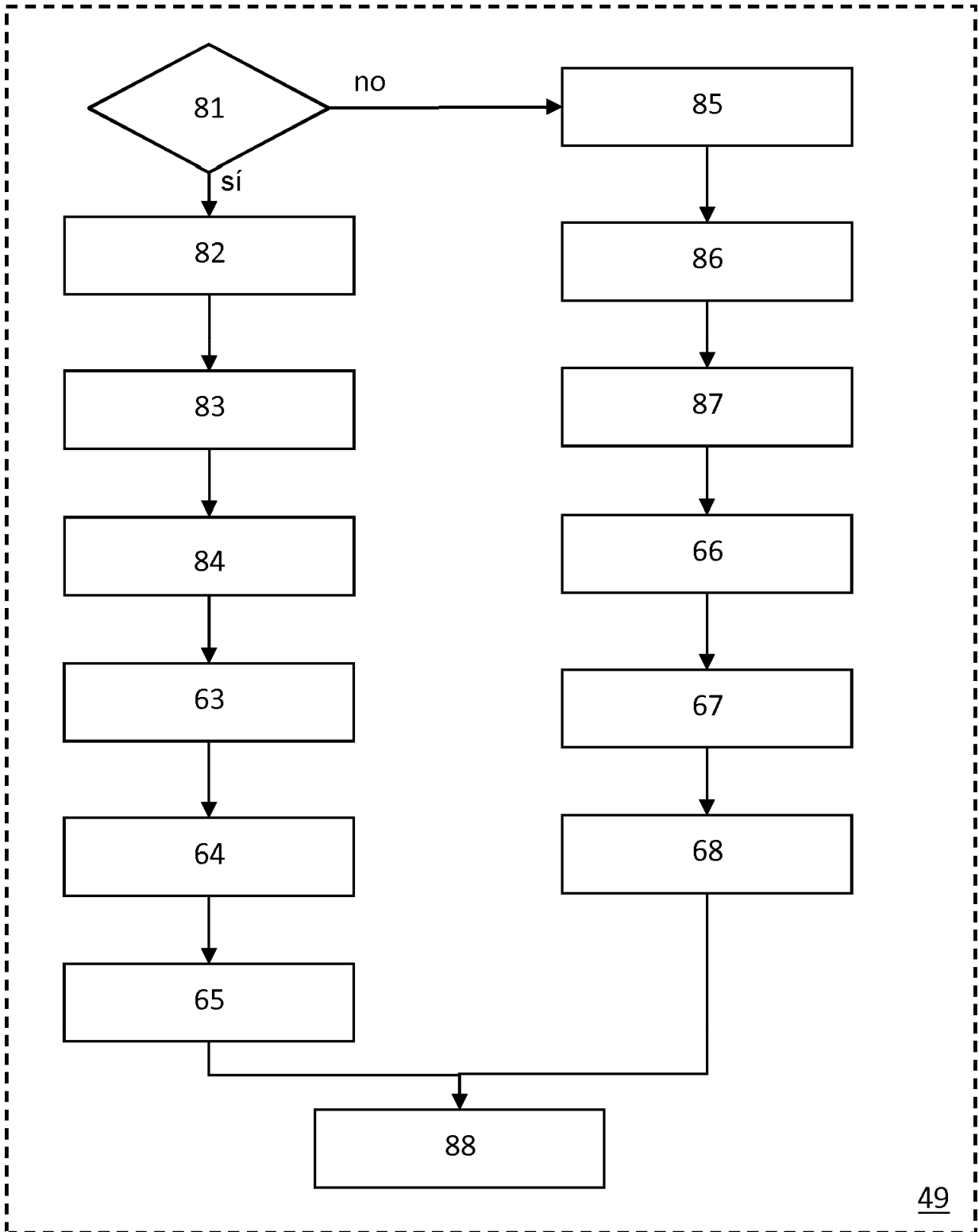


Fig. 5



