

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 671 700**

51 Int. Cl.:

**G06F 17/30** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **13.02.2007 PCT/US2007/004048**

87 Fecha y número de publicación internacional: **07.09.2007 WO07100509**

96 Fecha de presentación y número de la solicitud europea: **13.02.2007 E 07750856 (2)**

97 Fecha y número de publicación de la concesión europea: **28.03.2018 EP 1999615**

54 Título: **Almacenamiento entre pares fiable y eficaz**

30 Prioridad:

**22.02.2006 US 359276**

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

**08.06.2018**

73 Titular/es:

**MICROSOFT TECHNOLOGY LICENSING, LLC**

**(100.0%)**

**One Microsoft Way**

**Redmond, WA 98052, US**

72 Inventor/es:

**LI, JIN**

74 Agente/Representante:

**CARPINTERO LÓPEZ, Mario**

ES 2 671 700 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

Almacenamiento entre pares fiable y eficaz

**Antecedentes**

5 En una aplicación entre pares (P2P), los pares traen consigo el ancho de banda de la red y/o los recursos de almacenamiento del disco duro cuando se unen al servicio P2P. A medida que crece la demanda de un sistema P2P, también crece la capacidad del sistema. Esto está en agudo contraste con un sistema cliente-servidor, donde la capacidad del servidor es fija y se paga por el proveedor del sistema cliente-servidor. Como resultado, un sistema P2P es más económico de ejecutar que un sistema cliente-servidor y es superior debido a que es escalable.

10 En un sistema P2P, el par 110046OMEM contribuye no solo el ancho de banda, sino también al espacio de almacenamiento para servir a los demás pares. El espacio de almacenamiento colectivo contribuido por los pares forma una nube de almacenamiento distribuido. Los datos pueden almacenarse en y recuperarse de la nube. El almacenamiento P2P puede usarse para varias aplicaciones. Una es la copia de seguridad distribuida. El par puede hacer una copia de seguridad de sus propios datos en la nube P2P. Cuando el par falla, los datos pueden restaurarse desde la nube. Otra aplicación P2P es el acceso a datos distribuidos. Debido a que el cliente puede recuperar datos simultáneamente de múltiples pares de datos, la recuperación P2P puede tener un mayor rendimiento en comparación con la recuperación de datos de una sola fuente. Otra aplicación es la visualización de películas a demanda. Un servidor de medios puede sembrar la nube P2P con archivos de películas de manera preventiva. Cuando un cliente está viendo la película, puede transmitir la película desde la nube P2P y el servidor, reduciendo de este modo la carga del servidor, reduciendo el tráfico en la red troncal y mejorando la calidad de la transmisión de películas.

A pesar de que los pares en la red P2P pueden actuar como servidores, se diferencian de los servidores web/de bases de datos comerciales en un aspecto importante: la fiabilidad. Debido a que un par suele ser un ordenador ordinario que soporta la aplicación P2P en su espacio libre en el disco duro y el ancho de banda inactivo, es mucho menos fiable que el servidor típico. El usuario puede elegir apagar el ordenador par o la aplicación P2P de vez en cuando. Una necesidad obligatoria, por ejemplo, la carga/descarga de archivos grandes, puede privar al par del ancho de banda necesario para la actividad P2P. El ordenador par puede estar fuera de línea debido a la necesidad de actualizar o aplicar parches de software/hardware, o debido a un ataque de virus. El hardware de ordenador y el enlace de red del par también son inherentemente mucho menos fiables que un ordenador servidor típico y sus enlaces de red comerciales, que están diseñados para proporcionar fiabilidad. Mientras que los clústeres de servidor/servidor comerciales están diseñados para una fiabilidad de "seis nueves" (con una tasa de fallo de  $10^{-6}$ , a esa tasa, se permiten alrededor de 30 segundos de tiempo de inactividad cada año), un buen par de consumo puede tener solo una fiabilidad de "dos nueves" (una tasa de fallo de  $10^{-2}$  o aproximadamente 15 minutos de inactividad todos los días), y no es raro que los pares solo tengan un 50 % (inactivos la mitad del tiempo) o incluso un 10 % de fiabilidad (inactivos el 90 % del tiempo).

35 La mayoría de las aplicaciones P2P, por ejemplo, la copia de seguridad y la recuperación de datos P2P, quieren mantener el mismo nivel de fiabilidad para el almacenamiento P2P que el del servidor (una fiabilidad de "seis nueves"). El desafío radica en cómo construir un almacenamiento P2P fiable y eficaz usando un ancho de banda y unos recursos de almacenamiento mínimos de los pares.

40 LI, J. y col.: "Distributed Hosting of Web Content with Erasure Coding and Unequal Weight Assignment" Procedimientos de la Conferencia Internacional IEEE 2004 sobre Multimedia y Expo vol. 3, 27 de junio de 2004 - 30 de junio de 2004, páginas 2087-2090 describe el desarrollo de dos tecnologías, es decir, organización de contenido jerárquico con asignación de peso desigual y codificación de borrado, para reducir la cantidad de contenido a distribuir en una red entre pares.

45 El documento WO 99/15954 A1 describe una gran asignación de bloques para sistemas de archivos basados en discos.

Cuenca-Acuna, F.M. y col.: "Autonomous Replication for High Availability in Unstructured P2P Systems", Reliable Distributed Systems, 2003, Proceedings, 22º Simposio internacional, del 6 al 8 de octubre de 2003, Piscataway, NJ, USA, IEEE, 6 de octubre de 2003, páginas 99-108 describe un algoritmo que usa decisiones aleatorias extensamente junto con una aplicación novedosa de un código de borrado para tolerar acciones autónomas entre pares, así como la caducidad en el estado global débilmente sincronizado.

50 Por lo tanto, el objetivo de la invención reivindicada es proporcionar un procedimiento mejorado implementado por ordenador para codificar, un medio/sistema legible por ordenador, con el fin de mejorar la codificación resistente al borrado.

Este objetivo se resuelve por el objeto de las reivindicaciones independientes.

55 Las realizaciones preferidas están definidas por las reivindicaciones dependientes.

Se presenta un sistema y un procedimiento de almacenamiento de codificación adaptativa para almacenar datos de manera eficaz y fiable en una red entre pares (P2P). El sistema y el procedimiento de almacenamiento de codificación adaptativa ajustan un número de fragmentos para la codificación resistente al borrado (ERC), el número de fragmentos de ERC, basándose en el tamaño de archivo almacenado y distribuido.

Un número de realizaciones del sistema de almacenamiento de codificación adaptativa emplean procedimientos para mejorar la eficacia y la fiabilidad de una red P2P. Por ejemplo, en una realización, los datos dinámicos pequeños se desvían a los pares más fiables o incluso a un servidor, si está disponible el soporte de componentes de servidor. Además, en otra realización, para una red P2P balanceada, a los pares que no son fiables y están distribuyendo archivos más pequeños se les permite distribuir menos datos.

Se observa que mientras que las limitaciones anteriores en los sistemas de almacenamiento y distribución entre pares existentes descritos en la sección de Antecedentes se pueden resolver mediante una implementación específica del sistema de almacenamiento de codificación adaptativa de acuerdo con la presente invención, este sistema y procedimiento no está limitado de ninguna manera a implementaciones que solo resuelven una o todas las desventajas observadas. Más bien, el presente sistema y procedimiento tiene una aplicación mucho más amplia como se hará evidente a partir de las descripciones siguientes.

Además, debería observarse que este Sumario se proporciona para introducir una selección de conceptos, de manera simplificada, que se describen más adelante en la Descripción detallada. Este Sumario no pretende identificar las características clave o las características esenciales del objeto reclamado, ni está destinado a usarse como una ayuda para determinar el ámbito del objeto reclamado.

### **Descripción de los dibujos**

Las características, aspectos y ventajas específicos del sistema de almacenamiento de codificación adaptativa se entenderán mejor con respecto a la siguiente descripción, las reivindicaciones adjuntas, y los dibujos adjuntos donde:

La figura 1 es un diagrama de sistema general que representa un dispositivo informático de fin general que constituye un sistema a modo de ejemplo que implementa un sistema y un procedimiento de almacenamiento de codificación adaptativa como se describe en el presente documento.

La figura 2 ilustra una red entre pares (P2P) a modo de ejemplo que puede usarse con el sistema y el procedimiento de almacenamiento de codificación adaptativa, como se describe en el presente documento.

La figura 3 proporciona una gráfica que muestra el número de pares de almacenamiento de información para lograr la fiabilidad deseada de  $10^6$ .

La figura 4 proporciona una gráfica que muestra la fiabilidad de par y la relación de replicación deseada.

La figura 5 proporciona una gráfica que muestra el número de pares de almacenamiento de información necesarios para lograr la fiabilidad deseada de  $10^6$  usando la codificación resistente al borrado.

La figura 6 proporciona una gráfica del número de fragmentos de ERC y el tamaño de archivo adecuado asociado para el almacenamiento de información en una red P2P.

La figura 7 proporciona una gráfica que representa el uso de ancho de banda entre pares en una configuración P2P con una ERC adaptativa y una ERC fija (con una fiabilidad de par = 50 %).

La figura 8 proporciona una gráfica que representa el uso de ancho de banda entre pares en una configuración P2P con una ERC adaptativa y una ERC fija (con una fiabilidad de par = 99 %).

La figura 9 representa una realización del procedimiento de almacenamiento de codificación adaptativa.

