

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 469 818**

51 Int. Cl.:

**G06F 17/30** (2006.01)

**H03M 13/37** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **23.09.2010** **E 10010280 (5)**

97 Fecha y número de publicación de la concesión europea: **26.03.2014** **EP 2434413**

54 Título: **Método y sistema informático distribuido para sincronizar los conjuntos de datos memorizados en dispositivos de comunicación diferentes**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**20.06.2014**

73 Titular/es:

**DEUTSCHE TELEKOM AG (100.0%)**  
**Friedrich-Ebert-Allee 140**  
**53113 Bonn, DE**

72 Inventor/es:

**SACHIN KUMAR AGARWAL**

74 Agente/Representante:

**LEHMANN NOVO, María Isabel**

**ES 2 469 818 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**

Método y sistema informático distribuido para sincronizar los conjuntos de datos memorizados en dispositivos de comunicación diferentes

5

**CAMPO DE LA INVENCION**

La presente invención se refiere a un método y un sistema informático distribuido para sincronizar conjuntos de datos memorizados en diferentes dispositivos de comunicación.

10

**ANTECEDENTES DE LA INVENCION**

Un sistema informático distribuido comprende una pluralidad de ordenadores geográficamente distribuidos tales como PDAs, ordenadores portátiles o teléfonos móviles, que están conectados, de forma intermitente, por intermedio de una tecnología de transmisión digital. Dichos sistemas informáticos distribuidos comparten datos comunes entre los ordenadores geográficamente distribuidos y los datos pueden modificarse con independencia de los ordenadores durante la utilización del conjunto de datos por aplicaciones, pero puede producirse un gran solapamiento entre los elementos de conjuntos de datos de los conjuntos de datos. Por lo tanto, es necesario sincronizar estos datos a través de los diferentes ordenadores, de vez en cuando, con el fin de hacer volver el sistema informático distribuido a un estado sincronizado. La sincronización frecuente es un proceso que necesita una gran cantidad de recursos en términos de comunicación y de cálculo y por lo tanto, está limitado a solicitudes periódicas. El periodo de tiempo entre las sincronizaciones de datos es un parámetro dependiente de la aplicación y puede variar desde el tiempo real (fracción de un segundo) a una réplica "perezosa" (cada pocos meses). En algunas instancias operativas, el periodo de tiempo entre sincronizaciones de datos se deja en extremo abierto (sin periodicidad fijada) y la sincronización de datos se solicita manualmente cuando así se desea por los usuarios del sistema distribuido. Es importante disponer de sistemas de sincronización de datos eficientes con el fin de reducir al mínimo los costes de la sincronización de datos repetida en sistemas distribuidos. La sincronización de datos eficiente puede estimular también un mayor grado de sincronización entre ordenadores distribuidos y de este modo, aumentar la eficiencia de un sistema informático distribuido.

15

20

25

30

La sincronización de datos es un proceso de sincronización, a modo de ejemplo, de dos conjuntos de datos similares memorizados en dos ordenadores diferentes conectados por intermedio de un enlace de comunicación. Un conjunto de datos es un conjunto no ordenado de elementos de información digital que pueden memorizarse en un dispositivo de comunicación digital, también aquí denominado un concentrador o host. Dicho dispositivo de comunicación comprende medios de cálculo, de memorización de datos y de comunicación digital. El resultado final del procedimiento de sincronización de datos debe ser copias idénticas de los conjuntos de datos en ambos ordenadores.

35

40

El documento de S. Agarwal et al.: "Difusión de actualización casi óptima de conjuntos de datos", PROCEEDINGS, IEEE 8TH INTERNATIONAL CONFERENCE ON MOBILE DATA MANAGEMENT, 27 de enero de 2007 – 1 de febrero de 2007, páginas 356 – 360, XP002646583, es una contribución de conferencia que se refiere a la difusión de actualizaciones incrementales desde una fuente de datos a múltiples terminales que contienen copias fuera de uso (y posiblemente distintas) de los datos. En esta descripción, se ha utilizado un algoritmo de reconciliación de conjuntos para una codificación sin tasas. Un protocolo de dos canales se utiliza en el que los nuevos elementos de fuente se determinan a través de un canal y se utilizan para la decodificación sin tasa en un canal secundario.

45

El documento de D. Starobinski et al: "Sincronización de PDA eficiente", IEEE TRANSACTIONS ON MOBILE COMPUTING, vol. 2, nº 1, 1 enero 2003, páginas 40-51, XP001171684, es una publicación científica que describe un sistema de sincronización para redes heterogéneas de numerosas máquinas tales como ordenadores PCs y PDAs. En este caso, los datos se convierten a un determinado tipo de polinomio conocido como el polinomio característico. Cuando se sincroniza, el PDA envía valores muestreados de su polinomio característico al ordenador PC.

50

El documento de S. Agarwal: "Reconciliación eficiente de datos no estructurados y estructurados a través de redes", Boston University, febrero 2006, XP002646584, es un trabajo de disertación que se refiere a un sistema de sincronización de datos cuya propiedad más destacada es que su complejidad de comunicación depende del número de diferencias entre el PDA y el PC. Este método implica la transformación de una cadena en un multiconjunto de subcadenas que se reconcilian eficientemente utilizando los algoritmos de reconciliación establecidos y luego, dispuestos juntos en un host remoto utilizando las recientes técnicas de gráficos teóricos.

55

60

Es un objetivo de la presente invención hacer disponible un nuevo método y un sistema informático distribuido para sincronizar conjuntos de datos memorizados en diferentes dispositivos de comunicación que dan lugar a un intercambio de datos reducido entre dos dispositivos de comunicación durante el procedimiento de sincronización de datos.

65

Es un aspecto, además, de la presente invención dar a conocer garantías de protección de datos cuando se

sincronizan conjuntos de datos que se transmiten a través de canales propensos a errores/pérdidas, esto es, enlaces de comunicaciones.

5 Por lo tanto, la carga informática del procedimiento puede reducirse de modo que el procedimiento de sincronización pueda completarse en una pequeña cantidad de tiempo finita. La más baja complejidad informática durante el proceso de sincronización conserva también energía que es un recurso limitado en los dispositivos de comunicación accionados por baterías, tales como teléfonos móviles. Además, el proceso de sincronización de datos se realizará aún cuando el canal de comunicación entre los dos dispositivos de comunicación pierda algunos de los paquetes de datos de comunicaciones. Esta medición de la invención hace al proceso de sincronización de datos más sólido y  
10 tolerante a fallos en comparación con simplemente copiar los conjuntos de datos desde un dispositivo de comunicación al otro.

Es un aspecto adicional de la presente invención dar a conocer una valoración para reducir la complejidad informática desfavorable en la etapa de decodificación de un dispositivo de comunicación.

15 SUMARIO DE LA INVENCIÓN

Es un aspecto esencial de la presente invención utilizar un código de corrección de errores hacia delante para la sincronización de datos así como una transmisión de datos resistente a errores durante el proceso de sincronización de datos. En particular, se utiliza una tecnología de códigos lineales aleatorios que se emplea tradicionalmente para la detección y corrección de errores.

20 Con el fin de sincronizar grandes conjuntos de datos utilizando códigos lineales aleatorios, se realiza una partición de conjuntos en los grandes conjuntos de datos. Por lo tanto, un sistema informático distribuido se hace escalable dividiendo la realización de sincronización de datos en numerosas tareas de sincronización más pequeñas. Esta disposición da lugar a una complejidad informática notablemente menor en la decodificación del dispositivo de comunicación, permitiendo que la sincronización de datos se realice con mucha mayor rapidez (menor tiempo de cálculo del sistema) y también ahorra recursos de comunicación entre los dos dispositivos de comunicación en sincronización.

30 El problema técnico anteriormente descrito se resuelve por las etapas del método según la reivindicación 1.

Por lo tanto, un método para sincronizar conjuntos de datos memorizados en diferentes dispositivos de comunicación, que están conectados mediante enlaces de comunicación, se hizo disponible. El método comprende las etapas siguientes:

- 35 1a) Memorizar un primer conjunto de datos en un primer dispositivo de comunicación, comprendiendo dicho primer conjunto de datos un número  $m$  de elementos;
- 40 1b) Memorizar un segundo conjunto de datos en un segundo dispositivo de comunicación, comprendiendo dicho segundo conjunto de datos un número  $n$  de elementos, en donde  $n$  puede ser igual o mayor que  $m$ ;
- 45 1c) Codificación, en el primer dispositivo de comunicación, del primer conjunto de datos utilizando un código lineal aleatorio para obtener un número de  $k$  elementos codificados, en donde  $k$  es un parámetro ajustable;
- 50 1d) Determinación, en el segundo dispositivo de comunicación, del número total de elementos, que son diferentes entre dicho primero y segundo conjunto de datos, sobre la base de un mensaje recibido desde el primer dispositivo de comunicación;
- 55 1e) Determinación, en el segundo dispositivo de comunicación, de si el número total de diferentes elementos es menor que o igual al parámetro  $k$ ;
- 1f) Si la respuesta es afirmativa, entonces transmitir los  $k$  elementos codificados desde el primer dispositivo de comunicación al segundo dispositivo de comunicación y
- 1g) Decodificación, en el segundo dispositivo de comunicación, de al menos algunos de los elementos codificados recibidos utilizando un decodificador lineal aleatorio con el fin de obtener todos los elementos del primer conjunto de datos, que no estén contenidos en el segundo conjunto de datos.

60 Conviene señalar que el parámetro  $k$  puede ser menor que  $m$ .

Sin embargo, los códigos lineales aleatorios son muy poco prácticos y requieren una gran cantidad de recursos informáticos cuando los conjuntos de datos han de sincronizarse y el número de diferencias entre los conjuntos de datos son grandes. Con el fin de reducir la complejidad informática de códigos lineales aleatorios, se da a conocer un nuevo procedimiento de codificación y de decodificación que están basados en un método de división y resolución. Este método permite un ahorro rápido, eficiente y sólido en el procesamiento informático en el segundo

dispositivo de comunicación que funciona como un dispositivo de decodificación. El método de dividir y resolver consigue estas propiedades *inter alia* utilizando una comunicación limitada entre el primero y el segundo dispositivo de comunicación.

5 Por lo tanto, el método comprende, además, las etapas siguientes:

2a) Realizar una partición de conjuntos en el primer conjunto de datos con el fin de obtener una pluralidad de primeros subconjuntos de datos;

10 2b) Realizar una partición de conjuntos en el segundo conjunto de datos con el fin de obtener una pluralidad de segundos subconjuntos de datos, en donde cada uno de los primeros subconjuntos de datos corresponden a uno separado de los segundos subconjuntos de datos;

15 2c) Codificación, en el primer dispositivo de comunicación, de cada primer subconjunto de datos utilizando un código lineal aleatorio con el fin de obtener un número de  $k$  elementos codificados;

20 2d) Si el número de diferentes elementos determinados en la etapa 1d) es mayor que  $k$ , en tal caso seleccionar, en el segundo dispositivo de comunicación un segundo subconjuntos de datos del segundo conjunto de datos y seleccionar, en el primer dispositivo de comunicación, el primer subconjunto de datos correspondiente en respuesta a un mensaje recibido desde el segundo dispositivo de comunicación;

25 2e) Determinación, en el segundo dispositivo de comunicación, del número de elementos, que son diferentes entre el segundo subconjuntos de datos seleccionados y el primer subconjunto de datos correspondiente sobre la base de un mensaje recibido desde el primer dispositivo de comunicación;

2f) Determinación, en el segundo dispositivo de comunicación, de si el número de elementos diferentes es menor que o igual al parámetro  $k$ ;

30 2g) Si la respuesta es afirmativa, en tal caso transmitir los  $k$  elementos codificados del primer subconjunto de datos desde el primer dispositivo de comunicación al segundo dispositivo de comunicación y

35 2h) Decodificación, en el segundo dispositivo de comunicación, de al menos algunos de los elementos codificados recibidos utilizando un decodificador lineal aleatorio con el fin de obtener los elementos del primer subconjunto de datos seleccionado, que no estén contenidos en el segundo subconjunto de datos seleccionado del segundo conjunto de datos;

40 2i) Repetición de las etapas 2d) a 2h) hasta que todos los elementos del primer conjunto de datos, que no estaban inicialmente contenidos en el segundo conjunto de datos, hayan sido decodificados en el segundo dispositivo de comunicación.

Conviene señalar que el orden de ejecución de las etapas del método, según se describió anteriormente, no está fijado y pueden reordenarse si fuera adecuado.

45 La complejidad informática puede optimizarse cuando las etapas de partición de conjuntos dan lugar a una arborescencia multinivel jerárquica que comprende la pluralidad de primeros subconjuntos de datos y en una arborescencia multinivel jerárquica similar, que comprende la pluralidad de segundos subconjuntos de datos.

50 En una forma de realización preferida, el procedimiento de codificación de códigos lineales aleatorios ejecuta la etapa 1c) que comprende la etapa adicional de crear una matriz de números aleatorios de  $k \times m$ -Beta, en donde dichos  $m$  elementos del primer conjunto de datos están representados por un vector de columna de  $m$  elementos ( $V_A$ ) y dichos  $k$  elementos codificados están representados por un vector de columna de  $k$  elementos ( $V_Y$ ), que se calcula realizando la multiplicación matricial de la matriz de números aleatorios Beta y el vector de columna de  $m$  elementos; en donde antes de realizar la etapa 1f), cada uno de los  $k$  elementos codificados se encapsula junto con las entradas de la respectiva fila de la matriz de números aleatorios Beta en un paquete separado a transmitirse. En la forma de realización preferida, el procedimiento de decodificación de códigos lineales aleatorios ejecuta la etapa 1g) que comprende la etapa de calcular la inversa de dicha matriz de números aleatorios Beta recibida, en donde el vector de columna de  $m$  elementos se realizando la multiplicación matricial de la inversa de dicha matriz de números aleatorios Beta y el vector de columna de  $k$  elementos.