49

Fig. 6

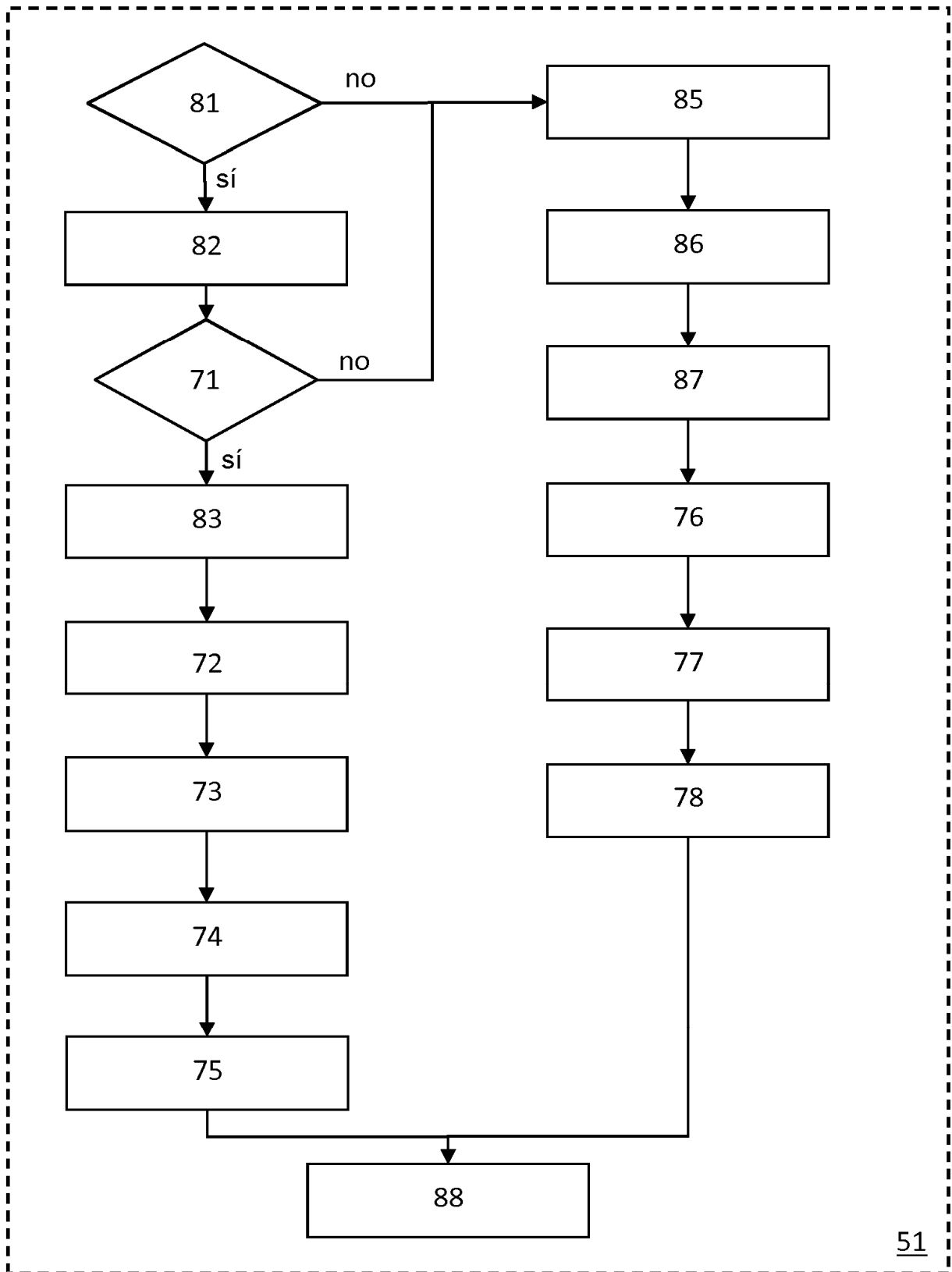