La figura 10 representa un diagrama de flujo operacional a modo de ejemplo que muestra cómo se emplea la técnica de almacenamiento de codificación adaptativa en una red P2P.

La figura 11 representa una realización del sistema y procedimiento de almacenamiento de codificación adaptativa que implementa un procedimiento para optimizar la eficacia de almacenamiento de una red P2P.

La figura 12 representa otra realización del sistema y procedimiento de almacenamiento de codificación adaptativa que implementa procedimientos para optimizar la eficacia de almacenamiento de un sistema P2P.

La figura 13 representa una realización del sistema y procedimiento de almacenamiento de codificación adaptativa que emplea una copia de seguridad P2P con soporte de servidor.

### **Descripción detallada**

En la siguiente descripción de las realizaciones preferidas del presente sistema de almacenamiento de codificación adaptativa, se hace referencia a los dibujos adjuntos que forman una parte de la misma, y en los que se muestran a modo de ilustración las realizaciones específicas en las que puede practicarse el sistema de almacenamiento de codificación adaptativa. Se entiende que pueden utilizarse otras realizaciones y pueden realizarse cambios estructurales sin alejarse del ámbito del presente sistema de almacenamiento de codificación adaptativa.

#### **1.0 Entorno operativo a modo de ejemplo:**

La figura 1 ilustra un ejemplo de un entorno 100 de sistema informático adecuado en el que puede implementarse la invención. El entorno 100 de sistema informático es solo un ejemplo de un entorno informático adecuado y no pretende sugerir ninguna limitación en cuanto al ámbito de uso o la funcionalidad de la invención. Tampoco debería interpretarse que el entorno 100 informático en cuanto a que tiene alguna dependencia o requisito relacionado con uno cualquiera o una combinación de componentes ilustrados en el entorno 100 operativo a modo de ejemplo.

La invención es operativa con otros numerosos entornos o configuraciones de sistema informático de fin general o de fin especial. Los ejemplos de sistemas, entornos y/o configuraciones informáticos bien conocidos que pueden ser

adecuados para su uso con la invención incluyen, pero no se limitan a, ordenadores personales, ordenadores servidores, ordenadores de mano, ordenadores portátiles o móviles o dispositivos de comunicaciones tales como teléfonos móviles y PDA, sistemas multiprocesador, sistemas basados en microprocesador, decodificadores, electrónica de consumo programable, PC en red, minicomputadoras, ordenadores centrales, entornos informáticos distribuidos que incluyen cualquiera de los sistemas o dispositivos anteriores y similares.

La invención puede describirse en el contexto general de unas instrucciones ejecutables por ordenador, tales como módulos de programa, que se ejecutan por un ordenador en combinación con módulos de hardware, incluyendo componentes de un conjunto 198 de micrófonos. En general, los módulos de programa incluyen rutinas, programas, objetos, componentes, estructuras de datos, etc., que realizan tareas específicas o implementan tipos de datos abstractos específicos. La invención también puede ponerse en práctica en entornos informáticos distribuidos en los que las tareas se realizan mediante dispositivos de procesamiento remoto que están enlazados a través de una red de comunicaciones. En un entorno informático distribuido, los módulos de programa pueden localizarse tanto en medios de almacenamiento informático locales como remotos incluyendo los dispositivos de almacenamiento de memoria. Haciendo referencia a la figura 1, un sistema a modo de ejemplo para implementar la invención incluye un dispositivo informático de fin general en forma de un ordenador 110.

Los componentes del ordenador 110 pueden incluir, pero no se limitan a, una unidad 120 de procesamiento, una memoria 130 de sistema, y un bus 121 de sistema que acopla diversos componentes del sistema incluyendo la memoria de sistema a la unidad 120 de procesamiento. El bus 121 de sistema puede ser cualquiera de diversos tipos de estructuras de bus que incluyen un bus de memoria o un controlador de memoria, un bus periférico y un bus local que usa cualquiera de una variedad de arquitecturas de bus. A modo de ejemplo, y no de limitación, tales arquitecturas incluyen el bus de arquitectura estándar de la industria (ISA), el bus de arquitectura de microcanal (MCA), el bus ISA mejorado (EISA), el bus local VESA (Asociación de estándares electrónicos y video) y el bus de interconexión de componentes periféricos (PCI) también conocido como bus Mezzanine.

El ordenador 110 incluye normalmente una variedad de medios legibles por ordenador. Los medios legibles por ordenador pueden ser cualquier medio disponible al que se pueda acceder mediante el ordenador 110 e incluye medios volátiles y no volátiles, medios extraíbles y no extraíbles. A modo de ejemplo, y no de limitación, los medios legibles por ordenador pueden comprender medios de almacenamiento informático y medios de comunicación. Los medios de almacenamiento informático incluyen medios extraíbles y no extraíbles volátiles y no volátiles implementados en cualquier procedimiento o tecnología para el almacenamiento de información tal como instrucciones legibles por ordenador, estructuras de datos, módulos de programa u otros datos.

Los medios de almacenamiento informático incluyen, pero no se limita a, RAM, ROM, PROM, EPROM, EEPROM, memoria flash u otra tecnología de memoria; CD-ROM, discos versátiles digitales (DVD) u otro almacenamiento en disco óptico; casetes magnéticos, cinta magnética, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético; o cualquier otro medio que pueda usarse para almacenar la información deseada y al que pueda accederse desde el ordenador 110. Los medios de comunicación normalmente incorporan instrucciones legibles por ordenador, estructuras de datos, módulos de programa u otros datos en una señal de datos modulada tal como una onda portadora u otro mecanismo de transporte e incluye cualquier medio de entrega de información. La expresión "señal de datos modulada" significa una señal que tiene una o más de sus características establecidas o cambiadas de tal manera que codifican información en la señal. A modo de ejemplo, y no de limitación, los medios de comunicación incluyen medios cableados tales como una red cableada o conexión de cableado directo, y medios inalámbricos tales como medios acústicos, RF, infrarrojos y otros medios inalámbricos. Las combinaciones de cualquiera de los anteriores también deberían incluirse dentro del ámbito de los medios legibles por ordenador.

La memoria 130 de sistema incluye unos medios de almacenamiento informático en forma de memoria volátil y/o no volátil tal como la memoria 131 de solo lectura (ROM) y la memoria 132 de acceso aleatorio (RAM). Un sistema 133 de entrada/salida básico (BIOS), que contiene las rutinas básicas que ayudan a transferir información entre elementos dentro del ordenador 110, tal como durante la puesta en marcha, se almacena normalmente en la ROM 131. La RAM 132 normalmente contiene datos y/o módulos de programas que pueden accederse de manera inmediata y/o actualmente se están operando por la unidad 120 de procesamiento. A modo de ejemplo, y no de limitación, la figura 1 ilustra el sistema 134 operativo, los programas 135 de aplicación, otros módulos 136 de programa y los datos 137 de programa.

El ordenador 110 también puede incluir otros medios de almacenamiento informático extraíbles/no extraíbles, volátiles/no volátiles. A modo de ejemplo solamente, la figura 1 ilustra una unidad 141 de disco duro que lee de o escribe en medios magnéticos no volátiles no extraíbles, una unidad 151 de disco magnético que lee de o escribe en un disco 152 magnético no volátil extraíble, y una unidad 155 de disco óptico que lee de o escribe en un disco 156 óptico no volátil extraíble tal como un CDROM u otro medio óptico. Otros medios de almacenamiento informático extraíbles/no extraíbles, volátiles/no volátiles que pueden usarse en el entorno operativo a modo de ejemplo incluyen, pero no se limitan a, son casetes de cinta magnética, tarjetas de memoria flash, discos versátiles digitales, cinta de video digital, RAM de estado sólido, ROM de estado sólido, y similares. La unidad 141 de disco duro está normalmente conectada al bus 121 de sistema a través de una interfaz de memoria no extraíble tal como la interfaz 140, y la unidad 151 de disco magnético y la unidad 155 de disco óptico están normalmente conectadas al bus 121 de sistema mediante una interfaz de memoria extraíble, tal como la interfaz 150.

Las unidades y sus medios de almacenamiento informáticos asociados tratados e ilustrados anteriormente en la figura 1, proporcionan un almacenamiento de instrucciones legibles por ordenador, estructuras de datos, módulos de programa y otros datos para el ordenador 110. En la figura 1, por ejemplo, la unidad 141 de disco duro se ilustra como un almacenamiento de sistema 144 operativo, programas 145 de aplicación, otros módulos 146 de programa y datos 147 de programa. Obsérvese que estos componentes pueden ser iguales o diferentes del sistema 134 operativo, los programas 135 de aplicación, otros módulos 136 de programa y los datos 137 de programa. El sistema 144 operativo, los programas 145 de aplicación, otros módulos 146 de programa y los datos 147 de programa reciben en este caso números diferentes para ilustrar que, como mínimo, son copias diferentes. Un usuario puede introducir comandos e información en el ordenador 110 a través de dispositivos de entrada tales como un teclado 162 y un dispositivo 161 señalador, comúnmente denominado como mouse, bola de seguimiento o teclado táctil.