60 Si se realiza una partición de conjuntos, la etapa 2c) preferentemente comprende la etapa adicional de crear con respecto a cada primer subconjunto de datos una matriz de números aleatorios de  $k \times l_i$ -Beta en donde  $l_i$  corresponde al número de elementos del  $i$ -ésimo primer subconjunto de datos y  $l_i$  es menor que  $m$ , en donde dichos  $l_i$  elementos del  $i$ -ésimo primer subconjunto de datos se representan por un vector de columna de  $l_i$  elementos y dichos  $k$  elementos codificados están representados por un vector de columna de  $k$  elementos, que se calcula realizando la multiplicación matricial de la matriz de números aleatorios Beta y el vector de columna de  $l_i$  elementos, en donde antes de realizar la etapa 2g), cada uno de los  $k$  elementos codificados se encapsula junto con las entradas de la

respectiva fila de la matriz de números aleatorios Beta en un paquete separado a transmitirse. Además, la etapa 2h) comprende la etapa de calcular la inversa de la matriz de números aleatorios de  $k \times l_i$  -Beta recibida, en donde el vector de columna de  $l_i$  elementos se calcula realizando la multiplicación matricial de la inversa de dicha matriz de números aleatorios Beta y del vector de columna de  $k$  elementos.

5 Con el fin de reducir la complejidad informática en la etapa de decodificación, es conveniente que antes de realizar la etapa de decodificación 1g), los elementos comunes entre el primer conjunto de datos y el segundo conjunto de datos se determine y que solamente los elementos codificados recibidos que no estén incluidos en el segundo conjunto de datos sean objeto de decodificación y/o que antes de realizar la etapa de decodificación 2h), los  
10 elementos comunes entre el segundo subconjunto de datos seleccionado y el primer subconjunto de datos correspondiente se determine y que solamente los elementos codificados recibidos que no estén incluidos en el subconjunto de datos seleccionado del segundo subconjunto de datos sean decodificados.

15 Con el fin de determinar, en el segundo dispositivo de comunicación, el número total de elementos, que son diferentes entre dicha primera y segunda etapa de conjuntos de datos 2d) comprende ventajosamente las etapas siguientes:

20 transmitir una demanda desde el segundo dispositivo de comunicación al primer dispositivo de comunicación para enviar un mensaje relativo a todos los elementos incluidos en el primer conjunto de datos; transmitir el mensaje en un paquete único al segundo dispositivo de comunicación y en donde

la etapa 2e) comprende ventajosamente las etapas siguientes:

25 transmitir una demanda desde el segundo dispositivo de comunicación al primer dispositivo de comunicación para enviar un mensaje con respecto a todos los elementos incluidos en el primer subconjunto de datos seleccionado;

transmitir el mensaje respectivo en un paquete único al segundo dispositivo de comunicación.

30 Con el fin de reducir la cantidad de datos a transmitirse, el mensaje comprende las *hashes* de cada elemento incluido en el primer conjunto de datos y/o las *hashes* de cada elemento del primer subconjunto de datos seleccionado. La etapa 1d) comprende, entonces, las etapas adicionales de:

35 la generación, en el segundo dispositivo de comunicación, de la función aleatoria *hash* de cada elemento incluido en el segundo conjunto de datos;

40 la comparación de las *hashes* de cada elemento incluido en el primer conjunto de datos recibido desde el primer dispositivo de comunicación con las *hashes* de los elementos del segundo conjunto de datos con el fin de determinar el número total de elementos del primer conjunto de datos que no estén incluidos en el segundo conjunto de datos y en donde la etapa 2e) comprende las etapas adicionales de:

45 la generación, en el segundo dispositivo de comunicación de la función *hash* de cada elemento incluido en el segundo subconjunto de datos seleccionado; la comparación de las *hashes* de cada elemento del primer subconjunto de datos seleccionado recibido desde el primer dispositivo de comunicación con las *hashes* de los elementos del segundo subconjunto de datos seleccionado con el fin de determinar el número de elementos del primer subconjunto de datos seleccionado que no estén incluidos en el segundo subconjunto de datos seleccionado.

Una forma de realización preferida comprende la etapa de demandar al primer dispositivo de comunicación para el envío de los  $k$  elementos codificados.

50 Con el fin de optimizar la complejidad informática del proceso de codificación y de decodificación, el método puede comprender las siguientes etapas de:

55 determinar, en el segundo dispositivo de comunicación, si el número de elementos del segundo conjunto de datos o el número de elementos de un segundo subconjunto de datos seleccionado es menor que o igual al parámetro  $k$ ;

60 si la respuesta es afirmativa, en tal caso, demandar al primer dispositivo de comunicación para transmitir todos los elementos del primer conjunto de datos o para transmitir todos los elementos del primer subconjunto de datos seleccionado, que corresponde al segundo subconjunto de datos seleccionado, al segundo dispositivo de comunicación.

65 El problema técnico antes citado se resuelve también mediante un sistema informático distribuido para sincronizar conjuntos de datos memorizados en diferentes dispositivos de comunicación que están conectados por intermedio de enlaces de comunicación. Ese sistema comprende un primer dispositivo de comunicación que comprende las características operativas siguientes:

- primer medio de memorización para memorizar un primer conjunto de datos, incluyendo dicho primer conjunto

de datos un número de m elementos,

- un codificador para codificar el primer conjunto de datos utilizando un código lineal aleatorio para obtener un número de k elementos codificados, en donde k es un parámetro ajustable,
- un divisor para transmitir los k elementos codificados al segundo dispositivo de comunicación en respuesta a una demanda.

El sistema comprende, además, un segundo dispositivo de comunicación conectado mediante un enlace de comunicación con el primer dispositivo de comunicación. El segundo dispositivo de comunicación comprende las características operativas siguientes:

- un primer medio de memorización para memorizar un segundo conjunto de datos, incluyendo dicho segundo conjunto de datos un número de n elementos;
- un medio para determinar el número total de elementos, que son diferentes entre dicho primero y segundo conjunto de datos, sobre la base de un mensaje recibido desde el primer dispositivo de comunicación y para determinar si el número total de elementos diferentes es menor que o igual al parámetro k,
- un decodificador para decodificar al menos algunos de los elementos codificados recibidos utilizando un decodificador lineal aleatorio con el fin de obtener todos los elementos del primer conjunto de datos, que no estén contenidos en el segundo conjunto de datos.

En una forma de realización preferida, el primer dispositivo de comunicación comprende un segundo medio de memorización para memorizar los elementos codificados.

Sin embargo, los códigos lineales aleatorios son muy poco prácticos y requieren grandes recursos informáticos cuando han de sincronizarse los conjuntos de datos y el número de diferencias entre los conjuntos de datos son grandes. Con el fin de reducir la complejidad informática de códigos lineales aleatorios, se da a conocer un nuevo procedimiento de codificación y de decodificación que se basa en un método de dividir y resolver.

Por lo tanto, el primer dispositivo de comunicación comprende, además, en una forma de realización preferida, un medio para realizar una partición de conjuntos en el primer conjunto de datos con el fin de obtener una pluralidad de primeros subconjuntos de datos,

un medio para seleccionar un primer subconjunto de datos en respuesta a un mensaje recibido desde el segundo dispositivo de comunicación, en donde el codificador está adaptado para codificar cada primer subconjunto de datos utilizando un código lineal aleatorio con el fin de obtener un número de k elementos codificados. Además, el segundo dispositivo de comunicación comprende preferentemente un medio para realizar una partición de conjunto en el segundo conjunto de datos con el fin de obtener una pluralidad de segundos subconjuntos de datos, en donde cada primer subconjunto de datos corresponde a un segundo subconjunto de datos separado del segundo conjunto de datos y un medio para seleccionar al menos un segundo subconjunto de datos en conformidad con una regla predeterminada.

En una forma de realización preferida, el codificador está adaptado para utilizar una matriz de números aleatorios de  $k \times m$ -Beta para codificar el primer conjunto de datos, en donde el primer dispositivo de comunicación comprende medios para encapsular cada uno de los k elementos codificados junto con las entradas de la respectiva fila de la matriz de números aleatorios Beta en un paquete separado. Además, el decodificador está adaptado para calcular la inversa de la matriz de números aleatorios Beta recibida.

Con el fin de sincronizar los subconjuntos de datos correspondientes, el codificador está adaptado para utilizar, para cada primer subconjunto de datos a codificarse, una matriz de números aleatorios de  $k \times l_i$ -Beta, en donde  $l_i$  corresponde al número de elementos del i-ésimo primer subconjunto de datos y  $l_i$  es menor que m, en donde el primer dispositivo de comunicación comprende medios para encapsular cada uno de los k elementos codificados de un respectivo primer subconjunto de datos junto con las entradas de la respectiva fila de la matriz de números aleatorios de  $k \times l_i$ -Beta en un paquete separado. El decodificador está adaptado para calcular la inversa de la matriz de números aleatorios de  $k \times l_i$ -Beta recibida.

#### BREVE DESCRIPCIÓN DE LOS DIBUJOS

La invención, junto con sus objetivos y sus ventajas puede entenderse mejor haciendo referencia a la siguiente descripción tomada en conjunción con los dibujos adjuntos, en donde:

La Figura 1 ilustra una forma de realización preferida de un sistema informático distribuido según la invención;

La Figura 2 ilustra un número de k paquetes codificados,

La Figura 3 ilustra dos estructuras de datos en arborescencia preferidas para el conjunto de datos  $S_A$  y el conjunto de datos  $S_B$ , respectivamente,

5 La Figura 4 ilustra un diagrama en donde el número de transmisiones se ilustra como una función del tamaño del conjunto de datos y

La Figura 5 representa un diagrama, en donde el número de transmisiones se ilustra como una función de la diferencia  $S_A - S_B$ .

10

#### DESCRIPCIÓN DETALLADA DE LA INVENCIÓN

15 La Figura 1 ilustra un sistema informático distribuido 1 en términos de una realización, a modo de ejemplo. El sistema 1 comprende al menos dos dispositivos de comunicación 10 y 20 que pueden ser teléfonos móviles, PDAs, ordenadores portátiles u otros dispositivos informáticos móviles o fijos. En esta forma de realización, el dispositivo de comunicación 10 es un ordenador y el dispositivo de comunicación 20 se representa por un teléfono móvil. Los dispositivos de comunicación 10 y 20 están conectados mediante un medio de comunicación inalámbrica 30 y se comunican entre sí utilizando, a modo de ejemplo, el denominado Bluetooth Standard u otras normas de comunicación inalámbrica. El ordenador 10 puede comprender una memoria 18 para memorizar los elementos de un conjunto de datos  $S_A$ , una memoria 17 para memorizar una estructura de datos basada en una arborescencia jerárquica del conjunto de datos  $S_A$ , un codificador 11 adaptado para crear varios paquetes codificados utilizados para la sincronización de datos en el teléfono móvil, una memoria 12, denominada depósito de paquetes, para memorizar los paquetes codificados, un decodificador 14 para decodificar un paquete codificado recibido desde el teléfono móvil 20 para realizar un procedimiento de sincronización de datos, un depósito de paquetes 15 para almacenar paquetes codificados recibidos desde el teléfono móvil 20, una etapa de transmisor/receptor 13 para transmitir mensajes y paquetes codificados a, y recibir mensajes y paquetes codificados desde, el teléfono móvil 20 y un microprocesador 16 que supervisa y controla los componentes del ordenador 10. Todos los componentes pueden conectarse entre sí.

20 En una manera similar, el teléfono móvil 20 puede comprender una memoria 28 para memorizar los elementos de un conjunto de datos  $S_B$ , una memoria 27 para memorizar la estructura de datos basada en una arborescencia jerárquica del conjunto de datos  $S_B$ , un codificador 21 adaptado para crear varios paquetes codificados para la sincronización de datos en el ordenador 10, una memoria 22, denominada depósito de paquetes, para almacenar los paquetes codificados, un decodificador 14 para decodificar el paquete codificado recibido desde el teléfono móvil 20 para realizar un procedimiento de sincronización de datos, un depósito de paquetes 25 para memorizar paquetes codificados recibidos desde el ordenador 10, una etapa de transmisor/receptor 23 para transmitir mensajes y paquetes codificados a, y recibir mensajes y paquetes desde, el ordenador 10 y un microprocesador 26 que supervisa y controla los componentes del teléfono móvil. Todos los componentes pueden estar conectados entre sí.

30 La sincronización de datos entre los dispositivos de comunicación 10 y 20 puede realizarse en un procedimiento de sincronización de datos unidireccional o bidireccional. En el proceso de sincronización de datos unidireccional, a modo de ejemplo desde el ordenador 10, al teléfono móvil 20, un conjunto de datos  $S_A$  almacenado en la memoria 18 del ordenador 10 se transfiere al teléfono móvil 20 y se utiliza para sincronizar un conjunto de datos  $S_B$  almacenado en la memoria 28. En un proceso de sincronización de datos bidireccional, el ordenador 10 y el teléfono móvil 20 tratarían de encontrar la unión de sus conjuntos de datos.