Fig. 7

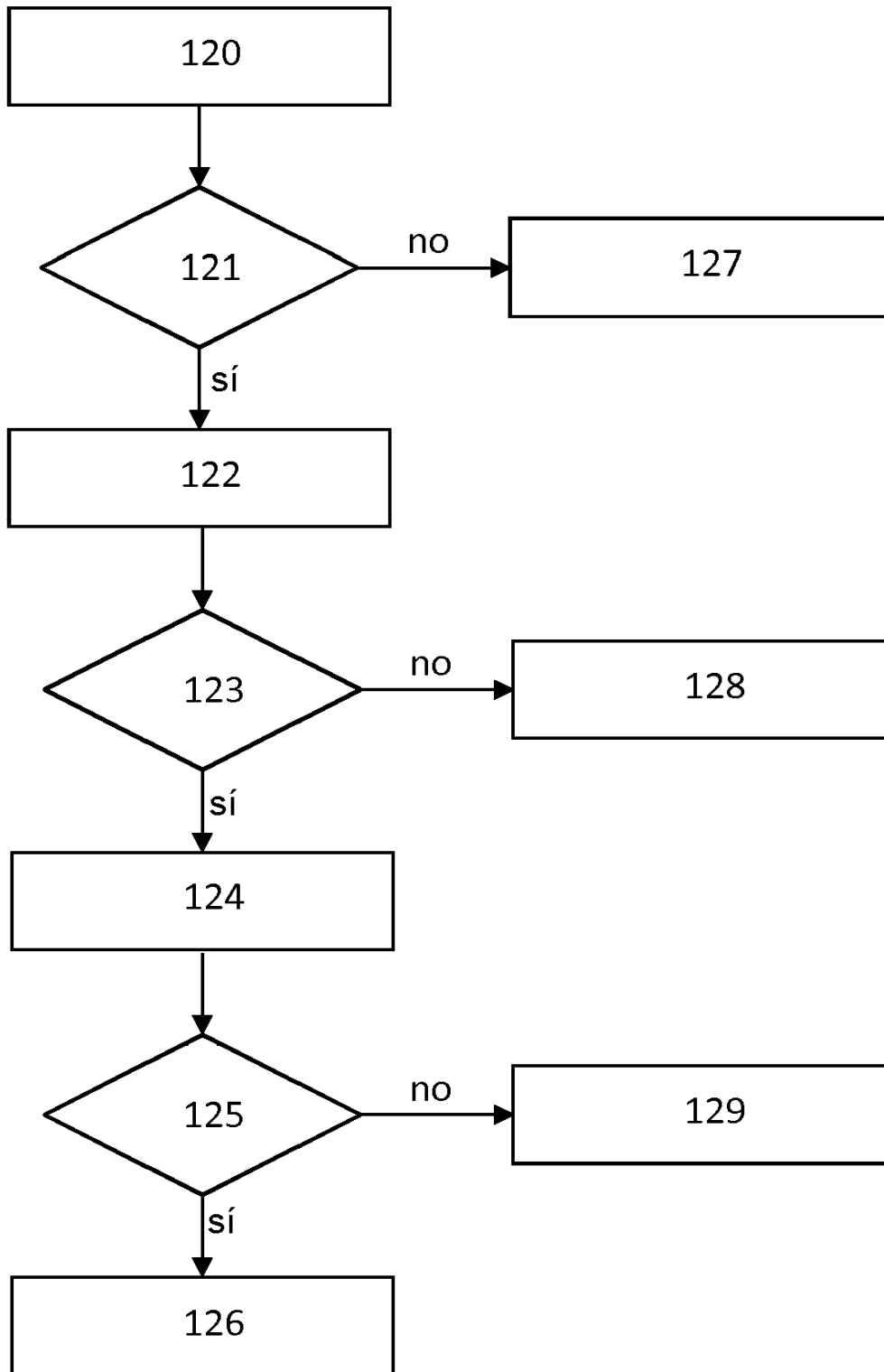