Otros dispositivos de entrada (no mostrados) pueden incluir un palanca de mando, un mando para juegos, una antena parabólica, un escáner, un receptor de radio y un televisor o receptor de vídeo de difusión, o similares. Estos y otros dispositivos de entrada a menudo están conectados a la unidad 120 de procesamiento a través de una interfaz 160 de entrada de usuario cableada o inalámbrica que está acoplada al bus 121 de sistema, pero que puede estar conectada mediante otras estructuras de interfaz y de bus convencionales, tales como, por ejemplo, un puerto paralelo, un puerto de juego, un bus serie universal (USB), una interfaz IEEE 1394, una interfaz inalámbrica Bluetooth™, una interfaz inalámbrica IEEE 802.11, etc. Además, el ordenador 110 también puede incluir un dispositivo de entrada de voz o audio, tal como un micrófono o un conjunto 198 de micrófonos, así como un altavoz 197 u otro dispositivo de salida de sonido conectado a través de una interfaz 199 de audio, incluyendo también unas interfaces convencionales cableadas o inalámbricas, tales como, por ejemplo, paralelo, serie, USB, IEEE 1394, Bluetooth™, etc.

Un monitor 191 u otro tipo de dispositivo de visualización también está conectado al bus 121 de sistema a través de una interfaz, tal como una interfaz 190 de vídeo. Además del monitor, los ordenadores también pueden incluir otros dispositivos de salida periféricos tales como una impresora 196, que puede estar conectada a través de una interfaz 195 periférica de salida.

El ordenador 110 puede operar en un entorno de red usando conexiones lógicas a uno o más ordenadores remotos, tales como un ordenador 180 remoto. El ordenador 180 remoto puede ser un ordenador personal, un servidor, un encaminador, un PC de red, un dispositivo par, u otro nodo de red común, y normalmente incluye muchos o todos los elementos descritos anteriormente en relación con el ordenador 110, aunque en la figura 1 se ha ilustrado solamente un dispositivo 181 de almacenamiento de memoria. Las conexiones lógicas representadas en la figura 1 incluyen una red 171 de área local (LAN) y una red 173 de área extensa (WAN), pero también pueden incluir otras redes. Dichos entornos de red son comunes en las oficinas, las redes informáticas de toda la empresa, intranets e Internet.

Cuando se usa en un entorno de red LAN, el ordenador 110 está conectado a la LAN 171 a través de una interfaz de red o adaptador 170. Cuando se usa en un entorno de red WAN, el ordenador 110 incluye normalmente un módem 172 u otro medio para establecer comunicaciones a través de la WAN 173, tal como Internet. El módem 172, que puede ser interno o externo, puede estar conectado al bus 121 de sistema a través de la interfaz 160 de entrada de usuario, u otro mecanismo apropiado. En un entorno de red, los módulos de programa representados en relación con el ordenador 110, o partes de los mismos, pueden almacenarse en el dispositivo de almacenamiento de memoria remoto. A modo de ejemplo, y no de limitación, la figura 1 ilustra unos programas 185 de aplicación remota que residen en el dispositivo 181 de memoria. Se apreciará que las conexiones de red mostradas son a modo de ejemplo y pueden usarse otros medios para establecer un enlace de comunicaciones entre los ordenadores.

En general, el sistema de almacenamiento de codificación adaptativa opera en una red P2P, tal como la red ilustrada en la figura. 2. Para una sesión de transmisión de datos específica, se define un "servidor" 200 como un nodo en la red P2P que origina inicialmente los datos o los medios de transmisión; se define un "cliente" (o receptor) 210 como un nodo que actualmente solicita los datos; y se define un "par 220 de servicio" como un nodo que sirve al cliente una copia completa o parcial de los datos.

En general, el servidor 200, el cliente 210 y el par 220 de servicio son todos nodos de usuario final conectados a una red tal como Internet. Debido a que el servidor 200 siempre es capaz de servir los datos, el nodo servidor también actúa como un par 220 de servicio. El nodo servidor 200 también puede realizar funcionalidades administrativas que no pueden realizarse por un par 220 de servicio, por ejemplo, mantener una lista pares de servicios disponibles, realizar la funcionalidad de gestión de derechos digitales (DRM), etc. Además, al igual que con los esquemas P2P convencionales, el sistema de almacenamiento de codificación adaptativa descrito en el presente documento se beneficia de una mayor eficacia a medida que se despliegan más y más nodos 220 de pares. En particular, a medida que aumenta el número de nodos 220 de pares, la carga en el servidor 200 de datos disminuirá, haciéndose de este modo menos costosa de ejecutar, mientras que cada nodo cliente 210 podrá recibir una calidad de datos mucho mejor durante una sesión de transferencia de datos específica.

Además, debería ser evidente que el papel de los nodos particulares puede cambiar. Por ejemplo, un nodo específico puede actuar como el cliente 210 en una transferencia de datos específica, mientras que actúa como un par 220 de servicio en otra sesión. Además, los nodos específicos pueden actuar simultáneamente tanto como

5 nodos 210 de cliente como servidores 200 o pares 220 de servicio para enviar simultáneamente uno o más archivos de datos, o partes de estos archivos, mientras que reciben otros datos de uno o más pares de servicio.

5 Durante una transmisión de datos, el cliente 200 localiza en primer lugar un número de pares 220 cercanos que retienen algunos o todos los datos deseados, y a continuación recibe los datos desde los múltiples pares (que pueden incluir el servidor 200). Consecuentemente, cada par 220 de servicio actúa para ayudar al servidor 200 reduciendo la carga de subida general atendiendo a una parte de la solicitud de descarga del cliente 210. Como resultado, el cliente 210, especialmente en el caso donde hay muchos clientes, puede recibir a menudo una calidad de datos mucho mejor, ya que hay un ancho de banda de servicio significativamente mayor disponible cuando hay muchos pares 220 de servicio para ayudar al servidor 200.

10 Habiéndose tratado ahora el entorno operativo a modo de ejemplo, las partes restantes de esta sección de la descripción se dedicarán a una descripción de los módulos de programa que realizan el sistema y el procedimiento de almacenamiento de codificación adaptativa.

2.0 ALMACENAMIENTO ENTRE PARES FIABLE Y EFICAZ.

15 El sistema de almacenamiento de codificación adaptativa proporciona un esquema de codificación resistente al borrado (ERC) adaptativa que determina de manera adaptativa si usar o no la codificación ERC y emplea el número óptimo de fragmentos a usarse por la codificación ERC para un tamaño de archivo dado para una fiabilidad y una eficacia óptimas. El número de fragmentos usados para la codificación ERC de un archivo se denominará "número de fragmentos de ERC" para los fines de esta exposición. Los siguientes párrafos proporcionan una exposición sobre la fiabilidad y la eficacia de almacenamiento entre pares (P2P) y el uso de la ERC en las redes P2P, así como  
20 una exposición del número de fragmentos de ERC usados. A continuación, se tratan diversas realizaciones del sistema y el procedimiento de almacenamiento de codificación adaptativa.

2.1 Fiabilidad en el almacenamiento P2P: Redundancia de datos

25 La solución a propósito para llevar fiabilidad a un sistema con partes poco fiables es usar la redundancia. Si cada par individual en la red tiene una fiabilidad de  $p$ , para lograr una fiabilidad deseada de  $p_0$ , se puede replicar simplemente la información a  $n$  pares:

$$n = \log(1 - p_0) / \log(1 - p), \tag{1}$$

donde  $n$  es el número de pares que retienen la información. En el momento de la recuperación, el cliente puede ponerse en contacto con la información que almacena los pares uno por uno. Siempre que uno de los pares que almacenan la información esté en línea, la información puede recuperarse de manera fiable.

30 A pesar de lograr la fiabilidad, la estrategia de replicación simple no es eficaz. La figura 3 representa gráficamente el número de pares que almacenan información necesarios para lograr la fiabilidad de "seis nueves". Con una fiabilidad de pares del 50 %, se necesita replicar y almacenar la información en 20 pares. Esto lleva a 20 veces más ancho de banda y espacio de almacenamiento para distribuir y almacenar la información. Obviamente, la eficacia se ha sacrificado a cambio de fiabilidad de la información.

35 2.2 Codificación resistente al borrado en P2P

Para mejorar la eficacia mientras que todavía se mantiene la misma fiabilidad, la ERC puede ser una herramienta útil. La ERC divide el archivo original en  $k$  fragmentos originales  $\{x_i\}$ ,  $i = 0, \dots, k-1$ , cada uno de los cuales es un vector sobre el campo Galois  $GF(q)$ , donde  $q$  es el orden del campo. Digamos que se codifica un archivo de 64 KB de longitud, si se usa  $q = 2^{16}$  and  $k = 16$ , cada fragmento tendrá 4 KB y consistirá en una palabra de 2 K, siendo  
40 cada palabra un elemento de  $GF(2^{16})$ . A continuación, la ERC genera fragmentos codificados a partir de los fragmentos originales. Un fragmento codificado de ERC se forma por la operación:

$$c_j = \mathbf{G}_j [x_0 \quad x_1 \quad \dots \quad x_{k-1}]', \tag{2}$$

45 donde  $c_j$  es un fragmento codificado,  $\mathbf{G}_i$  es un  $k$  vector generador dimensional, y la ecuación (2) es una multiplicación de matrices, todo en  $GF(q)$ . En el momento de la decodificación, el par recolecta los  $m$  fragmentos codificados, donde  $m$  es un número igual o un poco más grande que  $k$ , e intenta decodificar los  $k$  fragmentos originales. Esto es equivalente a resolver la ecuación:

$$\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{m-1} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_0 \\ \mathbf{G}_1 \\ \vdots \\ \mathbf{G}_{k-1} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{k-1} \end{bmatrix}, \tag{3}$$

Si la matriz formada por los vectores de generador tiene un rango completo, pueden recuperarse los mensajes originales.