35 En esta forma de realización, un proceso de sincronización de datos unidireccional desde el ordenador 10 al teléfono móvil 20 se describe a continuación. Por lo tanto, el ordenador 10 puede referirse a un dispositivo de codificación y el teléfono móvil 20 puede referirse como un dispositivo de decodificación, aunque el dispositivo de comunicación 10 y el dispositivo de comunicación 20 están adaptados para realizar la codificación además de los procedimientos de decodificación. La sincronización de datos bidireccional sería simplemente una extensión directa de la sincronización unidireccional que se aplica una vez en cada dirección, esto es, desde el ordenador 10 al teléfono móvil 20 y viceversa.

40 En la configuración de sincronización de datos, los dos dispositivos de comunicación 10 y 20 se supone que no tienen ninguna información anterior sobre la existencia de cualquier escenario operativo particular anteriormente descrito. El sistema 1, descrito a continuación, es capaz de realizar una secuencia de etapas de comunicación y de cálculo escalables y eficientes con el fin de sincronizar el conjunto de datos  $S_A$  memorizado en el ordenador 10 y el conjunto de datos  $S_B$  memorizado en el teléfono móvil 20. En particular, el sistema informático distribuido 1 está adaptado para realizar un procedimiento de codificación y de decodificación lineal aleatorio para la sincronización de datos y para la corrección de errores. Por lo tanto, una visión general de los códigos lineales aleatorios y de los procedimientos de codificación y de decodificación aleatorios lineales se proporciona a continuación por ser importante para la invención.

45 50 55 60 65 1. Códigos lineales aleatorios

5 Los códigos lineales aleatorios son una clase de códigos, sin tasa, correctores de errores conocidos utilizados para la transmisión de datos a través de canales de comunicación con pérdidas y para memorización de datos fiable. Se utilizan para codificar información en paquetes antes de transmitir la información o de su memorización, y los paquetes son posteriormente decodificados por un receptor de la información. La codificación ofrece protección  
10 contra las pérdidas de transmisión de datos introduciendo información redundante, de modo que el receptor pueda reconstruir la información original a partir de los paquetes codificados recibidos.

En las dos secciones siguientes, introducimos alguna notación y describimos el proceso de codificación y el proceso de decodificación de códigos lineales aleatorios, sobre la base de descripciones estándar de la codificación lineal aleatoria en la documentación teórica de la codificación.

2. Proceso de codificación sobre la base de códigos lineales aleatorios

15 El proceso de codificación tradicional se aplica a continuación. Se supone que el ordenador 1 es capaz de ejecutar dicho procedimiento de codificación.

Habida cuenta de que es un conjunto de datos de m elementos  $S_A = \{x_1, x_2, \dots, x_m\}$ , el que se almacena en la memoria 18 del ordenador 10 y un parámetro k ajustable que especifica el número de elementos codificados a transmitir al teléfono móvil 20. El conjunto de datos  $S_A$  puede representarse por un vector de columna de m elementos  $V_A$  como sigue:

$$V_A = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

25 El codificador 11 del ordenador 10 está adaptado para proporcionar k elementos codificados que pueden representarse como un vector de columna  $V_Y = \{y_1, y_2, \dots, y_k\}$ , con k entradas, en donde k suele ser mayor que m. El codificador puede denominarse también un codificador de código lineal aleatorio. El procedimiento de codificación requiere una matriz beta  $\beta$  que se representa como una matriz de k x m según se ilustra a continuación:

$$\beta = \begin{bmatrix} \beta_{11} & \beta_{12} & \dots & \beta_{1m} \\ \beta_{21} & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{k1} & \dots & \dots & \beta_{km} \end{bmatrix}$$

30 Cada elemento de la matriz aleatoria Beta de k x m se selecciona para ser 0 o 1 con una probabilidad de 0.5. Un generador de números aleatorios adecuado puede utilizarse para generar cada uno de los k x m elementos de esta matriz aleatoria Beta. De esta manera, aproximadamente la mitad de los elementos de la matriz aleatoria Beta será 1 y aproximadamente la mitad de los registros elementos será 0.

35 El vector de columna  $V_Y$  se calcula por el codificador 11 como sigue:

$$\text{Beta} * V_A = V_Y,$$

40 en donde \* indica una multiplicación matricial.

En particular, el vector de paquetes codificados  $V_Y$  se calcula como sigue:

$$V_Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix} = \beta \times V_A = \begin{bmatrix} \beta_{11} & \beta_{12} & \dots & \beta_{1m} \\ \beta_{21} & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{k1} & \dots & \dots & \beta_{km} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

45 Ahora está completado el procedimiento de codificación en el ordenador 10 y cada fila de la matriz Beta, junto con el respectivo elemento de vector  $V_Y$  puede almacenarse en la memoria 12.

Una vez que se haya acabado el proceso de codificación, las k entradas codificadas o elementos del vector de columna codificado  $V_Y$  se transmiten a través del transmisor/receptor 13 a la etapa de transmisor/receptor 23 del teléfono móvil 20 y se decodifican allí por el decodificador 24. Cada elemento del vector  $V_Y$  junto con la correspondiente fila de la matriz aleatoria Beta se encapsula en un paquete de transmisión separado y cada paquete se transmite, por separado, a través del canal de comunicación 30. Una estructura de paquete, a modo de ejemplo, se ilustra en la Figura 2. Comprende un campo de metadatos para transmitir la respectiva fila de la matriz Beta y un campo de carga útil para incluir la respectiva entrada del vector  $V_Y$ . La etapa de la obtención de paquetes puede ejecutarse por el codificador 11 bajo control del microprocesador 16. El microprocesador 16 controla la memoria 12 para dirigir la fila de la matriz Beta respectiva y el correspondiente elemento vectorial al codificador 11. Algunos de estos paquetes pueden perderse durante la transmisión. Sin embargo, el decodificador de códigos lineales aleatorios 24 del teléfono móvil 20 puede recuperar el conjunto de datos completo  $S_A$  tan pronto como al menos se reciban m paquetes en el decodificador 24.

### 3. Sistema de decodificación de códigos lineales aleatorios

Es la tarea del decodificador 24 decodificar el vector de columna codificado  $V_Y$  para recuperar el conjunto de datos  $S_A$  de m elementos original, representado como el vector de columna  $V_A$  con m entradas. Sin embargo, el decodificador 24 requiere al menos m filas de la matriz Beta de codificación  $\beta$  que están contenidas en los paquetes recibidos. Dicho de otro modo, al menos m de los k paquetes transmitidos deben recibirse correctamente en el teléfono móvil 20.

En el decodificador 24, los elementos de  $S_A$  se calculan luego como sigue:

$$V_A = (\text{Beta})^{-1} * V_Y,$$

en donde  $(\text{Beta})^{-1}$  es la inversa de la matriz Beta utilizada por el codificador 11 y \* indica la multiplicación matricial. Conviene señalar que este proceso es isomórfico para resolver m ecuaciones lineales para m variables desconocidas, esto es, los m elementos del conjunto de datos  $S_A$ . El número de m ecuaciones lineales corresponde exactamente al número de elementos del conjunto de datos  $S_A$ . Por lo tanto, el proceso de decodificación se completará satisfactoriamente solamente si el número de filas de la matriz Beta es al menos m, esto es, el número de variables desconocidas. El número de filas se define como el número de filas linealmente independientes de la matriz Beta. Una de las ineficiencias del método de codificación lineal aleatorio, anteriormente descrito es que puesto que el ordenador 10 no tiene ninguna información anterior sobre la diferencia  $S_A - S_B$ , k ha de ser mayor que o igual al tamaño del conjunto de datos  $S_A$  para permitir una sincronización satisfactoria para el caso de que no existan elementos comunes entre los conjuntos de datos  $S_A$  y  $S_B$ . La diferencia  $S_A - S_B$  se define como el número de los elementos del conjunto de datos  $S_A$  que no están incluidos en el conjunto de datos  $S_B$ . Por lo tanto, el número de paquetes codificados transmitidos estará en el orden del tamaño m del conjunto de datos  $S_A$ , lo que hace a este método muy poco escalable cuando la diferencia  $S_A - S_B$  es mucho más pequeña que el tamaño m del conjunto de datos  $S_A$ .

Después de todo, si  $k \geq m$ , entonces el decodificador 24 puede reconstruir el vector de columna  $V_A$  como sigue:

1. En primer lugar, si  $k > m$ , entonces cualesquiera  $k - m$  filas de la matriz Beta y los correspondientes elementos codificados pueden rechazarse desde  $V_Y$  porque para poder resolver los m desconocidos  $(x_1, x_2, \dots, x_m)$ , solamente se necesitan m ecuaciones lineales. La matriz Beta resultante y  $V_Y$  se utiliza en la etapa siguiente.

2. Calcular la inversa  $\beta^{-1}$  de la matriz Beta. Conviene señalar que  $\beta^{-1}$  tiene la dimensión de m x k.

3. Entonces, el vector  $V_A$  puede reconstruirse como sigue.

$$V_A = \beta^{-1} \times V_Y = \begin{bmatrix} \beta'_{11} & \beta'_{12} & \dots & \beta'_{1k} \\ \beta'_{21} & \dots & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \beta'_{k1} & \dots & \dots & \beta'_{mk} \end{bmatrix} \times \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

Para un mejor entendimiento del proceso de decodificación, según se describió anteriormente, se proporciona un ejemplo numérico con m=3 y k=4.

El conjunto de datos a almacenado en la memoria 18 del ordenador 10 comprende los elementos 2, 3 y 4 y se define por  $S_A = (2, 3, 4)$  y por lo tanto

$$V_A = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}.$$

Una matriz Beta de k x m de dimensiones 4 x 3 se calcula por un generador de números aleatorios como sigue:

$$\beta = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

5

Entonces, el codificador 11 en el ordenador 10 puede calcular el vector  $V_Y$  como sigue:

$$V_Y = \beta \times V_A = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \\ 2 \\ 9 \end{bmatrix}$$

10

Se supone que las cuatro filas de la matriz Beta y los cuatro elementos {4, 3, 2,9} se almacenan en la memoria 12.

A continuación, cuatro paquetes codificados se crean en el ordenador 10 como sigue:

15 Paquete 1: [0 0 1 | | 4]

Paquete 2: [0 1 0 | | 3]

20 Paquete 3: [0 0 1 | | 2]

Paquete 4: [1 1 1 | | 9].

Estos paquetes pueden almacenarse también en la memoria 12.

25 A continuación, cada paquete 1 a 4 se transmite desde el ordenador 10 a través del canal de comunicación 30 al teléfono móvil 20. Estos paquetes pueden almacenarse en el depósito de paquetes 25 del teléfono móvil 20, que está conectado al decodificador 24. En el teléfono móvil 20, pueden encontrarse los casos siguientes:

30 i. Ningún paquete perdido durante la transmisión

Todos los 4 paquetes se reciben en el teléfono móvil 20.

En este caso, el teléfono móvil 20 puede rechazar arbitrariamente cualquiera de los paquetes recibidos y utilizar las tres filas restantes de la matriz Beta y el respectivo elemento del vector de columna  $V_Y$  transmitido en los respectivos paquetes para decodificar el vector  $V_A$ .

35

A modo de ejemplo, se supone que se rechaza el paquete 2 por el teléfono móvil 20. Entonces

$$\beta = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, V_Y = \begin{bmatrix} 4 \\ 2 \\ 9 \end{bmatrix}$$

40

Ahora, la inversa  $\beta^{-1}$  de la matriz Beta se calcula como

$$\beta^{-1} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & -1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

45 Por último, el decodificador 24 calcula  $V_A$  como sigue:

$$V_A = \begin{bmatrix} 0 & 1 & 0 \\ -1 & -1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 4 \\ 2 \\ 9 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}$$

Por lo tanto, el decodificador 24 ha sido operativamente satisfactorio en la decodificación de los elementos 2, 3 y 4 del conjunto de datos  $S_A$ .

5

ii. Un paquete perdido

En caso de que solamente se reciban tres paquetes en el teléfono móvil 20, se puede realizar un proceso de decodificación similar al descrito en la sección i). Sin embargo, ningún paquete puede rechazarse en el teléfono móvil 20 puesto que el contenido de al menos tres paquetes codificados se requiere para decodificar los tres elementos desconocidos del conjunto de datos  $S_A$  en el decodificador 24.

10

iii. Más de dos paquetes perdidos

15 En el caso de que menos de tres paquetes lleguen al teléfono móvil 20, es matemáticamente imposible decodificar  $V_A$  en el decodificador 24 porque el número de tres elementos desconocidos de  $V_A$  supera el máximo valor posible de la matriz Beta.

20 La complejidad informática del proceso de decodificación en el decodificador 24 es principalmente debida al método Gaussiano para encontrar la inversa de la matriz Beta recibida en el teléfono móvil 20. Esta circunstancia se ha demostrado que requiere ( $k^3$ ) operaciones matemáticas, haciendo inviable poner en práctica este método para un parámetro muy grande k. Por lo tanto, los grandes conjuntos de datos no se pueden sincronizar prácticamente utilizando el método de la codificación lineal anteriormente descrito. Cómo resolver este problema se describe con detalle a continuación.

25

Ahora, la función y operación del sistema informático distribuido 1 se ilustra en detalle.

El sistema informático distribuido 1 está adaptado para reconstruir eficientemente el conjunto de datos  $S_A$  del ordenador 10 en el teléfono móvil 20 sacando partido operativo de una comunicación bidireccional.