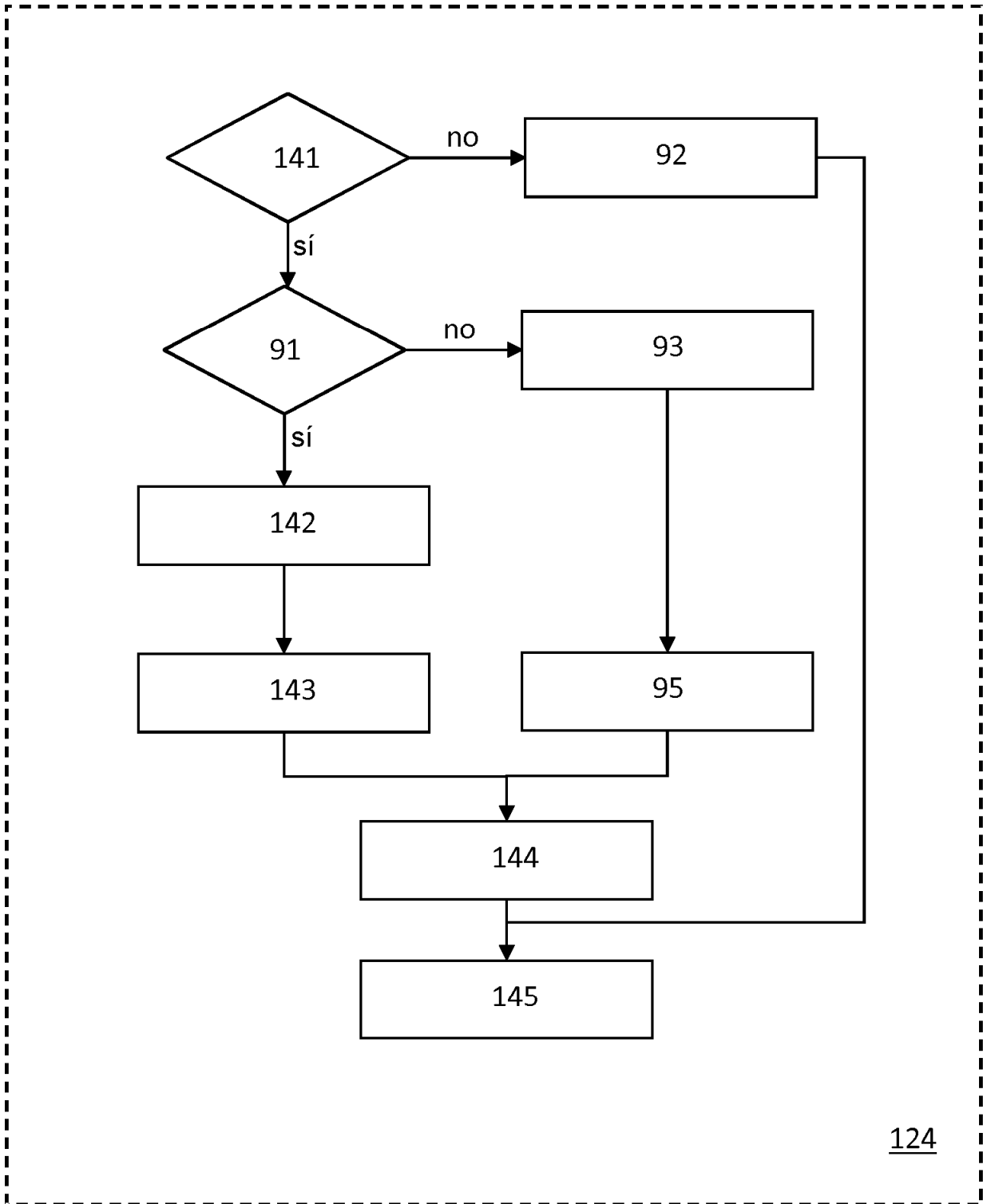