Hay muchas ERC disponibles. Una específicamente interesante es el código Reed-Solomon (RS). El código RS usa vectores de generador estructurados, y es la distancia máxima separable (MDS). Como resultado, cualquier  $k$  fragmento codificado distintivo podrá decodificar los fragmentos originales. Otra ventaja del código RS es que el fragmento codificado puede identificarse y gestionarse fácilmente por el índice  $l$  del vector de generador, facilitando de este modo la detección de códigos RS duplicados. En la siguiente exposición de la ERC, se supone que se usa el código RS. Sin embargo, el sistema de almacenamiento de codificación adaptativa puede implementarse con cualquier cantidad de ERC convencionales.

### 2.3 ERC: Número de fragmentos.

Usando la ERC en el almacenamiento P2P, un archivo de datos se distribuye a más pares, pero cada par solo necesita almacenar un fragmento codificado que es de tamaño  $1/k$  del archivo original, lo que lleva a una reducción general en el espacio ancho de banda y almacenamiento necesarios para alcanzar el mismo nivel de fiabilidad y, por lo tanto, una mejora de la eficacia. Sea  $n_1$  el número de pares que los fragmentos codificados deben distribuirse para alcanzar un cierto nivel de fiabilidad deseado. Ya que el código RS es un código MDS, los  $k$  pares que retienen los  $k$  fragmentos codificados distintivos serán suficientes para recuperar el archivo original. La probabilidad de que haya exactamente  $m$  pares disponibles puede calcularse a través de la distribución binomial:

$$p(m, n_1) = \binom{n_1}{m} p^m (1-p)^{n_1-m}. \quad (4)$$

Por lo tanto, puede calcularse  $n_1$  a partir de  $p$ ,  $p_0$  y  $k$  como:

$$n_1 = \arg \min_{n_1} \left\{ \sum_{m < k} \binom{n_1}{m} p^m (1-p)^{n_1-m} < 1 - p_0 \right\}. \quad (5)$$

La relación de replicación  $r$  se define como:

$$r = n_1 / k. \quad (6)$$

La relación de replicación  $r$  es un buen indicador de la eficacia, ya que  $r$  copias de archivos deben distribuirse y almacenarse en la nube P2P.

Se muestra en la figura 4, la relación de replicación deseada para lograr la fiabilidad de “seis nueves” para diferentes números de fragmentos  $k$  de ERC. Se observa que el uso de ERC reduce en gran medida la relación de replicación necesaria. Comparando los números de fragmentos no ERC ( $k = 1$ ) y ERC  $k = 256$ , la relación de replicación deseada disminuye desde  $r = 132$  a  $r = 13,1$  para la fiabilidad de pares del 10 %, desde  $r = 20$  a  $r = 2,5$  para la fiabilidad de pares del 50 %, y desde  $r = 3$  a  $r = 1,05$  para la fiabilidad de pares del 99 %. La ERC puede mejorar la eficacia sin sacrificar la fiabilidad.

Se observa también que un mayor número de fragmentos de ERC reduce aún más la relación de replicación. Con una fiabilidad de pares del 50 %, pasar de  $k = 8$  a 16, 32, 64, 128 y 256 conduce a una reducción de la relación de replicación de  $r = 5,75$  a 4,375, 3,53, 3,02, 2,68 y 2,48. La mejora de eficacia correspondiente es del 24 %, 19 %, 15 %, 11 % y 8 %, respectivamente. Esto parece sugerir que debería usarse un gran número de fragmentos de ERC para una mayor eficacia.

Sin embargo, un mayor número de fragmentos de ERC implica que se necesitan más pares para almacenar y recuperar los fragmentos codificados. Como se muestra en la figura 5, se representa gráficamente el número de pares que necesitan retener los fragmentos codificados para lograr la fiabilidad de “seis nueves”. De nuevo, con un 50 % de fiabilidad de pares, pasar desde  $k = 8$  a 16, 32, 64, 128 y 256 aumenta el número de pares de almacenamiento de información desde  $n_1 = 46$  a 70, 113, 193, 343 y 630. Cada duplicación de  $k$  resulta en un 52 %, 61 %, 71 %, 78 %, 84 % más pares necesarios para almacenar la información. La duplicación de  $k$  también necesita al menos el doble del número de pares a conectar durante la recuperación de la información.

En las redes P2P más prácticas, el establecimiento de una conexión entre los pares necesita una cantidad no trivial de sobrecarga. Una parte de la sobrecarga puede atribuirse a la recuperación de una identidad de pares adecuada y a encontrar la ruta de encaminamiento adecuada (por ejemplo, a través de una tabla Hash distribuida (DHT)). Otra parte de la sobrecarga se debe a la necesidad de invocar ciertos algoritmos de traducción de direcciones de red (NAT), por ejemplo, STUN (recorrido simple de UDP a través de NAT) si uno o ambos pares están detrás de NAT. Suponiendo que la sobrecarga promedio para establecer la conexión entre dos pares es una *sobrecarga* (establecida en 16 KB en este ejemplo), puede calcularse el ancho de banda de red total necesario para almacenar un archivo de tamaño  $s$  para que sea:

$$\text{ancho de banda}_{\text{almacenar}} = s * r + n_1 * \text{sobrecarga}. \quad (7)$$

Con la ecuación (7), se reconoce que un mayor número de fragmentos de ERC no siempre conduce a la mejor eficacia. En cambio, para un archivo pequeño, debería usarse un número pequeño de fragmentos de ERC o incluso no ERC. Se calcula el ancho de banda total necesario en la ecuación (7) para diferentes tamaños de archivo y número de fragmentos de ERC, y se representan gráficamente las curvas mostradas en la figura 6. El límite entre los diferentes números de fragmentos de ERC es el intervalo de tamaño de archivo óptimo adecuado para un número de fragmentos de ERC específico. Por ejemplo, la curva inferior de la figura 6 muestra el límite de tamaño de archivo por debajo del cual debería usarse no ERC, y por encima del cual debería usarse ERC con un número de fragmentos  $k = 2$ . Una observación interesante es que el límite de tamaño de archivo es relativamente insensible a la disponibilidad de los pares, lo que simplifica enormemente la elección del parámetro de fragmento de ERC óptimo. En general, para un archivo menor que aproximadamente 10 KB, no debería usarse ERC. Para la ERC con un número de fragmentos  $k = 2, 4, 8, 16, 32, 128$  y  $256$ , el intervalo de tamaño de archivo más adecuado es de aproximadamente 10-33 KB, 33-100 KB, 100-310 KB, 310-950 KB, 950 KB-2,9 MB, 2,9 MB-8,9 MB, 8,9-26 MB, > 26 MB, respectivamente.

#### 2.4 Esquema de ERC adaptativa

El sistema y el procedimiento de almacenamiento de codificación adaptativa eligen de manera adaptativa el número de fragmentos de ERC apropiado para almacenar de manera eficaz el contenido en una red P2P de manera fiable. Usando la curva de límite de tamaño de archivo establecida en la figura 6, una realización del sistema elige de manera adaptativa usar no ERC, y ERC con un número de fragmentos de  $k = 2, 4, 8, 16, 32, 64, 128, 256$  para diferentes tamaños de archivo. El enfoque de ERC adaptativa se compara con la ERC de parámetro fijo, y la diferencia en el uso del ancho de banda de la red se muestra en la figura 7 y en la figura 8, donde la fiabilidad de pares es del 50 % y del 99 %, respectivamente. Comparado con el uso de un número de fragmentos de ERC fijo de  $k = 1$  (no ERC), 8, 32 y 256, el procedimiento de ERC adaptativa puede mejorar la eficacia en un promedio del 61 %, 26 %, 25 % y 50 % para la fiabilidad de par del 50 %, y del 50 %, 18 %, 29 % y 57 % para la fiabilidad de par del 99 %. La mejora en la eficacia es significativa.

En el sentido más general, se muestra una realización del procedimiento de almacenamiento de codificación adaptativa en la figura 9. Como se muestra en la acción 902 de procedimiento, el sistema de almacenamiento de codificación adaptativa calcula los límites de tamaño de archivo óptimos para un número de fragmentos diferente. Se introduce un archivo de un tamaño de archivo dado a codificar (acción 904 de procedimiento). Se realiza una comprobación en cuanto a si el tamaño de archivo de entrada corresponde a la codificación sin borrado ( $k = 1$ ), como se muestra en la acción 906 de procedimiento. Si el tamaño de archivo de entrada no corresponde a la ERC, el archivo se codifica sin usar la ERC (acción 908 de procedimiento). Si el tamaño de archivo corresponde a un intervalo de tamaño de archivo ERC, el archivo se codifica usando la codificación ERC y el número de fragmentos correspondiente al tamaño de archivo del archivo de entrada, que es el número óptimo de fragmentos para ese archivo de tamaño (acción 910 de procedimiento).