30

El sistema 1 utiliza la técnica de corrección de errores, que se emplea para la corrección de errores de datos, también para la finalidad de la sincronización de datos. La técnica de corrección de errores utilizada está basada en la técnica de códigos lineales aleatorios según fue anteriormente descrito. Sin embargo, varias modificaciones y características adicionales se han introducido, que se presentan a continuación a modo de ejemplo.

35

4. Sincronización de datos básica

Se supone que un conjunto de datos  $S_B$  que tiene un conjunto de n elementos de datos  $\{x_1, \dots, x_n\}$  se memoriza en la memoria 28, que el conjunto de datos  $S_A$  que tiene un conjunto de m elementos de datos  $\{x_1, \dots, x_m\}$  se memoriza en la memoria 18 y que el parámetro k se establece a un valor predeterminado. Ninguno, algunos o todos los elementos del conjunto de datos  $S_B$  pueden ser idénticos a los del conjunto de datos  $S_A$ . En caso de que algunos elementos sean idénticos entre los conjuntos de datos  $S_A$  y  $S_B$ , el decodificador 24 se adapta para detectar los elementos idénticos y solamente realiza la sincronización de datos en los elementos de datos diferentes. Esto da lugar a una comunicación reducida desde el ordenador 10 al teléfono móvil 20. En particular, dichos elementos del conjunto de datos  $S_A$  que son idénticos a los elementos en el conjunto de datos  $S_B$  no necesitan decodificarse por el decodificador 24.

40

45

En primer lugar, el teléfono móvil 20 como el dispositivo de decodificación comprueba si su propio conjunto de datos  $S_B$  tiene menos o un número igual a k elementos, en donde k es el parámetro ajustable.

50

Si éste fuera el caso, entonces el teléfono móvil 20 puede simplemente enviar una demanda a través del canal de comunicación 30 al ordenador 10 para transmitir todos los elementos del conjunto de datos  $S_A$  puesto que esto último es más eficiente que codificar los elementos del conjunto de datos  $S_A$  en k paquetes, transmitir estos k paquetes codificados al teléfono móvil 20 y decodificar los paquetes recibidos por el decodificador 24. Después de recibir el conjunto de datos  $S_A$  desde el ordenador 10, el teléfono móvil 20 simplemente necesita determinar la diferencia  $S_A - S_B$  para completar el procedimiento de sincronización de datos.

55

Por otro lado, si el teléfono móvil 20 determina que el conjunto de datos  $S_B$  contiene más de k elementos, envía un respectivo mensaje al ordenador 10. En respuesta a dicho mensaje, el ordenador 10 transmite un mensaje al teléfono móvil 20. El mensaje incluye una información sobre todos los m elementos del conjunto de datos  $S_A$ . Con el fin de reducir la cantidad de datos a transmitirse, el mensaje puede incluir las cadenas aleatorias, denominadas *hashes* de cada elemento del conjunto de datos  $S_A$ . Las cadenas aleatorias (*hashes*) se calculan utilizando cualquier

60

función de *hash* estándar  $h$ . Una función de *hash* es cualquier procedimiento bien definido o función matemática que convierte un elemento de datos del conjunto de datos en un número entero único, en donde la magnitud de la función *hash* es mucho más pequeña que la magnitud del elemento de datos. Por lo tanto, las *hashes* de cada elemento se representan por la función siguiente:

5

$$\text{Hash de } x_1 = h(x_1)$$

$$\text{Hash de } x_2 = h(x_2)$$

10

$$\text{Hash de } x_m = h(x_m)$$

Todas las funciones *hashes*  $[h(x_1), h(x_2), \dots, h(x_m)]$  pueden enviarse en un paquete de claves aleatorias *hashes* único desde el ordenador 10 al teléfono móvil 20.

15 El teléfono móvil 20 calcula también las claves aleatorias *hashes* de todos los elementos del conjunto de datos  $S_B$  y compara estas *hashes* con las *hashes* recibidas en el paquete de funciones *hash*. El teléfono móvil 20 está adaptado para determinar los elementos comunes entre los conjuntos de datos  $S_A$  y  $S_B$ . Estos elementos comunes no necesitan decodificarse por el decodificador 24, puesto que están ya disponibles en el conjunto de datos  $S_B$ . En el caso de que cada *hash* transmitida en el paquete de funciones *hash* corresponda a una *hash* de un elemento del conjunto de datos  $S_B$ , todos los  $m$  elementos del conjunto de datos  $S_A$  están presentes en el conjunto de datos  $S_B$ . En respuesta a este resultado, el teléfono móvil 20 envía una respuesta al ordenador 10 a través del canal de comunicación 30 indicando que se ha completado la sincronización de datos y que no se requiere ninguna otra acción. Si, sin embargo, el teléfono móvil 20 determina que faltan algunas *hashes*, entonces requiere al ordenador 10 el envío de los paquetes codificados.

25

Este procedimiento se explica en relación con el procedimiento de reducción del rango matricial.

#### 5. Reducción de la complejidad informática utilizando la reducción de rango matricial

30 Se supone que el conjunto de datos  $S_B$  comprende los elementos 3 y 5 y el conjunto de datos  $S_A$  incluye los elementos 2, 3 y 4. Además, el codificador 11 ha codificado ya el conjunto de datos  $S_A$  en los cuatro paquetes: paquete 1  $[0\ 0\ 1\ ||\ 4]$ , paquete 2  $[0\ 1\ 0\ ||\ 3]$ , paquete 3  $[0\ 0\ 1\ ||\ 2]$  y paquete 4  $[1\ 1\ 1\ ||\ 9]$  según se describe en la sección 3. Estos paquetes se almacenan en el depósito de paquetes 13 del ordenador 10. Además, el teléfono móvil 20 ha recibido ya un paquete de funciones *hash* que contiene las *hashes*  $h(2)$ ,  $h(3)$  y  $h(4)$ . Ahora, calculando la clave aleatoria *hash* de sus propios elementos establecidos  $h(3)$  y  $h(5)$  y comparando estas *hashes* con las *hashes* del paquete de funciones *hash*, el teléfono móvil 20 puede conocer que el elemento establecido '3' es común para los conjuntos de datos  $S_A$  y  $S_B$  y que los elementos (2, 4) faltan en el conjunto de datos  $S_B$ .

35

40 Se supone, además, que el teléfono móvil 20 ha demandado al ordenador 1 el envío de los paquetes codificados y que tres paquetes codificados, a modo de ejemplo, los paquetes 1, 3 y 4, han sido recibidos por el teléfono móvil 20 puesto que el paquete 2 se ha perdido durante la transmisión (véase sección 4, ii). Sin embargo, solamente dos paquetes son suficientes para decodificar los dos elementos que faltan. El teléfono móvil 20 puede obtener la matriz Beta y el vector de columna  $V_Y$  de los tres paquetes como sigue:

40

$$\beta = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, V_Y = \begin{bmatrix} 4 \\ 2 \\ 9 \end{bmatrix}$$

45

Puesto que el elemento '3' correspondiente a la segunda columna de la matriz Beta, generada en el ordenador 10 para la etapa de codificación, está ya disponible en el conjunto de datos  $S_B$  en el teléfono móvil 20, puede sustraerse del vector de columna  $V_Y$  y puede suprimirse la segunda columna de la matriz Beta. Como resultado, el nuevo vector  $V_Y'$  y la nueva matriz Beta  $\beta'$  son como sigue:

50

$$V_Y' = \begin{bmatrix} 4 - 0 \times 3 \\ 2 - 0 \times 3 \\ 9 - 1 \times 3 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \\ 6 \end{bmatrix}$$

y Beta es

55

$$\beta' = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

5 Denominamos este procedimiento como “reducción de rango” en el ámbito matricial. Después de la reducción del rango, sin embargo, permanecen solamente 2 desconocidos para determinar y tres filas en la matriz Beta. Por lo tanto, según se indicó con anterioridad, cualquiera de las tres filas puede borrarse para obtener un vector reducido  $V_Y''$  y una matriz Beta reducida  $\beta''$

$$\beta'' = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \text{ y } V_Y'' = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

10 Como resultado, solamente la inversa de la matriz Beta reducida  $\beta''$  se calcula en el decodificador 24 como sigue:

$$(\beta'')^{-1} = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}$$

15 Por último, el decodificador 24 realiza la última etapa de decodificación, esto es, la multiplicación matricial de

$$V_{A}'' = (\beta'')^{-1} \times V_Y'' = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \times \begin{bmatrix} 4 \\ 6 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

para obtener exactamente los elementos 2 y 4 del conjunto de datos  $S_A$ , que faltan en el conjunto de datos  $S_B$ .

20 Con la reducción del rango, esto es, la estrategia de decodificar solamente los elementos que faltan en el conjunto de datos  $S_B$ , se puede reducir la complejidad de cálculo informático en el decodificador 24.

### 6. Reducción de la complejidad de cálculo informático utilizando tres estructuras de datos

25 La cuestión de la reducción de la comunicación y complejidad de cálculo informático se resuelve, a continuación, utilizando una estructura de datos en arborescencia en el ordenador 10 y el teléfono móvil 20 para poner en práctica una técnica de dividir y resolver. Esta técnica permite utilizar grandes conjuntos de datos y la técnica de codificación lineal aleatoria.

#### 30 Estructura de datos en arborescencia

Ahora, se considera un ejemplo adicional. Se supone que el conjunto de datos  $S_A$ , memorizado en el ordenador 10, comprende  $m=6$  elementos, esto es, 0, 1, 6, 10, 14 y 15, mientras que el conjunto de datos  $S_B$  memorizado en el teléfono móvil 20 comprende  $n=3$  elementos, esto es, 0, 1 y 9. Los pequeños tamaños del conjunto de datos son solamente elegidos para facilitar la ilustración.

35 En general, el ordenador 10 y el teléfono móvil 20 pueden crear una estructura de datos en arborescencia a partir de los elementos del conjunto de datos  $S_A$  y del conjunto de datos  $S_B$ , respectivamente. Un ejemplo de una estructura de datos en arborescencia se ilustra en la Figura 3, que representa la estructura de datos en arborescencia utilizada por el ordenador 10 así como la estructura de datos en arborescencia utilizada por el teléfono móvil 20. En la forma de realización preferida, la estructura de datos en arborescencia del conjunto de datos  $S_A$  se memoriza en la memoria 17, en donde la estructura de datos en arborescencia del conjunto de datos  $S_B$  se memoriza en la memoria 27. Una arborescencia es una estructura de datos jerárquica multinivel constituida por nodos, según se ilustra en la Figura 3. El nodo del nivel superior se denomina un nodo raíz, mientras que los nodos en niveles más bajos se denominan nodos hijo. Los nodos pertenecientes al mismo nivel se establecen para el mismo tamaño de campo predeterminado. Cada nodo contiene los elementos establecidos que se adaptan entre los tamaños de campo mínimo y máximo del nodo respectivo. El nodo raíz contiene el conjunto de datos íntegro del respectivo dispositivo de comunicación puesto que abarca el campo que puede alojar el elemento valorado mínimo y máximo en el conjunto de datos. El nodo raíz contiene también enlaces a todos sus nodos hijos en la arborescencia. Un ejemplo de sincronización de datos, que utiliza una estructura de datos en arborescencia, se explica a continuación.

División en más pequeños subconjuntos de datos

La división del método de sincronización de datos se realiza dividiendo ambos conjuntos de datos  $S_A$  y conjuntos de datos  $S_B$  en subconjuntos de datos disjuntos según se ilustra en la Figura 3. La estructura en arborescencia de lado izquierdo está situada en el ordenador 10, mientras que la estructura en arborescencia del lado derecho está memorizada en el teléfono móvil 20. Conviene señalar que un subconjunto de datos puede considerarse como el contenido del respectivo nodo hijo, mientras que el propio conjunto de datos puede considerarse como el contenido de un nodo raíz. En esta forma de realización, el tamaño del campo del nodo raíz se establece en  $2^4 = 16$ . Todos los elementos de ambos conjuntos de datos  $S_A = \{0, 1, 6, 10, 14, 15\}$  y  $S_B = \{1, 9, 15\}$  residen entre 0 y 15. Además, se supone que un ordenamiento de los elementos del conjunto de datos es posible. En caso de datos numéricos, este ordenamiento puede basarse en el valor, mientras que en el caso de elementos de datos de cadenas o textuales, este ordenamiento puede ser sobre una base lexicográfica. En la forma de realización ilustrada en la Figura 3, se utiliza el ordenamiento numérico para distribuir los elementos del conjunto de datos para los nodos hijo.