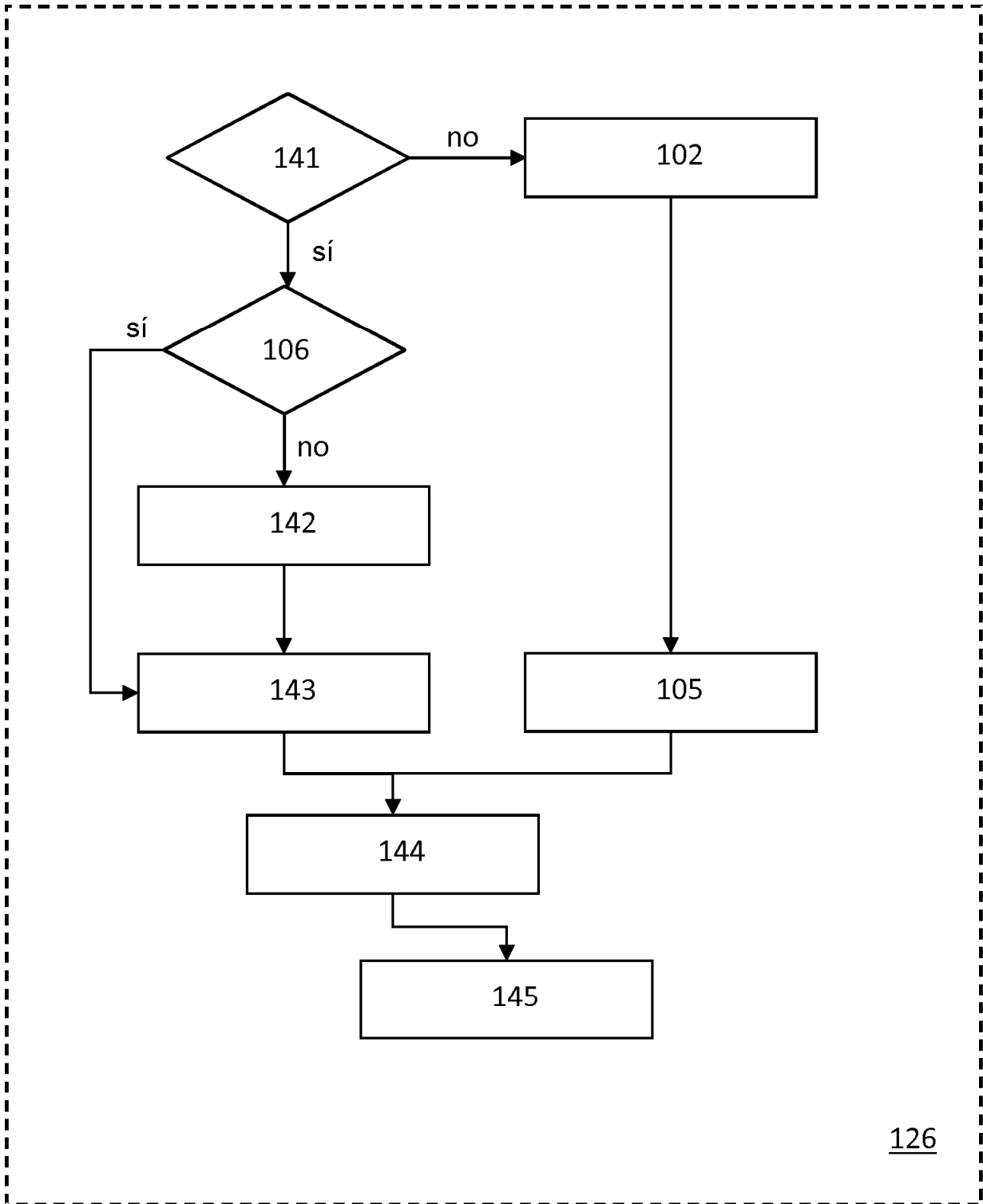