Una aplicación importante del procedimiento de ERC adaptativa descrito en el presente documento es en una copia de seguridad o de restauración P2P. Un par puede hacer una copia de seguridad de los archivos a otros pares en una red y a continuación restaurar estos archivos recuperándolos desde sus pares en la red en el caso de que se pierdan (por ejemplo, en el caso de que se pierdan en un colapso informático). En general, la figura 10 ilustra un diagrama de flujo operacional a modo de ejemplo que muestra cómo puede emplearse la técnica de almacenamiento de codificación adaptativa en un sistema P2P. Debería observarse que cualquier caja e interconexión entre cajas que están representadas por líneas de trazos o discontinuas en la figura 10 representan unas realizaciones alternativas del sistema de almacenamiento de codificación adaptativa descrito en el presente documento, y que cualquiera o todas estas realizaciones alternativas, como se describe a continuación, pueden usarse en combinación con otras realizaciones alternativas que se describen a lo largo de este documento.

En particular, como se ilustra en la figura 10, antes de las operaciones de transferencia de datos, tal como cuando se desea hacer una copia de seguridad de datos a los pares en una red, el servidor 200 o el par 220 codifica 1000 los datos a transferir a los otros pares para su almacenamiento. El sistema de almacenamiento de codificación adaptativa es capaz de operar con cualquiera de un número de códec convencionales, tales como, por ejemplo, MPEG 1/2/4, WMA, WMV, etc. Además, durante el procedimiento 1000 de codificación, el servidor 200 o el par 220 también genera una cabecera de datos como un archivo complementario que contiene la estructura de datos.

Como se ha descrito anteriormente, en una realización, una vez que los datos se codifican 1000, los paquetes de datos codificados se dividen 1005 en un número de unidades de datos de un tamaño fijo. Además, al igual que con los datos codificados, la cabecera de datos y la estructura de datos también se dividen 1005 en un número de unidades de datos del mismo tamaño fijo que el usado para dividir los paquetes de datos codificados. La división 1005 de esta información en unidades de datos de longitud fija permite a los pares preasignar los bloques de memoria antes de las operaciones de transferencia de datos, evitando de este modo las operaciones de asignación de memoria computacionalmente caras durante el procedimiento de transferencia de datos. Además, el uso de unidades de datos más pequeñas permite un control más fino por parte del cliente o del par que almacena los datos sobre la cantidad exacta de ancho de banda gastada por cada par para satisfacer las solicitudes de la unidad de datos del cliente durante las operaciones de transferencia de datos.

Además, para dividir 1005 los datos codificados, la cabecera de datos, y la estructura de datos en unidades de datos más pequeñas, si se emplea la codificación resistente al borrado, se usa una capa adicional de codificación para proporcionar un aumento de la redundancia en un entorno P2P típico donde los pares de servicio son intrínsecamente poco fiables. En particular, como se ha descrito anteriormente, en una realización, si se determina que la codificación resistente al borrado es apropiada para el archivo de datos, las unidades de datos se dividen adicionalmente en un número de bloques de datos y se usa un procedimiento 1010 de codificación resistente al borrado para codificar el archivo.

El uso de tal codificación 1010 garantiza que uno o más de los pares tendrán los bloques de datos necesarios para reconstruir las unidades de datos específicas al tiempo que simplifica la demanda en el cliente para identificar cuál de los pares contiene los datos necesarios. Además, en una realización, las claves de codificación resistente al borrado usadas por cada par 220 de servicio se asignan automáticamente a cada par por el servidor 200. Sin embargo, en otra realización, cada par 220 de servicio simplemente elige una clave de codificación resistente al borrado de manera aleatoria. A continuación, estas claves se recuperan por el cliente 210 cuando el cliente se pone en contacto inicialmente con cada par 220.

Una vez que el archivo de datos se ha codificado 1000 inicialmente, se divide 1005 en unidades de datos, y posiblemente se codifica 1010 para un borrado adicional, las unidades de datos o los bloques de datos resultantes se distribuyen 1015 a continuación a los diversos pares 220. Esta distribución 1015 puede ser deliberada en el sentido de que los bloques o paquetes de los datos codificados simplemente se proporcionan en su totalidad o en parte a un número de pares donde a continuación se almacenan en caché o se almacenan para una transferencia de datos futura cuando se llaman por un cliente que desea recuperar los datos.

Una vez que los datos se han distribuido 1015 a los pares 220 de servicio, entonces el cliente 210 está listo para comenzar las solicitudes de datos a los pares en el caso de que el cliente desee recuperar estos datos desde el almacenamiento. Además, como se ha observado anteriormente, el servidor 200 también puede actuar como un par 220 con el fin de transferir datos al cliente 210.

En este punto, el cliente 210 inicia una sesión de transferencia de datos recuperando en primer lugar una lista de pares 220 de servicio disponibles. Esta lista se recupera directamente desde el servidor 200, desde uno de los pares 220, o usando un procedimiento de tabla hash distribuida (DHT) convencional para identificar posibles pares de servicio. Una vez que el cliente 1010 ha recuperado la lista de pares, el cliente se conecta a cada par 220 de servicio y recupera 1025 una lista de archivos disponibles de cada par. Una vez que el cliente 210 ha recuperado la lista de archivos disponibles de cada par 220, el cliente recupera 1035 la cabecera de datos y la estructura de datos de los datos a transferir de uno o más de los pares solicitando a las unidades de datos correspondientes esa información desde uno o más de los pares a través de una conexión de red entre el cliente y esos pares.

La cabecera de datos contiene, en general, la información global que describe los datos, por ejemplo, el número de canales en los datos, las propiedades y características (velocidad de muestreo de audio, velocidad de tramas / resolución de vídeo) de cada canal, los códec usados, autor / propietario del copyright de los medios, y así sucesivamente. En consecuencia, la recuperación de la cabecera de datos al comienzo de la sesión de transferencia de datos permite que el cliente 220 configure o inicialice 1040 las herramientas necesarias para decodificar 1065 los paquetes recibidos posteriormente antes de recibir esos paquetes durante la sesión de transferencia de datos.

Además, después de recuperar 1035 la estructura de datos de los datos específicos, el cliente analiza esa estructura de datos y calcula 1045 los ID de unidad de datos de las unidades de datos de los datos transferidos que necesitarán solicitarse durante el procedimiento de transferencia de datos. A continuación, el cliente 210 solicita 1050 esas unidades de datos, una por una, desde uno o más de los pares 220.

Por último, una vez que todas las unidades de datos que constituyen un paquete de datos específico se han recuperado de acuerdo con la solicitud 1050 del cliente 210, esos paquetes de datos se reensamblan 1055 en el paquete de datos original. Los paquetes de datos reensamblados se decodifican 1060 a continuación y se pueden restaurar 1065 en el cliente 210.

### 3.0 Almacenamiento P2P: políticas y estrategias de diseño

Además de ajustar el número de fragmentos de ERC basándose en el tamaño de archivo a almacenar en una red P2P, también puede mejorarse la eficacia. Varias realizaciones del sistema de almacenamiento de codificación adaptativa descritas en el presente documento están diseñadas para mejorar la eficacia de almacenamiento mediante el empleo de ciertas estrategias como se describe a continuación. Estas estrategias pueden emplearse junto con el sistema de almacenamiento de codificación adaptativa o emplearse en cualquier red P2P.

#### 3.1 Coste de almacenamiento P2P.

En esta sección, el almacenamiento de un archivo en una red P2P se compara con el almacenamiento del archivo directamente en un servidor fiable de "seis nueves". Se observa que la solución P2P reduce el ancho de banda y el coste del servidor, pero requiere que el par gaste más ancho de banda para distribuir el archivo en el almacenamiento P2P. El uso general del ancho de banda de la red aumenta en la solución P2P. El aumento en el

ancho de banda de subida del cliente puede considerarse un coste del sistema de almacenamiento P2P. Este coste para las diferentes fiabilidades de pares y tamaños de archivo se tabula en la Tabla 1.

Tabla 1 Coste del aumento del uso de ancho de banda en P2P.

Fiabilidad	Tamaño de archivo				
	10KB	100K B	1MB	10MB	100M B
10 %	332,9	79,1	29,5	16,5	12,5
50 %	51,0	12,11	4,34	2,23	1,56
99 %	9,4	1,87	0,65	0,22	0,09

- 5 Se observa que el coste del uso de almacenamiento P2P es pequeño si la fiabilidad de par es alta y el tamaño de archivo es grande. Por ejemplo, almacenar 100 MB de archivos con una fiabilidad del 99 % solo incurre en un coste del 9 %. Sin embargo, cuando la fiabilidad de par es baja y el tamaño de archivo es pequeño, el coste puede ser significativo.

### 3.2 Políticas de almacenamiento P2P.

- 10 De la Tabla 1, puede obtenerse la siguiente política de uso de la nube de almacenamiento P2P:

a) Deberían usarse pares no fiables para almacenar archivos grandes y usar pares fiables para almacenar archivos pequeños. El coste para el sistema P2P será menor si se asignan archivos grandes a pares no fiables, y se asignan archivos más pequeños a pares fiables.

- 15 b) Deberían usarse pares no fiables para almacenar archivos estáticos y usar pares fiables para almacenar archivos dinámicos. Se llama a los archivos que no cambian como estáticos y se llaman a los archivos que cambian constantemente como dinámicos. Múltiples pequeños archivos estáticos pueden agruparse en un gran archivo estático y almacenarse en la nube de almacenamiento P2P. La misma estrategia no es efectiva para los archivos dinámicos, ya que el cambio de un solo archivo requiere que se actualice todo el archivo combinado.