Definimos que el nodo raíz radica en el nivel 0. Cada nodo raíz 40 y 50, que tiene un tamaño de campo de 16 elementos del conjunto de datos posibles 0,1, 2, ..., 15 se dividen entonces en dos nodos hijo de nivel -1 denominados 41, 42 y 51, 52, respectivamente, en donde cada nodo hijo de nivel -1 tiene un tamaño de campo de 8 elementos de datos posibles. Comenzando desde la izquierda a la derecha, el nodo hijo de nivel -1 41 puede incluir los posibles elementos de datos 0 a 7, en donde el nodo hijo de nivel -1 42 puede incluir los posibles elementos de datos 8 a 15. En este caso, el nodo hijo 41 memoriza los elementos de datos  $\{0, 1, 6\}$  del conjunto de datos  $S_A$  y el nodo hijo 42 memoriza los elementos  $\{10, 14, 15\}$  del conjunto de datos  $S_A$ . En un nodo hijo de nivel -1 51 de forma similar, comprende los elementos de datos  $\{0, 1\}$  y el nodo hijo de nivel -1 52 comprende el elemento de  $\{9\}$  del conjunto de datos  $S_B$ . A continuación, cada nodo hijo de nivel -1 se divide, a su vez, en dos nodos hijos de nivel -2 del nivel 2, en donde cada nodo hijo de nivel -2 tiene un tamaño de campo de 4 elementos. Por lo tanto, todos los 16 elementos posibles del conjunto de datos pueden distribuirse a los respectivos nodos hijo según se ilustra en la Figura 3, comenzando de nuevo en el lado izquierdo de la estructura en arborescencia. En particular, el nodo hijo 41 de nivel -1 se divide en los nodos hijos 43, 44 de nivel -2, el nodo hijo 42 de nivel -1 se divide en los nodos hijo 45, 46 de nivel -2 el nodo hijo 51 de nivel -1 se divide en los nodos hijo 53, 54 de nivel -2 y el nodo hijo 52 de nivel -1 se divide en los nodos hijo 55, 56 de nivel -2. Los elementos de los respectivos subconjuntos de datos de nivel -2 del conjunto de datos  $S_A$  se distribuyen en los nodos hijo 43 a 46 del nivel -2, según se ilustra en la Figura 3. Los elementos de los respectivos subconjuntos de datos de nivel -2 del conjunto de datos  $S_B$  se distribuyen en los nodos hijo 53 a 56 del nivel -2, según se ilustra en la Figura 3, en donde los nodos hijo 54 y 56 representan subconjuntos de datos vacíos. Como puede deducirse, los resultados de divisiones repetidas en menos elementos en cada nodo hijo, está en conformidad con el número creciente de niveles. Conviene señalar que cada nodo desde 40 a 46 de la estructura de datos en arborescencia en el ordenador 10 está lógicamente enlazado con un nodo correspondiente de la estructura de datos en arborescencia en el teléfono móvil 20, en donde los nodos enlazados radican al mismo nivel. En particular, el nodo raíz 40 está asociado con el nodo raíz 50, el nodo hijo 41 está asociado con el nodo hijo 51, el nodo hijo 42 está asociado con el nodo hijo 52 y así sucesivamente, según se ilustra en la Figura 3 por las líneas de trazos. Las estructuras de datos en arborescencia, ilustradas en la Figura 3, pueden memorizarse en la memoria 17 y 27, respectivamente, en donde cada nodo puede representar el respectivo espacio de posición de memoria. Como resultado, los elementos de cada subconjunto de datos pertenecientes al conjunto de datos  $S_A$  pueden memorizarse en posiciones direccionales de la memoria 17, mientras que los elementos de cada subconjunto de datos pertenecientes al conjunto de datos  $S_B$  pueden memorizarse en posiciones direccionales de la memoria 27.

Conviene señalar que las estructuras de datos en arborescencia se crean de modo que los elementos idénticos del conjunto de datos  $S_A$  y del conjunto de datos  $S_B$  radiquen en los mismos nodos de la respectiva estructura de datos en arborescencia. El procedimiento de división se puede repetir hasta que el tamaño del campo de los nodos hijo correspondientes sea suficientemente pequeño, a modo de ejemplo, menor que o igual al parámetro k. El número de divisiones del tamaño del campo se denomina el tamaño de partición p de la arborescencia y corresponde al número máximo de niveles creados. En la forma de realización ilustrada en la Figura 3, el tamaño de partición p es 2. El procedimiento de división, según se describió anteriormente, puede denominarse una partición establecida.

Con la realización de una partición establecida según se describió anteriormente, el proceso de codificación y de decodificación se hace mucho menos costoso, desde el punto de vista del cálculo informático, puesto que los correspondientes subconjuntos de datos de los conjuntos de datos  $S_A$  y  $S_B$  se procesan posteriormente en lugar de los conjunto de datos completos. Por lo tanto, se puede utilizar una matriz Beta más pequeña para codificar y decodificar los respectivos subconjuntos de datos. Aunque aumenta el número de nodos, las ventajas debidas a la resolución de varios pequeños problemas de sincronización y la posterior fusión de los resultados superan las ventajas de una sincronización grande única. Los correspondientes subconjuntos en el ordenador 10 y en el teléfono móvil 20 se sincronizan utilizando el procedimiento de sincronización básica anteriormente descrita. Los resultados de varias pequeñas sincronizaciones se combinan con el fin de obtener la sincronización completa de los conjuntos de datos  $S_A$  y  $S_B$ .

Procedimiento de codificación aplicado a una estructura de datos en arborescencia

El objetivo del procedimiento de codificación puesto en práctica en el codificador 11 es crear  $k$  paquetes codificados para cada nodo de la arborescencia que puedan transmitirse al decodificador 24 para su posterior decodificación. El valor de  $k$  es un parámetro ajustable del sistema. Un alto valor de  $k$  aumenta el tamaño de la matriz Beta para la inversión utilizando el proceso de inversión matricial y por lo tanto, esto incrementa la complejidad del cálculo informático. Por el contrario, un valor muy bajo de  $k$  aumentará el número de mensajes de realimentación entre el ordenador 10 y el teléfono móvil 20, puesto que el teléfono móvil 20 necesita sincronizar más nodos de la arborescencia antes de que se concluya la sincronización completa. Por lo tanto,  $k$  debe ser relativamente grande, p.e., 64, si el dispositivo de decodificación es capaz desde el punto de vista informático, p.e., un ordenador personal y debe ser más pequeño si el dispositivo de decodificación es menos capaz, como, a modo de ejemplo, el teléfono móvil 20.

El procedimiento de codificación básico, según se explica en la sección 2, se ejecuta por el codificador 11 para cada nodo 40 a 46 de la estructura de datos en arborescencia memorizada en la memoria 17 y los  $k$  paquetes codificados resultantes, para cada subconjunto de datos, se memorizan, a modo de ejemplo, en posiciones direccionables de la memoria 12 para su posterior sincronización con el teléfono móvil 20. El procedimiento de codificación para cada nodo, esto es, cada subconjunto de datos, es preferentemente ejecutado a priori para la sincronización de datos con el fin de acelerar el proceso de sincronización reduciendo el tiempo de espera de los cálculos descritos a continuación. Puesto que los mismos paquetes codificados, memorizados en la memoria 12, pueden reutilizarse en múltiples sincronizaciones con múltiples dispositivos de decodificación, este coste de codificación se amortiza a través de todos los procedimientos de sincronización.

En particular, una matriz Beta de  $k \times l_i$  se crea para cada nodo hijo 41 a 46 por un generador de números aleatorios del ordenador 10, en donde  $k$  es 2,  $l_i$  es el número de elementos del nodo hijo  $i$  e  $i$  varía de 41 a 46. En particular,  $l_{41}$  es 3,  $l_{42}$  es 3 y así sucesivamente. En una forma similar, se crea una matriz Beta de  $k \times m$  para el nodo raíz 40, en donde  $m$  es 6. A continuación, los elementos de cada nodo, que se representan por un vector de columna  $V_A$ , se multiplican en el codificador 11, con la respectiva matriz Beta para obtener un vector de columna de  $k$  elementos  $V_Y$  para cada nodo. Las filas de cada matriz Beta y los elementos de cada vector  $V_Y$  se memorizan en posiciones de memoria 12 direccionables, de modo que se puedan encapsular en paquetes de una estructura según se ilustra en la Figura 2.

Procedimiento de decodificación aplicado a una estructura de datos en arborescencia

El procedimiento de decodificación en el decodificador 24 del teléfono móvil 20 utiliza la estructura de datos en arborescencia según se ilustra en el lado derecho de la Figura 3. El proceso de decodificación sirve para recuperar los elementos del conjunto de datos  $S_A$  que no están incluidos en el conjunto de datos  $S_B$ . El decodificador 24 comienza con el nodo raíz 50 y ejecuta el procedimiento de decodificación según se describe en la sección 3. Conviene señalar que el mismo procedimiento de decodificación es aplicable, si fuera necesario, a cada nodo hijo en la estructura de datos en arborescencia.

La decisión de si continuar decodificando un nodo hijo, o no, se determina en la manera siguiente. Si el rango de la matriz  $B$ , que se define por el parámetro  $k$ , es menor que el número de elementos diferentes entre dos nodos correspondientes, entonces el nodo hijo situado más a la izquierda del siguiente nivel más bajo se activa y el procedimiento de decodificación, según se describe en la sección 3, se repite para cada nodo hijo de izquierda a derecha. Por lo tanto, siguiendo ese procedimiento recursivo, el problema de sincronización de datos se subdivide en más pequeñas tareas de sincronización.

Para ilustrar este procedimiento de sincronización de datos recursivo, consideramos el ejemplo siguiente:

Volvemos a la Figura 3 y suponemos que el número  $m$  de elementos del conjunto de datos  $S_A$  es 6, el número  $n$  de elementos del conjunto de datos  $S_B$  es 3, el tamaño de la partición  $p$  es 2, el parámetro  $k$  es 2 y el tamaño máximo del nodo raíz es 16 ( $b=4$ ).

Por lo tanto, con respecto al nivel 0 de la arborescencia que se ilustra en la Figura 3, el campo mínimo, máximo es 0,15 respectivamente, en donde los elementos de campos mínimo y máximo están marcados en la parte superior izquierda y en la parte superior derecha de cada nodo de la arborescencia.

En primer lugar, se puede ejecutar el procedimiento descrito en la sección 5.

De este modo, el decodificador 24 comprueba si el nodo raíz 50, esto es, el conjunto de datos  $S_B$ , tiene menos o igual a  $k=2$  elementos. Si éste es el caso, entonces el teléfono móvil 20 simplemente envía una demanda a través del canal de comunicación 30 al ordenador 10 para enviar todos los elementos del nodo raíz 40, que corresponden a los elementos del conjunto de datos  $S_A$ . Después de recibir el conjunto de datos  $S_A$  desde el ordenador 10, el teléfono móvil 20 determina la diferencia  $S_A - S_B$ .

En el presente ejemplo, el decodificador 24 determina que el nodo raíz 50 contiene más de  $k=2$  elementos. Por lo tanto, el teléfono móvil 20 demanda al ordenador 10 el envío del paquete de funciones *hash* que contiene las *hashes*

para todos los seis elementos, es decir  $h(0)$ ,  $h(1)$ ,  $h(6)$ ,  $h(10)$ ,  $h(14)$ ,  $h(16)$ . Puesto que el teléfono móvil 20 conoce los *hashes* de todos los elementos de su conjunto de datos  $S_B$ , es decir  $h(0)$ ,  $h(1)$  y  $h(9)$ , determina que el tamaño de la diferencia  $S_A - S_B$  es 4, esto es, el número total de elementos no incluidos en el conjunto de datos  $S_B$  es 4. Por lo tanto, el decodificador 24 no puede decodificar los 4 elementos puesto que el codificador 11 del ordenador 10 solamente envía 2 paquetes codificados.

Como resultado, el teléfono móvil 20 solicita el nivel 1 y el nodo hijo 51 que contiene los elementos  $\{0, 1\}$  de conjunto de datos  $S_B$ . Además, el teléfono móvil 20 informa al ordenador 10 que utiliza ahora el nodo hijo 51. En respuesta a esa información, el ordenador 10 conoce que ha de utilizar el nodo hijo correspondiente 41. Por lo tanto, el ordenador 10 envía las *hashes* de todos los elementos incluidos en el nodo hijo 41, esto es, las funciones *hashes*  $h(0)$ ,  $h(1)$  y  $h(6)$ . Conviene señalar que los elementos  $\{0, 1$  y  $6\}$  corresponden a los elementos del respectivo subconjunto de datos. Puesto que el teléfono móvil 20 conoce las *hashes* de los elementos  $\{0, 1\}$  incluidos en el nodo hijo 51, determina que la diferencia es solamente de 1 elemento. Ahora, el teléfono móvil 20 demanda al ordenador 10 el envío de los  $k=2$  paquetes codificados pertenecientes al nodo hijo 41. Por lo tanto, si uno de ambos paquetes codificados se recibe satisfactoriamente por el teléfono móvil 20, el elemento establecido ausente  $\{6\}$  puede decodificarse por el decodificador 24 según se describe en la sección 3. Ahora, el teléfono móvil 20 reduce el número total de diferencias a 3, lo que indica que todavía faltan 3 elementos en el conjunto de datos  $S_B$ . Puesto que el número de elementos diferentes es todavía mayor que  $k=2$ , el teléfono móvil 20 solicita el nodo hijo 52 de nivel -1 siguiente que contiene el elemento  $\{9\}$ . El teléfono móvil 20 demanda al ordenador 10 el uso del nodo hijo correspondiente 42. En consecuencia, el ordenador 10 envía las claves aleatorias *hashes* de los elementos  $\{10, 14, 15\}$ , que están contenidos en el nodo hijo 42, al teléfono móvil 20. En respuesta a las *hashes* recibidas y a la *hash* del elemento  $\{9\}$  del nodo hijo 52, el teléfono móvil 20 determina que el número de diferencias es todavía 3 y en consecuencia, mayor que 2. Por lo tanto, el decodificador 24 es incapaz de decodificar los 3 elementos ausentes. A continuación, el teléfono móvil 20 salta desde el nodo hijo 52 al nivel 2 y elige el nodo hijo 55 del nivel -2 más a la izquierda que depende del nodo hijo 52. El nodo hijo 55 contiene el elemento  $\{9\}$ . Ahora, el teléfono móvil 20 demanda al ordenador 10 la selección del nodo hijo correspondiente 45 que contiene el elemento  $\{10\}$ . El ordenador 10 transmite la *hash* del elemento 10 al teléfono móvil 20 que reconoce que la diferencia es solamente de 1 elemento entre el nodo hijo 45 y el nodo hijo 55, por lo tanto, el teléfono móvil 20 demanda al ordenador 10 que envíe los dos paquetes codificados que representan al nodo hijo 45. Si al menos un paquete codificado se recibe por el teléfono móvil 20, entonces, el decodificador 24 puede decodificar el elemento ausente  $\{10\}$  según se describió anteriormente y la sincronización de datos entre el nodo hijo 45 y el nodo hijo 55 es operativamente satisfactoria. De un modo similar, el teléfono móvil 20 solicita, por último, al nodo hijo 56 y detecta que la diferencia entre los elementos del nodo hijo 56 y los elementos del nodo hijo correspondiente 46 en el ordenador 10 es 2. Por lo tanto, el teléfono móvil 20 demanda al ordenador 10 el envío de los dos paquetes codificados memorizados en la memoria 12 con respecto al nodo hijo 46. Si ambos paquetes codificados se reciben, el decodificador 24 puede decodificar los últimos elementos ausentes  $\{14, 15\}$  utilizando el procedimiento de decodificación anteriormente descrito. Por último, el decodificador 24 combina los resultados de todos los procedimientos de sincronización de datos ejecutados posteriormente, lo que proporciona la diferencia completa  $S_A - S_B$ . Dependiendo de una aplicación puesta en práctica, los elementos ausentes  $\{6, 10, 14$  y  $15\}$  decodificados por el decodificador 24 pueden añadirse al conjunto de datos  $S_B$  y puede suprimirse el elemento  $\{9\}$ . Como resultado, el conjunto de datos  $S_A$  ha sido completamente reconstruido en el teléfono móvil 20.