Fig. 8



124

Fig. 9



126

Fig. 10

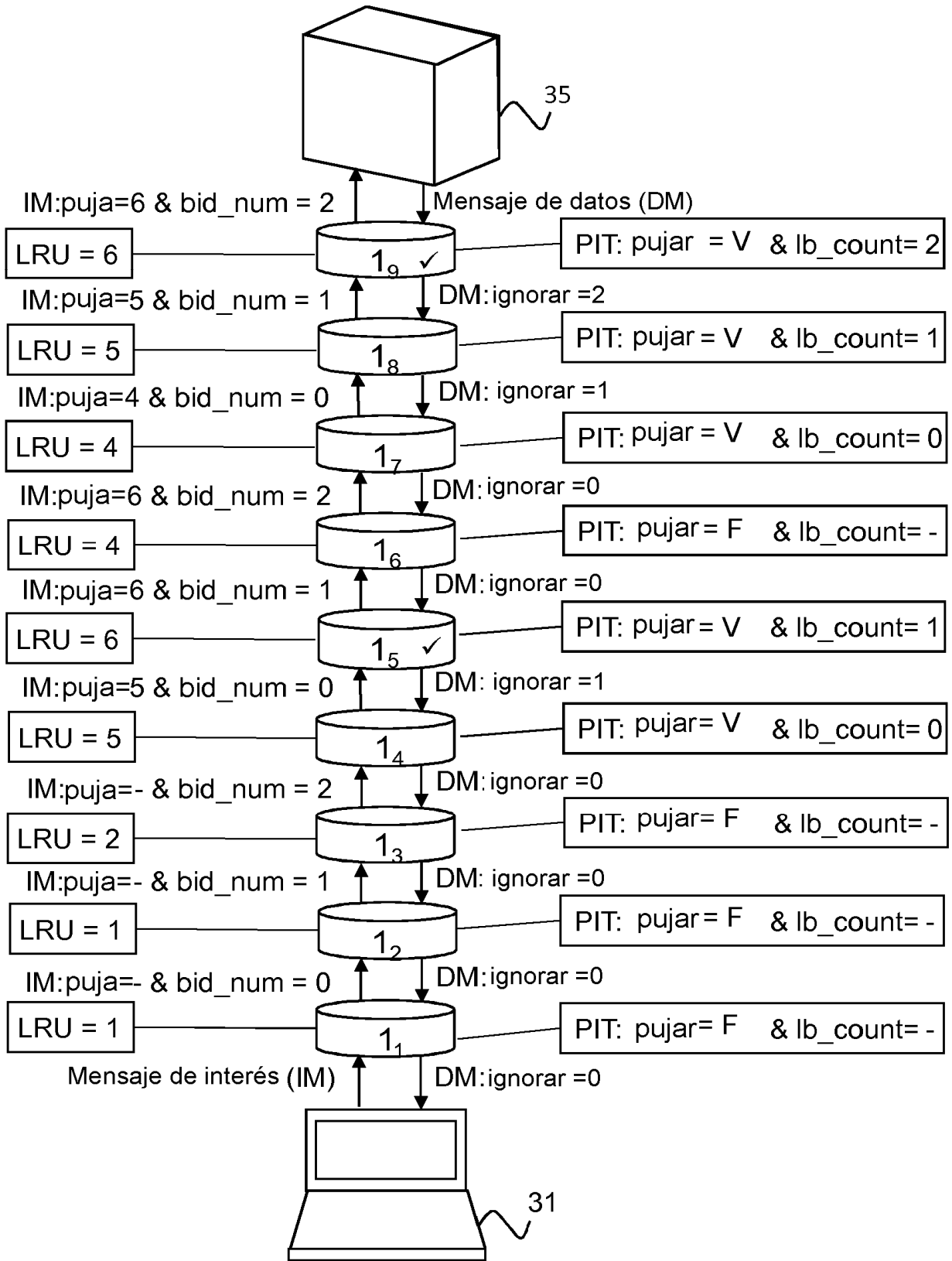


Fig. 11