20 Un corolario de esta política es que si se usa la red P2P para almacenar el estado de una aplicación, la información de estado de par, etc., debería desviarse la información a los pares más fiables de la red. Si se restringe que el archivo que contiene el estado de la aplicación solo se coloque en pares de alta fiabilidad (en esencia, los pares de alta fiabilidad formarán una subred que constituirá los núcleos de la red P2P extendida), puede reducirse enormemente esta velocidad de replicación y el coste de actualizar el archivo de estado y mejorar la eficacia.

- 25 c) Debería permitirse que los pares no fiables distribuyan menos, y debería permitirse que los pares fiables distribuyan más.

d) Debería asignarse un mayor coste de distribución a los archivos más pequeños, y debería asignarse un menor coste de distribución a los archivos más grandes.

30 Las políticas c) y d) son para aplicaciones de copia de seguridad y de recuperación P2P, donde un par puede distribuir el contenido en la nube de almacenamiento P2P, y almacenar el contenido para otros pares. Una red de almacenamiento P2P balanceada debería permitir a cada par balancear su contribución y beneficio. En trabajos anteriores se ha señalado que el ancho de banda es el recurso principal en la aplicación de almacenamiento P2P. Permitir que la contribución del par sea la cantidad de fragmentos codificados que recibe y almacena para los otros pares. Permitir que el beneficio del par sea la cantidad de contenido que distribuye en la nube P2P. Teniendo en cuenta que la baja fiabilidad conduce a un almacenamiento más redundante de los datos, debería castigarse a los pares poco fiables de tal manera que se les permita distribuir menos, y recompensar a los pares fiables de tal manera que se les permita distribuir más. Dicha política puede tener un beneficio positivo en la economía P2P, ya que puede alentar al usuario a mantener la aplicación P2P en línea, mejorando de este modo la fiabilidad general de la red P2P y reduciendo la relación de replicación necesaria.

40 También puede castigarse la distribución de un pequeño archivo asignándole con un alto coste de distribución, requiriendo que el par contribuya más proporcionalmente; y recompensarse la distribución de archivos grandes asignándola con un bajo coste de distribución, permitiendo que los pares contribuyan menos proporcionalmente. Como corolario, las aplicaciones de copia de seguridad P2P deberían diseñarse para minimizar la frecuencia de las copias de seguridad. En lugar de actualizar inmediatamente el archivo justo después de su cambio, puede considerarse el agrupar diversos cambios en un archivo grande y actualizarlo solo una vez, por ejemplo, cada medianoche, en la nube de almacenamiento P2P.

50 Una realización del sistema y el procedimiento de almacenamiento de codificación adaptativa que está diseñado alrededor de las políticas anteriores se muestra en la figura 11. Como se muestra en la acción 1102 de procedimiento, se determina la fiabilidad de cada par en la red distribuida o P2P. Se introduce un archivo a distribuir o almacenarse (acción 1104 de procedimiento). El tamaño del archivo se evalúa (acción 1106 de procedimiento) y

se asigna un coste de distribución al archivo basándose en el ancho de banda de almacenamiento esperado en la ecuación (7) (acción 1108 de procedimiento). Si el archivo es un archivo grande, puede asignarse un coste de distribución más alto. Si el archivo es pequeño, puede asignarse al archivo un menor coste de distribución. En función del tamaño del archivo, el sistema de almacenamiento de codificación adaptativa elegirá unos pares con la fiabilidad adecuada para almacenar el archivo (acción 1110 de procedimiento). Es decir, los pares cuya fiabilidad está por debajo de un umbral dado se usan para almacenar y distribuir el archivo grande, y los pares cuya fiabilidad está por encima de un umbral dado se usan para almacenar y distribuir el archivo pequeño.

Otra realización del sistema y el procedimiento de almacenamiento de codificación adaptativa que está diseñado alrededor de las políticas anteriores se muestra en la figura 12. Como se muestra en la acción 1202 de procedimiento, se determina la fiabilidad de cada par en la red distribuida o P2P. Se introduce un archivo a distribuir o almacenarse (acción 1204 de procedimiento). El archivo se compara con el mismo archivo que se ha almacenado anteriormente para determinar si el archivo es estático o dinámico (procedimiento 1206 de acción). La primera vez que se deposita el archivo, se supone que el archivo es dinámico. Si se observan cambios frecuentes en el archivo, el archivo permanece designado como dinámico. Si se observa que el archivo no cambia durante un período de tiempo prolongado, el archivo se designa como estático. Los archivos dinámicos se almacenan en pares altamente fiables (acción 1210 de procedimiento). (Por lo tanto, al principio, los archivos se almacenarán en servidores o en pares altamente fiables). Una vez que se observa que los archivos no cambian y se vuelven estáticos, estos archivos estáticos se redistribuirán y se almacenarán en pares de menor fiabilidad.

Debería observarse que las realizaciones mostradas en las figuras 11 y 12 pueden usarse solas o en combinación con el fin de aumentar la eficacia y la fiabilidad generales de una red distribuida o entre pares.

### 3.3 Almacenamiento P2P con soporte de componentes de servidor.

Si un componente de servidor se usa como complemento de la red P2P, puede usarse el almacenamiento P2P de archivos de gran tamaño y estáticos, y usar el servidor para archivos pequeños y dinámicos. Ya que son los archivos grandes los que consumen la mayor parte del recurso del servidor, el almacenamiento P2P complementa bien al servidor.

Como se muestra en la figura 13, una realización del sistema y el procedimiento de almacenamiento de codificación adaptativa emplean la copia de seguridad P2P con soporte de servidor. Como se muestra en la figura 13, los archivos dinámicos en la red se hacen copia de seguridad en el servidor (procedimiento 1302 de acción). Entonces el cliente y/o el servidor pueden detectar automáticamente aquellos archivos dinámicos que ya no se modifican y se convierten en archivos estáticos (procedimiento 1304, 1306 de acción). Estos archivos estáticos detectados pueden agruparse a continuación en un archivo grande, como se muestra en la acción 1308 de procedimiento, y distribuirse con la ERC en la nube de almacenamiento P2P (acción 1310 de procedimiento). Esto aumenta efectivamente el tamaño de archivo almacenado en la nube P2P. Combinado con la ERC de una gran cantidad de fragmentos, esto puede mejorar la eficacia.

La realización mostrada en la figura 13 puede usarse sola o en combinación con las realizaciones mostradas en las figuras 11 y 12 para aumentar la eficacia y fiabilidad generales de una red distribuida o entre pares. También debería observarse que esta realización puede usarse tanto con codificación resistente al borrado como sin la misma.

Debería observarse que cualquiera o todas las realizaciones alternativas mencionadas anteriormente pueden usarse en cualquier combinación deseada para formar realizaciones híbridas adicionales. Aunque el objeto se ha descrito en un lenguaje específico para las funciones estructurales y/o las acciones metodológicas, debe entenderse que el objeto definido en las reivindicaciones adjuntas no está necesariamente limitado a las funciones o acciones específicas descritas anteriormente. Por el contrario, las funciones y las acciones específicas descritas anteriormente se desvelan como formas de ejemplo para implementar las reivindicaciones.

### Lista de ejemplos

1. Un procedimiento implementado por ordenador para codificar archivos a almacenar en una red distribuida, que comprende las acciones de procedimiento de:

- calcular intervalos de tamaño de archivo óptimos correspondientes para diferentes números de fragmentos de codificación de recuperación de errores (ERC), en el que cada número de fragmentos es el número óptimo de fragmentos para un intervalo correspondiente de tamaños de archivos (902);
- introducir un archivo de un tamaño de archivo dado (904);
- si el tamaño de archivo es más pequeño que el intervalo de los tamaños de archivo para el menor número de fragmentos de ERC de dos, codificar el archivo sin usar la codificación resistente al borrado (906, 908);
- si el tamaño de archivo del archivo de entrada corresponde a un intervalo de tamaños de archivo, codificar el archivo usando la codificación resistente al borrado y el número óptimo de fragmentos correspondientes al intervalo de tamaño de archivo del archivo de entrada (906, 910).

2. El procedimiento implementado por ordenador del Ejemplo 1, que comprende además la acción de procedimiento de enviar el archivo codificado a uno o más pares en una red distribuida.