Conviene señalar que el procedimiento de reducción del rango según se describe en la sección 5 puede ejecutarse por el decodificador 24 durante cada sincronización de subconjunto de datos.

## 7. Datos de rendimiento

En esta sección, se proporcionan algunos rendimientos con respecto a una puesta en práctica ejemplar del sistema 1. El sistema 1 es eficiente en la comunicación puesto que el número de paquetes transmitidos desde un dispositivo de comunicación, tal como el ordenador 10, a un dispositivo de decodificación, tal como un teléfono móvil 20, es independiente de la magnitud de los conjuntos de datos y solamente depende de las diferencias entre los dos conjuntos de datos  $S_A$  y  $S_B$ . Este comportamiento se ilustra en la Figura 4 en donde el número de diferencias entre los dos conjuntos de datos  $S_A$  y  $S_B$  se mantienen constantes en 500 elementos diferentes y el número de elementos del conjunto de datos se aumenta desde 0 a 20000 (eje X). En esta ejecución de prueba, los elementos establecidos se eligen aleatoriamente desde un campo de 32 bits,  $k$  se establece en 32 y el número de particiones se establece en 4. El número total de paquetes codificados (eje Y) transmitidos desde el dispositivo de codificación al dispositivo de decodificación permanece aproximadamente constante incluso con una magnitud creciente establecida del conjunto. Esto indica la escalabilidad inherente del sistema en comparación con la copia del conjunto de datos completo desde el dispositivo de codificación al dispositivo de decodificación cuando los conjuntos de datos se hacen grandes.

En el siguiente experimento, la magnitud del conjunto de datos se fija en 100000 elementos y el número de diferencias entre los conjuntos de datos  $S_A$  y  $S_B$  (eje X) se incrementa. La Figura 5 ilustra el número de transmisiones (eje Y) para el número creciente de diferencias. Los valores de  $k$  utilizados en este experimento fueron 8, 32 y 64 y el número de particiones fue establecido en 4. La Figura 5 ilustra que el número de transmisiones de paquetes crece linealmente con el número de diferencias entre los conjuntos de datos.

5 El método y sistema descritos dan lugar a una sincronización de datos, eficiente y escalable, entre dos dispositivos de comunicación 10 y 20. El ordenador 10 codifica su conjunto de datos y envía estos datos codificados al teléfono móvil 20. El teléfono móvil 20 decodifica, entonces, esta información con el fin de extraer los elementos del conjunto de datos del ordenador 10 que están ausentes a nivel local. Con el fin de evitar una gran complejidad de cálculo informático cuando se utilizan códigos lineales aleatorios, se utiliza un proceso de aceleración nuevo para reducir el número de cálculos mediante la realimentación de comunicación inteligente limitada desde el teléfono móvil 20 al ordenador 10 durante el proceso de sincronización de datos. En comparación con los códigos lineales aleatorios bien conocidos ordinarios, el uso de los resultados de la realimentación inteligente da lugar a la reducción de la complejidad de decodificación informática en el lado de la decodificación, permitiendo, de este modo, la utilización del código lineal aleatorio para conjuntos de datos mucho mayores y/o permitiendo el despliegue del sistema en dispositivos de potencia informática baja tales como microsensores y teléfonos móviles.

15 El sistema se adapta dinámicamente a diferentes escenarios operativos de sincronización según se requiera por la aplicación. Además, puede sincronizar satisfactoriamente datos en escenarios operativos en donde existan ninguno, unos pocos, muchos o todos los elementos de datos idénticos entre los conjuntos de datos que se sincronizan.

20 En la forma de realización preferida, según se describió anteriormente, el conjunto de datos  $S_B$  puede ser eficiente reconstruido en el teléfono móvil sacando partido operativo de un enlace de comunicación bidireccional 30. El sistema es operativamente satisfactorio en la limitación de la comunicación hacia dentro de un múltiplo constante del número de diferencias entre los conjuntos de datos  $S_A$  y  $S_B$ . Esto significa que, prescindiendo del número de elementos en el conjunto de datos  $S_A$ , el número de bytes transferidos entre el ordenador 10 y el teléfono móvil 20 es proporcional al número de diferencias entre los conjuntos de datos  $S_A$  y  $S_B$  y no proporcional al número total de elementos establecidos en los conjuntos de datos. Además, el sistema 1 se adapta a los errores en la transmisión de datos utilizando técnicas de corrección de errores entre los dispositivos de comunicación durante la sincronización de datos y de este modo, reduce las posibilidades de fallos de sincronización cuando el conjunto de datos  $S_A$  no es idéntico al conjunto de datos  $S_B$  al final del procedimiento de sincronización.

30

**REIVINDICACIONES**

1. Un método para sincronizar conjuntos de datos memorizados en diferentes dispositivos de comunicación (10, 20) que están conectados por intermedio de enlaces de comunicación, que comprende las etapas de:
- 5
- 1a) Memorización de un primer conjunto de datos en un primer dispositivo de comunicación (10), comprendiendo dicho primer conjunto de datos un número de  $m$  de elementos;
- 10
- 1b) Memorización de un segundo conjunto de datos en un segundo dispositivo de comunicación (20), comprendiendo dicho segundo conjunto de datos un número de  $n$  de elementos;
- 15
- 1c) Codificación, en el primer dispositivo de comunicación (10), del primer conjunto de datos utilizando un código lineal aleatorio para obtener un número de  $k$  de elementos codificados, en donde  $k$  es un parámetro ajustable;
- 1d) Determinación, en el segundo dispositivo de comunicación (20), del número total de elementos, que son diferentes entre dicho primer conjunto de datos y dicho segundo conjunto de datos, sobre la base de un mensaje recibido desde el primer dispositivo de comunicación (10);
- 20
- 1e) Determinación, en el segundo dispositivo de comunicación (20), de si el número total de diferentes elementos es menor que o igual al parámetro  $k$ ;
- 1f) Si la respuesta es afirmativa, entonces transmitir los  $k$  elementos codificados desde el primer dispositivo de comunicación (10) al segundo dispositivo de comunicación (20);
- 25
- 1g) Decodificación, en el segundo dispositivo de comunicación (20), de al menos algunos de los elementos codificados recibidos utilizando un decodificador lineal aleatorio con el fin de obtener todos los elementos del primer conjunto de datos, que no estén contenidos en el segundo conjunto de datos;
- 30
- 2a) Realización, de una partición de conjunto en el primer conjunto de datos con el fin de obtener una arborescencia multinivel jerárquica que comprende una pluralidad de primeros subconjuntos de datos;
- 2b) Realización de una partición de conjunto en el segundo conjunto de datos con el fin de obtener una arborescencia multinivel jerárquica que comprende una pluralidad de segundos subconjuntos de datos, en donde cada uno de los primeros subconjuntos de datos corresponden a uno separado de los segundos subconjuntos de datos;
- 35
- 2c) Codificación, en el primer dispositivo de comunicación (10), de cada primer subconjunto de datos utilizando un código lineal aleatorio con el fin de obtener un número de  $k$  elementos codificados;
- 40
- 2d) Si el número de diferentes elementos determinados en la etapa 1d) es mayor que  $k$ , entonces seleccionar, en el segundo dispositivo de comunicación (20) un segundo subconjunto de datos del segundo conjunto de datos y seleccionar, en el primer dispositivo de comunicación (10), el primer subconjunto de datos correspondiente en respuesta a un mensaje recibido desde el segundo dispositivo de comunicación;
- 45
- 2e) Determinación, en el segundo dispositivo de comunicación (20), del número de elementos, que son diferentes entre el segundo subconjunto de datos seleccionado y el primer subconjunto de datos correspondiente sobre la base de un mensaje recibido desde el primer dispositivo de comunicación (10);
- 50
- 2f) Determinación, en el segundo dispositivo de comunicación (20), de si el número de diferentes elementos seleccionados entre el segundo subconjunto de datos seleccionado y el primer subconjunto de datos seleccionado es menor que o igual al parámetro  $k$ ;
- 2g) Si la respuesta es afirmativa, entonces transmitir los  $k$  elementos codificados del primer subconjunto de datos seleccionado entre el segundo subconjunto de datos seleccionado y el primer subconjunto de datos seleccionado desde el primer dispositivo de comunicación (10) al segundo dispositivo de comunicación (20) y
- 55
- 2h) Decodificación, en el segundo dispositivo de comunicación (20), de al menos algunos de los elementos codificados recibidos utilizando un decodificador lineal aleatorio con el fin de obtener los elementos del primer subconjunto de datos seleccionado, que no están contenidos en el segundo subconjunto de datos seleccionado del segundo conjunto de datos;
- 60
- 2i) Repetición de las etapas 2d) a 2h) hasta que todos los elementos del primer conjunto de datos, que no estaban inicialmente contenidos en el segundo conjunto de datos, hayan sido decodificados en el segundo dispositivo de comunicación (20).
- 65
2. El método según la reivindicación 1, en donde la etapa 1c) comprende la etapa adicional de creación de una

matriz de números aleatorios  $k \times m$ -Beta (Beta), en donde dichos  $m$  elementos del primer conjunto de datos están representados por un vector de columna de  $m$  elementos ( $V_A$ ) y dichos  $k$  elementos codificados se representan por un vector de columna de  $k$  elementos ( $V_V$ ) que se calcula realizando la multiplicación matricial de la matriz de número aleatorio Beta y el vector de columna de  $m$  elementos; en donde antes de realizar la etapa 1f) cada uno de los  $k$  elementos codificados es encapsulado junto con las entradas de la respectiva fila de la matriz de números aleatorios Beta en un paquete separado a transmitirse y en donde la etapa 1g) comprende la etapa de calcular la inversa de dicha matriz de números aleatorios Beta recibida, en donde el vector de columna de  $m$  elementos se calcula realizando la multiplicación matricial de la inversa de dicha matriz de números aleatorios Beta y el vector de columna de  $k$  elementos.

**3.** El método según la reivindicación 1, en donde la etapa 2c) comprende la etapa adicional de crear, con respecto a cada primer subconjunto de datos, de una matriz de números aleatorios Beta de  $k \times l_i$ , en donde  $l_i$  corresponde al número de elementos del primer subconjunto de datos y  $l_i$  es menor que  $m$ , en donde dichos  $l_i$  elementos del  $i$ -ésimo primer subconjunto de datos están representados por un vector de columna de  $l_i$  elementos y dichos  $k$  elementos codificados están representados por un vector de columna de  $k$  elementos, que se obtiene realizando la multiplicación matricial de la matriz de números aleatorios Beta y el vector de columna de  $l_i$  elementos; en donde antes de realizar la etapa 2g), cada uno de  $k$  elementos codificados se encapsula junto con las entradas de la fila respectiva de la matriz de números aleatorios Beta en un paquete separado para transmitirse y en donde

la etapa 2h) comprende la etapa de calcular la inversa de la matriz de números aleatorios Beta de  $k \times l_i$  recibida, en donde el vector de columna de  $l_i$  elementos se obtiene realizando la multiplicación matricial de la inversa de dicha matriz de números aleatorios Beta y el vector de columna de  $k$  elementos.

**4.** El método según cualquiera de las reivindicaciones precedentes, en donde antes de realizar la etapa de decodificación 1g), los elementos comunes entre el primer conjunto de datos y el segundo conjunto de datos se determina y en donde solamente los elementos codificados recibidos, que no están incluidos en el segundo conjunto de datos, son objeto de decodificación y/o en donde antes de realizar la etapa de decodificación 2h), los elementos comunes entre el segundo subconjunto de datos seleccionado y el primer subconjunto de datos correspondiente se determinan y en donde solamente los elementos codificados recibidos, que no están incluidos en el subconjunto de datos seleccionado del segundo subconjunto de datos, son objeto de decodificación.