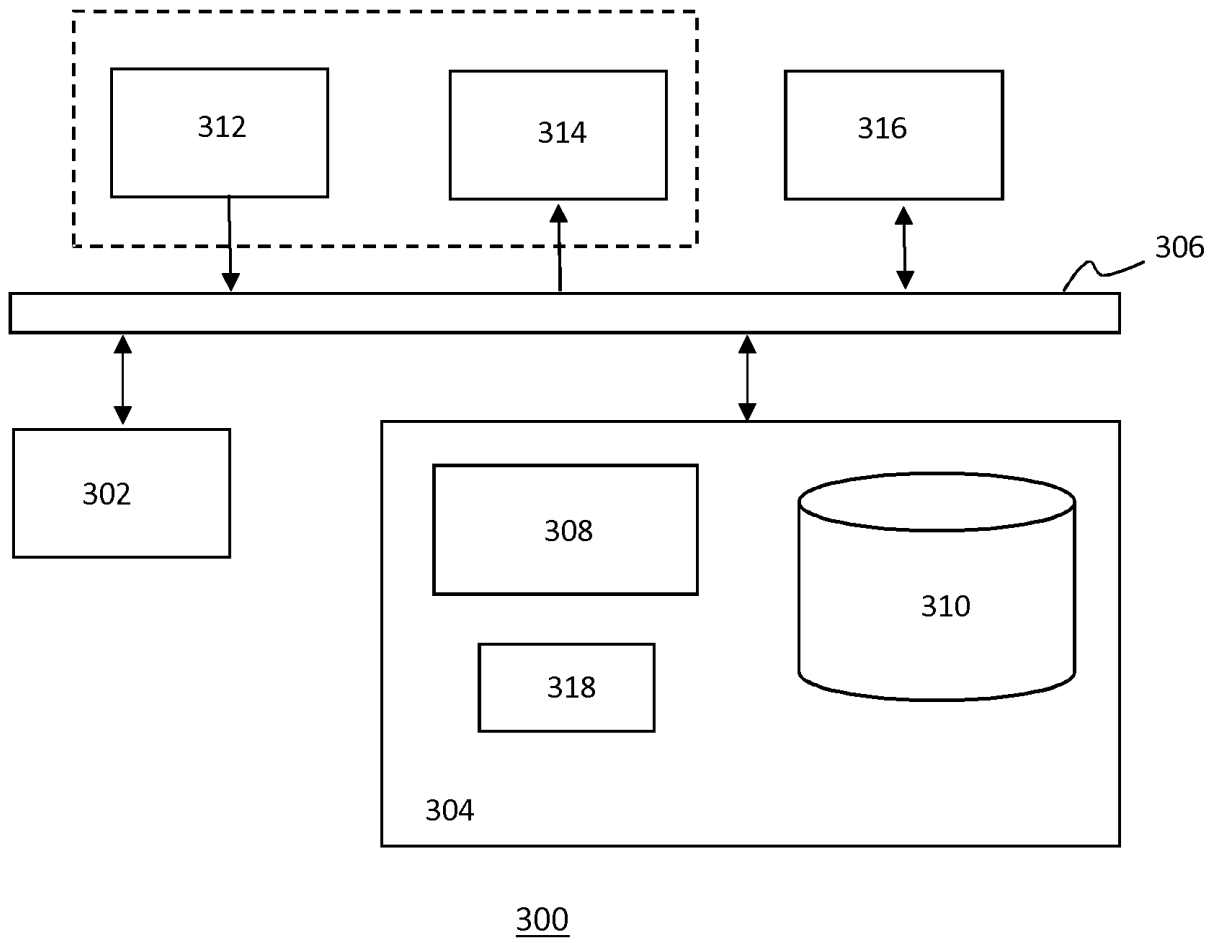


Fig. 12

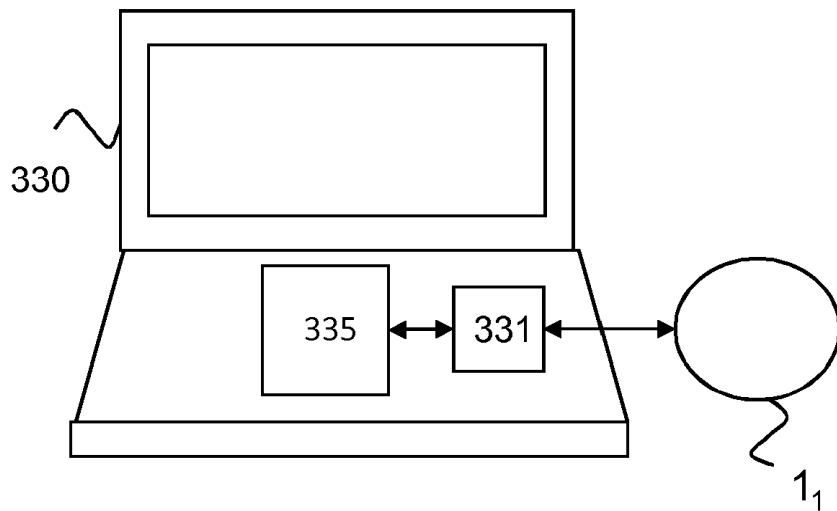


Fig. 13