3. El procedimiento implementado por ordenador del Ejemplo 1, que comprende además calcular el número de pares que se almacenarán en el archivo codificado de acuerdo con la fiabilidad de par y la fiabilidad deseada del contenido de archivo.
- 5 4. El procedimiento implementado por ordenador del Ejemplo 1, en el que el cálculo de intervalos de tamaño de archivo óptimos correspondientes a diferentes números de fragmentos de codificación de recuperación de errores (ERC) comprende las acciones de procedimiento de:
- determinar un límite entre diferentes números de fragmentos como el tamaño de archivo óptimo adecuado para un número de fragmentos de ERC específico.
- 10 5. El procedimiento implementado por ordenador del Ejemplo 1, que comprende además las acciones de procedimiento de:
- obtener un conjunto de fragmentos de archivos del archivo codificado igual o mayor que un número de fragmentos en los que el archivo se divide para la codificación; y  
decodificar los fragmentos de archivo codificados con una decodificación resistente al borrado si el archivo se ha codificado resistente al borrado para obtener una versión decodificada del archivo codificado; y  
15 decodificar los fragmentos codificados sin una decodificación resistente al borrado si el archivo no se ha codificado resistente al borrado para obtener una versión decodificada del archivo codificado.
6. El procedimiento implementado por ordenador del Ejemplo 1, en el que la codificación resistente al borrado usada en codificar el archivo es la codificación Reed Solomon.
7. El procedimiento implementado por ordenador del Ejemplo 6, en el que:
- 20 si el tamaño de archivo es menor que aproximadamente 10 KB, no se usa la codificación resistente al borrado;  
si el tamaño de archivo es de aproximadamente 10 KB a 33 KB, el número óptimo de fragmentos es dos;  
si el tamaño de archivo es de aproximadamente 33 KB a 100 KB, el número óptimo de fragmentos es cuatro;  
si el tamaño de archivo es de aproximadamente 100 KB a 310 KB, el número óptimo de fragmentos es ocho;  
25 si el tamaño de archivo es de aproximadamente 310 KB a 950 KB, el número óptimo de fragmentos es dieciséis;  
si el tamaño de archivo es de aproximadamente 950 KB a 2.9 MB, el número óptimo de fragmentos es treinta y dos;  
si el tamaño de archivo es de aproximadamente 2,9 MB a 8,9 MB, el número óptimo de fragmentos es sesenta y cuatro;  
si el tamaño de archivo es de aproximadamente 8,9 MB a 26 MB, el número óptimo de fragmentos es ciento  
30 veintiocho; y  
si el tamaño de archivo es mayor que aproximadamente 26 MB, el número óptimo de fragmentos es doscientos cincuenta y seis.
8. Un medio legible por ordenador que tiene instrucciones ejecutables por ordenador para realizar el procedimiento descrito en el Ejemplo 1.
- 35 9. Un sistema para mejorar la fiabilidad y eficacia de almacenamiento de una red entre pares, que comprende:
- un dispositivo informático de fin general;  
un programa informático que comprende unos módulos de programa ejecutables por el dispositivo informático de fin general, en el que el dispositivo informático se dirige por los módulos de programa del programa informático para,  
40 determinar el número óptimo de fragmentos para codificar un archivo de un tamaño de archivo dado con una codificación (1010) resistente al borrado;  
si el número óptimo de fragmentos para codificar el archivo con una codificación resistente al borrado es uno, no se codifica el archivo con la codificación (1010) resistente al borrado; y  
45 si el número óptimo de fragmentos es mayor que uno, codificar el archivo dividiendo el archivo en el número óptimo de fragmentos y codificar el archivo con la codificación (1010) resistente al borrado.
10. El sistema del Ejemplo 9, que comprende además un módulo de programa para calcular el número de pares en los que se almacenarán los fragmentos de archivo codificados de acuerdo con la fiabilidad de par y la fiabilidad deseada del contenido de archivo.
- 50 11. El sistema del Ejemplo 10, que comprende además un módulo de programa para distribuir el archivo codificado con una codificación resistente al borrado a uno o más pares en una red.
12. El sistema del Ejemplo 11, en el que el módulo de programa para distribuir el archivo comprende unos submódulos para:
- determinar la fiabilidad de cada par en la red distribuida;  
determinar el tamaño del archivo;

y usar uno o más pares con una fiabilidad adecuada determinada por el tamaño de archivo para distribuir el archivo.

13. El sistema del Ejemplo 12, en el que el módulo de programa para distribuir el archivo comprende unos submódulos para:

- 5 si el archivo es grande, usar unos pares cuya fiabilidad esté por debajo de un umbral dado para distribuir el archivo grande; y  
si el archivo no es grande, usar unos pares cuya fiabilidad esté por encima de un umbral dado para distribuir el archivo.

10 14. El sistema del Ejemplo 11, en el que el módulo de programa para distribuir el archivo comprende unos submódulos para:

- determinar la fiabilidad de cada par en la red distribuida;  
determine si el archivo es estático;  
si el archivo es estático, usar unos pares cuya fiabilidad esté por debajo de un umbral determinado para distribuir el archivo; y  
15 si el archivo no es estático, usar unos pares cuya fiabilidad esté por encima de un umbral determinado para distribuir el archivo.

15. El sistema del Ejemplo 11, en el que el módulo de programa para distribuir el archivo comprende unos submódulos para:

- 20 determinar la fiabilidad de cada par en la red distribuida;  
monitorizar los cambios del archivo;  
distribuir en primer lugar el archivo a los pares más fiables;  
si se observa que el archivo no cambia, redistribuir el archivo a los pares menos fiables.

16. El sistema del Ejemplo 9, que comprende además un módulo de programa para mejorar la eficacia de la red distribuida que usa un servidor, comprendiendo además unos submódulos para:

- 25 hacer una copia de seguridad de todos los archivos dinámicos en la red distribuida al servidor;  
tener periódicamente los pares y el servidor en la comprobación de red distribuida para ver si han cambiado los archivos dinámicos copiados en el servidor;  
si los archivos dinámicos no han cambiado, designar estos archivos como estáticos y agruparlos en un gran archivo; y  
30 distribuir el gran archivo con una codificación resistente al borrado en la red distribuida.

17. El sistema del Ejemplo 9, en el que el módulo de programa para determinar el número óptimo de fragmentos para codificar un archivo de un tamaño dado con una codificación resistente al borrado comprende unos submódulos para:

- 35 determinar un intervalo de tamaño de archivo óptimo para cada posible número de fragmentos de codificación resistente al borrado, en el que cada número de fragmentos es el número óptimo de fragmentos para el intervalo correspondiente;  
determinar en qué intervalo de tamaño de archivo se encuentra el tamaño de un archivo de entrada; y  
utilizar como el número óptimo de fragmentos el número correspondiente al intervalo de tamaño de archivo en el que se encuentra el tamaño de archivo de entrada.

40 18. Un procedimiento implementado por ordenador para decodificar un archivo codificado almacenado en una red distribuida, que comprende las acciones de procedimiento de:

- recuperar un conjunto de fragmentos de un archivo codificado igual o mayor que el número de fragmentos que se han usado para codificar el archivo, en el que el archivo se ha codificado resistente al borrado con un número óptimo de fragmentos para un tamaño de archivo dado; y  
45 decodificar los fragmentos codificados con una decodificación resistente al borrado para obtener una versión decodificada del archivo codificado.

19. El procedimiento implementado por ordenador del Ejemplo 18, en el que al menos algunos de los fragmentos codificados se recuperan a partir de un medio de almacenamiento de uno o más pares en la red distribuida.

50 20. El procedimiento implementado por ordenador del Ejemplo 18, en el que al menos algunos de los fragmentos codificados se recuperan a partir de un medio de almacenamiento de un servidor en la red distribuida.

**REIVINDICACIONES**

1. Un procedimiento implementado por ordenador para codificar archivos a almacenar en una red entre pares distribuida, que comprende las acciones de procedimiento de:

5 introducir (904) un archivo de un tamaño de archivo dado;  
 si el tamaño de archivo es menor que un intervalo de tamaños de archivo para el número más pequeño de fragmentos de dos de codificación resistente al borrado, ERC, no usar (906, 908) la codificación resistente al borrado para el archivo, en el que un número de fragmentos de ERC corresponde al número de fragmentos usados para la codificación ERC del archivo; y  
 10 si el tamaño de archivo del archivo de entrada corresponde a un intervalo de tamaño de archivo de codificación resistente al borrado, codificar (906, 910) el archivo usando una codificación resistente al borrado y un número óptimo de fragmentos correspondiente a un intervalo de tamaño del archivo de entrada, en el que para cada uno de los múltiples tamaños de archivo diferentes, se predetermina un número óptimo de fragmentos correspondientes para el tamaño de archivo respectivo basándose en un ancho de banda de red total necesario para almacenar el archivo en un número necesario de pares en la red distribuida, en el que los diferentes intervalos de tamaño de archivo corresponden a diferentes números de fragmentos de codificación resistente al borrado, y en el que cada número de fragmentos es el número óptimo de fragmentos para un intervalo correspondiente de tamaños de archivo.

2. El procedimiento implementado por ordenador de la reivindicación 1, que comprende además la acción de procedimiento de enviar el archivo codificado a uno o más pares (210, 220) en una red distribuida.

20 3. El procedimiento implementado por ordenador de la reivindicación 1, que comprende además calcular el número de pares (210, 220),  $n$ , que se almacenarán en el archivo codificado de acuerdo con la fiabilidad de par,  $p$ , y la fiabilidad deseada,  $p_0$  del contenido de archivo por  $n = \log(1 - p_0) / \log(1 - p)$ , en el que la fiabilidad de par corresponde al tiempo de actividad y al tiempo de inactividad de un par.

25 4. El procedimiento implementado por ordenador de la reivindicación 1, en el que los diferentes intervalos de tamaño de archivo que corresponden a diferentes números de fragmentos de codificación resistente al borrado, ERC, se determinan:

calculando y trazando el ancho de banda total necesario para diferentes tamaños de archivo y número de fragmentos de ERC, y determinando un límite entre diferentes números de fragmentos como el tamaño de archivo óptimo adecuado para un número de fragmentos de ERC específico en la representación gráfica.