**5.** El método según cualquiera de las reivindicaciones precedentes, en donde la etapa 2d) comprende las etapas siguientes:

- transmitir una demanda desde el segundo dispositivo de comunicación al primer dispositivo de comunicación para enviar un mensaje relativo a todos los elementos incluidos en el primer conjunto de datos;
- transmitir el mensaje en un paquete único al segundo dispositivo de comunicación y

en donde la etapa 2e) comprende las etapas siguientes:

- transmitir una demanda desde el segundo dispositivo de comunicación al primer dispositivo de comunicación para enviar un mensaje con respecto a todos los elementos incluidos en el primer subconjunto de datos seleccionado;
- transmitir la información respectiva en un paquete único al segundo dispositivo de comunicación.

**6.** El método según la reivindicación 5,

en donde la información comprende las cadenas aleatorias, denominadas *hashes*, de cada elemento incluido en el primer conjunto de datos y/o las *hashes* de cada elemento del primer subconjunto de datos seleccionado, en donde

la etapa 1d) comprende las etapas adicionales de:

generación, en el segundo dispositivo de comunicación, de la clave aleatoria *hash* de cada elemento incluido en el segundo conjunto de datos; la comparación de las *hashes* de cada elemento incluido en el primer conjunto de datos recibido desde el primer dispositivo de comunicación con las *hashes* de los elementos del segundo conjunto de datos con el fin de determinar el número total de elementos del primer conjunto de datos que no están incluidos en el segundo conjunto de datos y en donde

la etapa 2e) comprende las etapas adicionales de:

generación, en el segundo dispositivo de comunicación, de la *hash* de cada elemento incluido en el segundo subconjunto de datos seleccionado;

comparar las *hashes* de cada elemento del primer subconjunto de datos seleccionado recibido desde el primer

dispositivo de comunicación con las *hashes* de los elementos del segundo subconjunto de datos seleccionado con el fin de determinar el número de elementos del primer subconjunto de datos seleccionado que no están incluidos en el segundo subconjunto de datos seleccionado.

5 7. El método según cualquiera de las reivindicaciones precedentes, en donde la etapa 2g) comprende la etapa de demandar al primer dispositivo de comunicación para el envío de los k elementos codificados.

8. El método según cualquiera de las reivindicaciones precedentes, que comprende las siguientes etapas de:

10 determinar, en el segundo dispositivo de comunicación de si el número de elementos del segundo conjunto de datos o el número de elementos de un segundo subconjunto de datos seleccionado es menor que o igual al parámetro k;

15 si la respuesta es afirmativa, demandar al primer dispositivo de comunicación para transmitir todos los elementos del primer conjunto de datos o para transmitir todos los elementos del primer subconjunto de datos seleccionado, que corresponde al segundo subconjunto de datos seleccionado, al segundo dispositivo de comunicación.

9. Un sistema informático distribuido (1) para sincronizar conjuntos de datos memorizados en diferentes dispositivos de comunicación que están conectados a través de enlaces de comunicación, que comprende:

20 a) un primer dispositivo de comunicación (10) que comprende las características siguientes:

- un primer medio de memorización (18) para memorizar un primer conjunto de datos, incluyendo dicho primer conjunto de datos un número de m elementos,

25 - un codificador (11) para codificar el primer conjunto de datos utilizando un código lineal aleatorio para obtener un número de k elementos codificados, en donde k es un parámetro ajustable,

- un transmisor (13) para transmitir los k elementos codificados al segundo dispositivo de comunicación en respuesta a una demanda del segundo dispositivo de comunicación;

30 - un medio para realizar una partición de conjuntos en el primer conjunto de datos con el fin de obtener una arborescencia multinivel jerárquica, que comprende una pluralidad de primeros subconjuntos de datos;

35 - un medio para seleccionar un primer subconjunto de datos en respuesta a un mensaje recibido desde el segundo dispositivo de comunicación,

- en donde el codificador (11) está adaptado para codificar cada primer subconjunto de datos utilizando un código lineal aleatorio con el fin de obtener un número de k elementos codificados y

40 b) un segundo dispositivo de comunicación (20) conectado a través de un enlace de comunicación (30) con el primer dispositivo de comunicación (10), comprendiendo dicho segundo dispositivo de comunicación (20) las características siguientes:

45 - un primer medio de memorización (28) para memorizar un segundo conjunto de datos, incluyendo dicho segundo conjunto de datos un número de n elementos;

- un medio (26) para determinar el número total de elementos, que sean diferentes entre dicho primero y segundo conjunto de datos, sobre la base de un mensaje recibido desde el primer dispositivo de comunicación (10) y para determinar si el número total de diferentes elementos es menor que o igual al parámetro k;

50 - un decodificador (24) para decodificar, si el número de elementos diferentes es menor que o igual a k, los k elementos codificados recibidos utilizando un decodificador lineal aleatorio con el fin de obtener todos los elementos del primer conjunto de datos que no estén contenidos en el segundo conjunto de datos,

55 - un medio para realizar una partición de conjuntos en el segundo conjunto de datos con el fin de obtener una arborescencia multinivel jerárquica, que comprende una pluralidad de segundos subconjuntos de datos, en donde cada primer subconjunto de datos corresponde a un segundo subconjunto de datos separado del segundo conjunto de datos,

60 - un medio para seleccionar, si el número de diferentes elementos es mayor que k, al menos un segundo subconjunto de datos en conformidad con una regla predeterminada,

65 - un medio para determinar el número de elementos, que son diferentes entre el segundo subconjunto de datos seleccionado y el primer subconjunto de datos seleccionado sobre la base de un mensaje recibido desde el primer dispositivo de comunicación (10) en donde el primer subconjunto de datos seleccionado y el segundo subconjunto de datos seleccionado se corresponden entre sí y para determinar si el número de diferentes

elementos es menor que o igual al parámetro  $k$ ,

- 5 - en donde el decodificador (24) está adaptado para decodificar, si el número de elementos diferentes entre dicho segundo subconjunto de datos seleccionado y dicho primer subconjunto de datos seleccionado es menor que o igual a  $k$ , al menos algunos de los  $k$  elementos codificados recibidos del primer subconjunto de datos seleccionado utilizando un decodificador lineal aleatorio con el fin de obtener los elementos del primer subconjunto de datos, que no están contenidos en el segundo subconjunto de datos seleccionado del segundo conjunto de datos.
- 10 **10.** El sistema informático distribuido según la reivindicación 9, en donde el primer dispositivo de comunicación (10) comprende un segundo medio de memorización (12) para memorizar los elementos codificados.
- 15 **11.** El sistema informático distribuido según cualquiera de las reivindicaciones 8 a 9, en donde el codificador (11) está adaptado para utilizar una matriz de números aleatorios  $k \times m$ -Beta para codificar el primer conjunto de datos, en donde el primer dispositivo de comunicación comprende medios para la encapsulación de cada uno de los  $k$  elementos codificados junto con las entradas de la respectiva fila de la matriz de números aleatorios Beta en un paquete separado y en donde el decodificador (24) está adaptado para calcular la inversa de la matriz de números aleatorios Beta recibida.
- 20 **12.** El sistema informático distribuido según cualquiera de las reivindicaciones 9 a 11,
  - 25 - en donde el codificador (11) está adaptado para utilizar, para cada primer subconjunto de datos a codificarse, una matriz de números aleatorios  $k \times l_i$ -Beta, en donde  $l_i$  corresponde al número de elementos del  $i$ -ésimo primer subconjunto de datos y  $l_i$  es menor que  $m$ ,
  - en donde el primer dispositivo de comunicación comprende medios para la encapsulación de cada uno de los  $k$  elementos codificados de un respectivo primer subconjunto de datos junto con las entradas de la respectiva fila de la matriz de números aleatorios  $k \times l_i$ -Beta en un paquete separado y
  - 30 - en donde el decodificador (24) está adaptado para calcular la inversa de la matriz de números aleatorios de  $k \times l_i$ -Beta recibida.

35

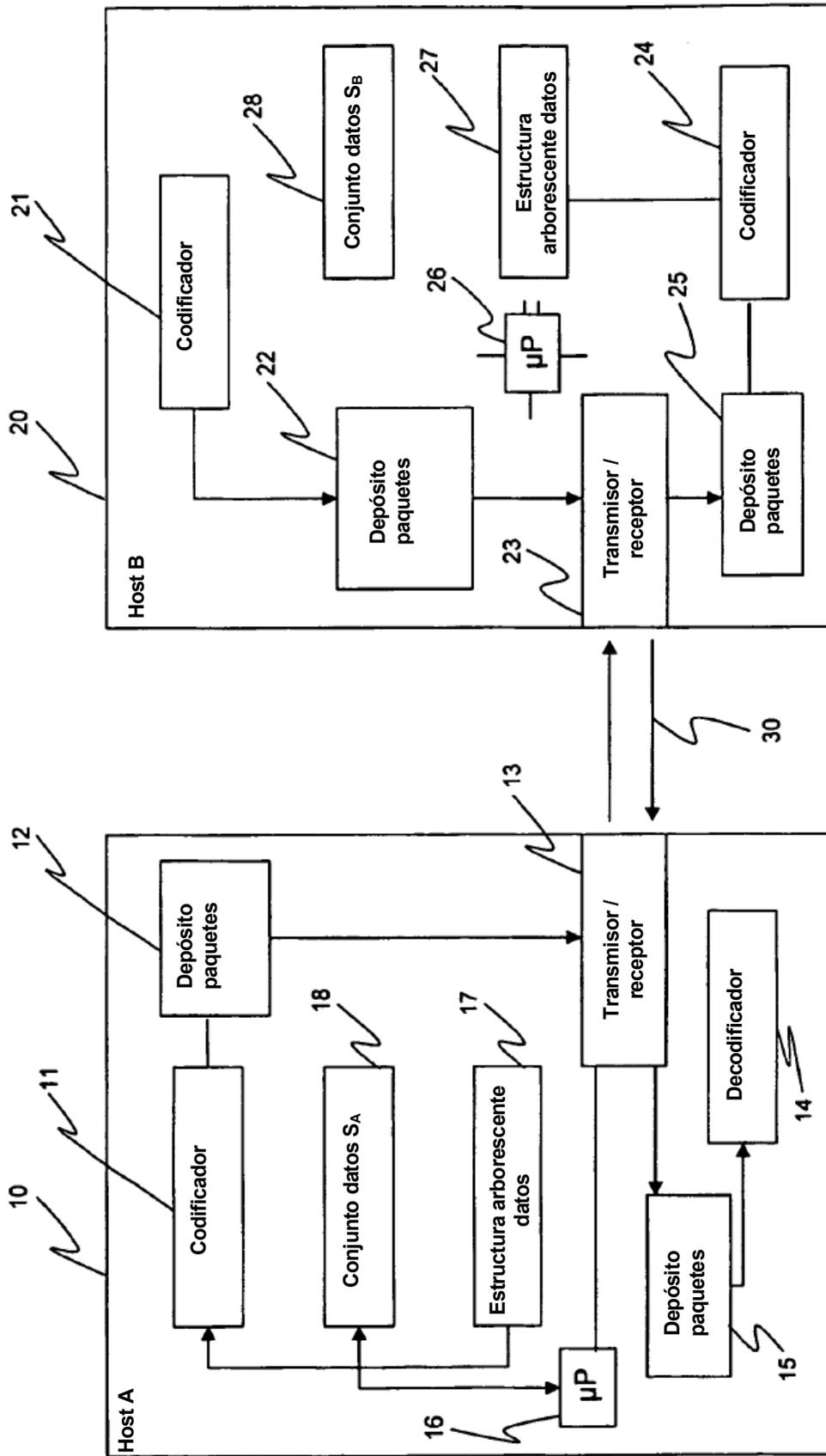


Fig. 1

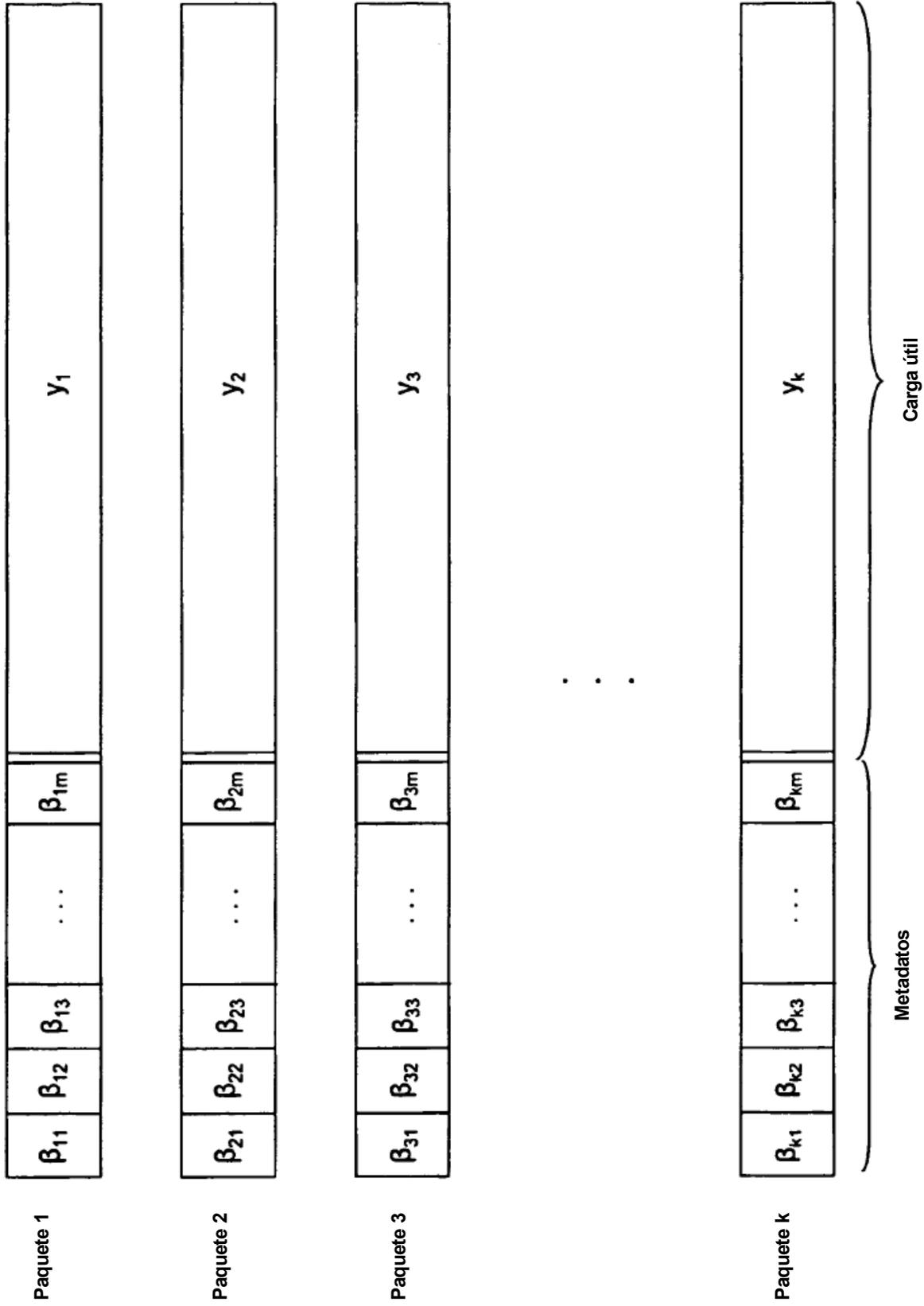
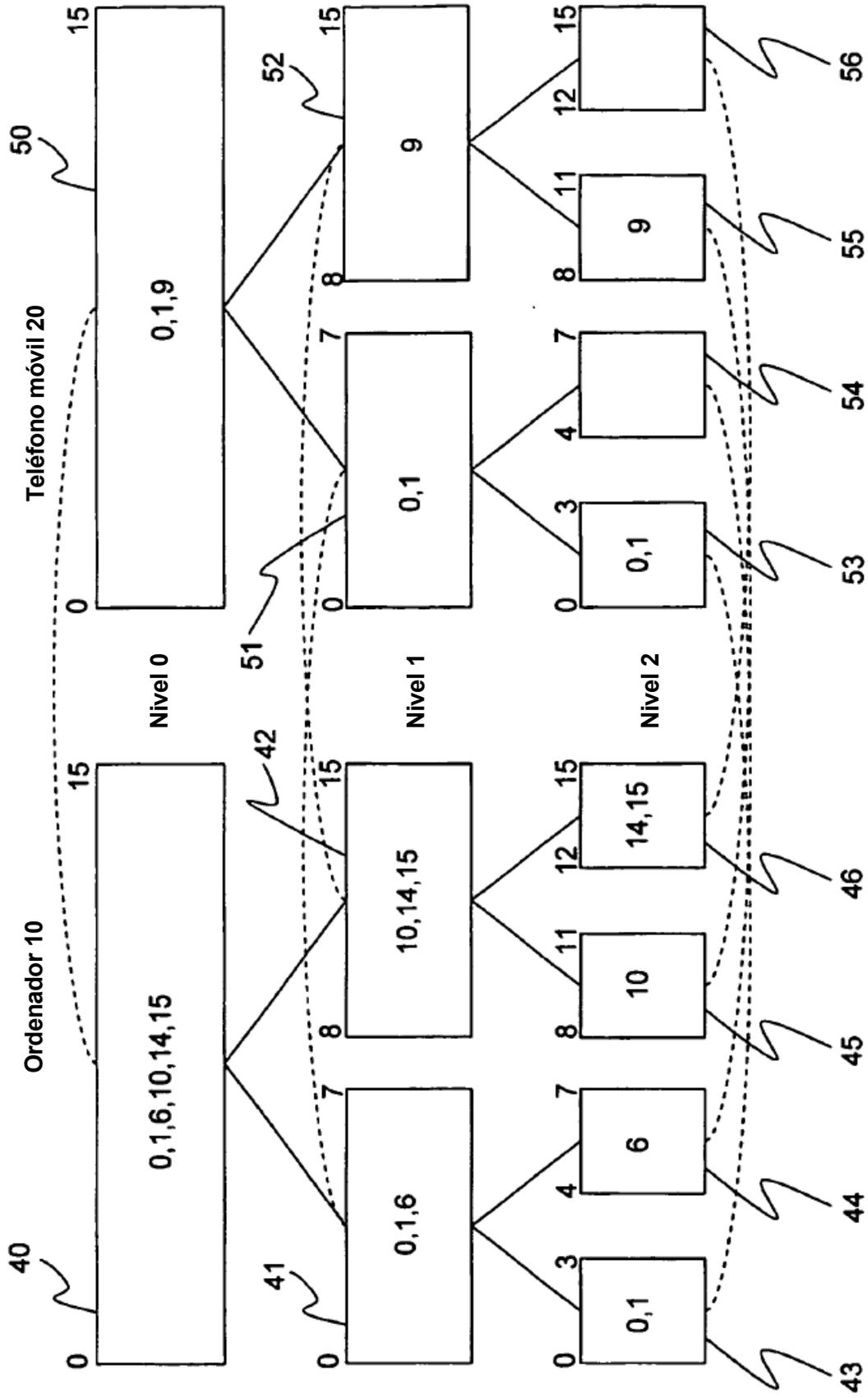


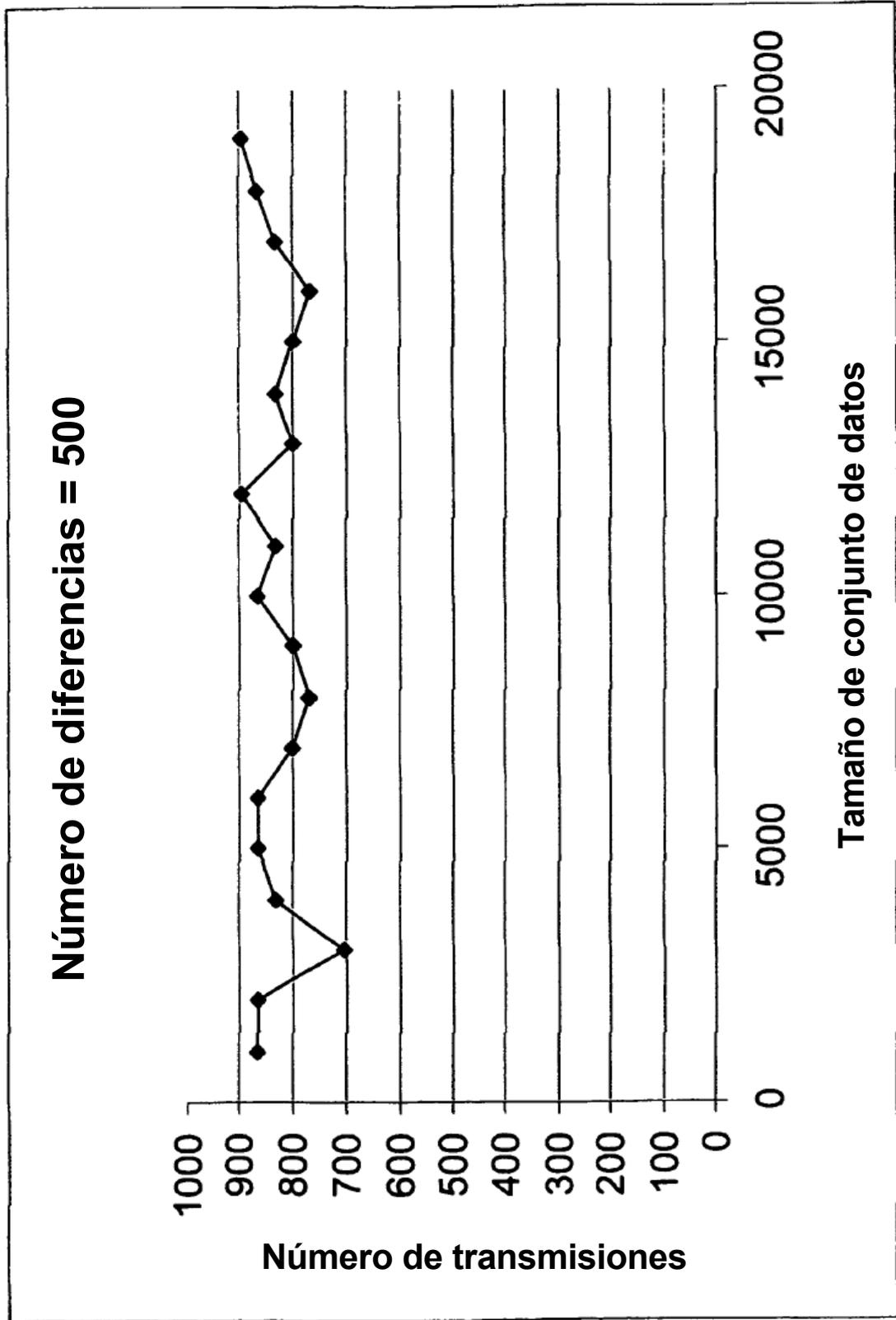
Fig. 2



$p = 2, b = 4$

Fig. 3

Fig. 4



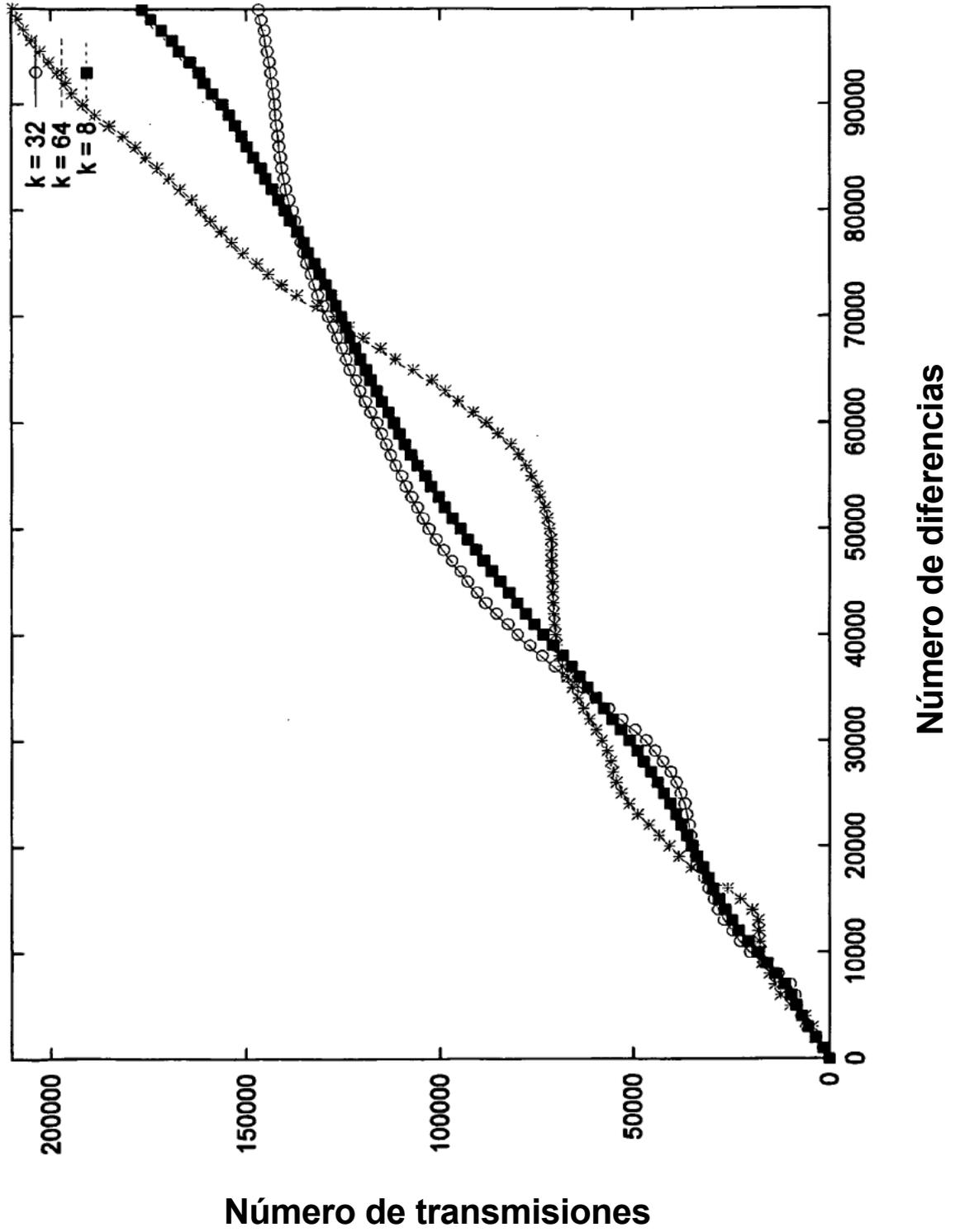


Fig. 5