30 5. El procedimiento implementado por ordenador de la reivindicación 1, en el que la codificación resistente al borrado usada para codificar el archivo es la codificación Reed Solomon y  
 si el tamaño de archivo es menor que aproximadamente 10 KB, no se usa la codificación resistente al borrado;  
 si el tamaño de archivo es de aproximadamente 10 KB a 33 KB, el número óptimo de fragmentos es dos;  
 si el tamaño de archivo es de aproximadamente 33 KB a 100 KB, el número óptimo de fragmentos es cuatro;  
 35 si el tamaño de archivo es de aproximadamente 100 KB a 310 KB, el número óptimo de fragmentos es ocho;  
 si el tamaño de archivo es de aproximadamente 310 KB a 950 KB, el número óptimo de fragmentos es dieciséis;  
 si el tamaño de archivo es de aproximadamente 950 KB a 2,9 MB, el número óptimo de fragmentos es treinta y dos;  
 si el tamaño de archivo es de aproximadamente 2,9 MB a 8,9 MB, el número óptimo de fragmentos es sesenta y cuatro;  
 40 si el tamaño de archivo es de aproximadamente 8,9 MB a 26 MB, el número óptimo de fragmentos es ciento veintiocho; y  
 si el tamaño de archivo es mayor que aproximadamente 26 MB, el número óptimo de fragmentos es doscientos cincuenta y seis.

45 6. Un medio legible por ordenador que tiene unas instrucciones ejecutables por ordenador que, cuando se ejecutan por un dispositivo informático, hacen que dicho dispositivo informático realice el procedimiento de acuerdo con una de las reivindicaciones 1 a 5.

7. Un sistema para mejorar la fiabilidad y la eficacia del almacenamiento de una red entre pares, que comprende:

un dispositivo informático de fin general;  
 un programa informático que comprende unos módulos de programa ejecutables por el dispositivo informático de fin general, en el que el dispositivo informático se dirige por los módulos de programa del programa informático para:  
 50

determinar el número óptimo de fragmentos para codificar un archivo de un tamaño de archivo dado con una codificación (1010) resistente al borrado, en el que para cada uno de los múltiples tamaños de archivo diferentes, se predetermina un número óptimo correspondiente de fragmentos para el tamaño de archivo respectivo basándose en un ancho de banda de red total necesario para almacenar el archivo en un número necesario de pares en la red distribuida, en el que los diferentes intervalos de tamaño de archivo corresponden a diferentes números de fragmentos, en el que cada número de fragmentos es el número  
 55

- 5            óptimo de fragmentos para un intervalo correspondiente de tamaños de archivo;  
              si el número óptimo de fragmentos para codificar el archivo con una codificación resistente al borrado es uno,  
              no se codifica el archivo con una codificación (1010) resistente al borrado; y  
              si el número óptimo de fragmentos es mayor que uno, se codifica el archivo usando el número óptimo de  
              fragmentos y usando la codificación (1010) resistente al borrado.
- 10            8. El sistema de la reivindicación 7, que comprende además un módulo de programa para calcular el número de  
              pares en los que se almacenarán los fragmentos de archivo codificados de acuerdo con la fiabilidad de par y la  
              fiabilidad deseada del contenido de archivo y  
              que comprende además un módulo de programa para distribuir el archivo codificado con una codificación resistente  
              al borrado a uno o más pares en una red, en el que la fiabilidad de par se corresponde con el tiempo de actividad y  
              el tiempo de inactividad de un par.
- 15            9. El sistema de la reivindicación 8, en el que el módulo de programa para distribuir el archivo comprende unos  
              submódulos para:  
              determinar (1102) la fiabilidad de cada par en la red distribuida;  
              determinar (1106) el tamaño del archivo;  
              y usar (1110) uno o más pares con una fiabilidad determinada por el tamaño de archivo para distribuir el archivo.
- 20            10. El sistema de la reivindicación 8, en el que el módulo de programa para distribuir el archivo comprende unos  
              submódulos para:  
              determinar (1202) la fiabilidad de cada par en la red distribuida;  
              determinar (1206) si el archivo es estático;  
              si el archivo es estático, usar (1208) unos pares cuya fiabilidad esté por debajo de un umbral dado para distribuir  
              el archivo; y  
              si el archivo no es estático, usar (1210) unos pares cuya fiabilidad esté por encima de un umbral dado para  
              distribuir el archivo o  
25            en el que el módulo de programa para distribuir el archivo comprende unos submódulos para:  
              determinar la fiabilidad de cada par en la red distribuida;  
              monitorizar los cambios del archivo;  
              distribuir en primer lugar el archivo a los pares más fiables;  
              si se observa que el archivo no cambia, redistribuir el archivo a los pares menos fiables.
- 30            11. El sistema de la reivindicación 7, que comprende además un módulo de programa para mejorar la eficacia de la  
              red distribuida usando un servidor, comprendiendo además unos submódulos para:  
              hacer una copia de seguridad (1302) de todos los archivos dinámicos en la red distribuida al servidor;  
              tener periódicamente los pares y el servidor en la comprobación de red distribuida (1304) para ver si han  
              cambiado los archivos dinámicos copiados en el servidor;  
35            si los archivos dinámicos no han cambiado, designar (1306) estos archivos como estáticos y agruparlos en un  
              solo archivo; y  
              distribuir (1310) el archivo con una codificación resistente al borrado en la red distribuida.
- 40            12. El sistema de la reivindicación 7, en el que el módulo de programa para determinar el número óptimo de  
              fragmentos para codificar un archivo de un tamaño dado con una codificación resistente al borrado comprende unos  
              submódulos para:  
              determinar un intervalo de tamaño de archivo para cada posible número de fragmentos de codificación resistente  
              al borrado, en el que cada número de fragmentos es el número óptimo de fragmentos para el intervalo  
              correspondiente;  
              determinar en qué intervalo de tamaño de archivo se encuentra el tamaño de un archivo de entrada; y  
45            usar como el número óptimo de fragmentos el número correspondiente al intervalo de tamaño de archivo en el  
              que se encuentra el tamaño de archivo de entrada.

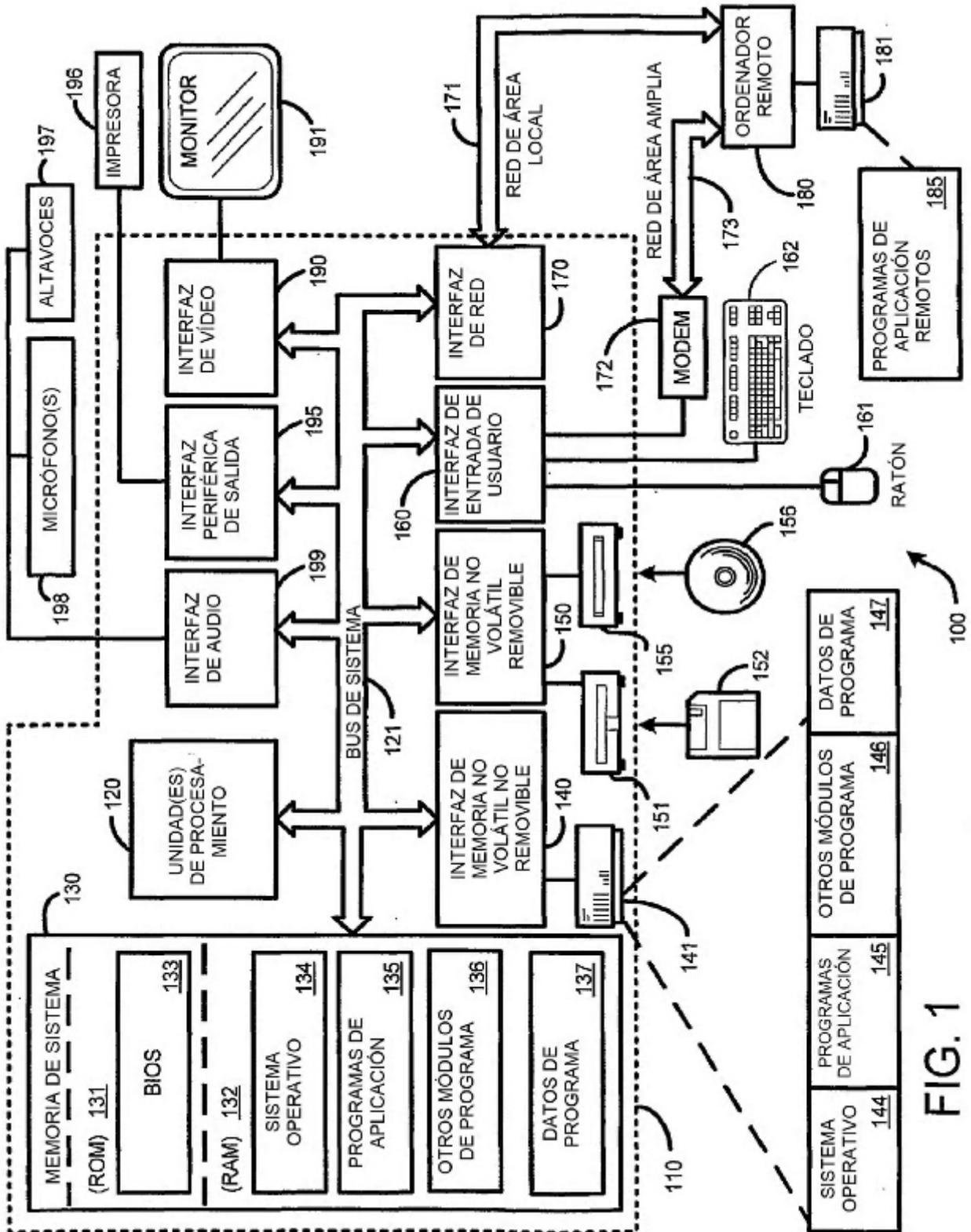
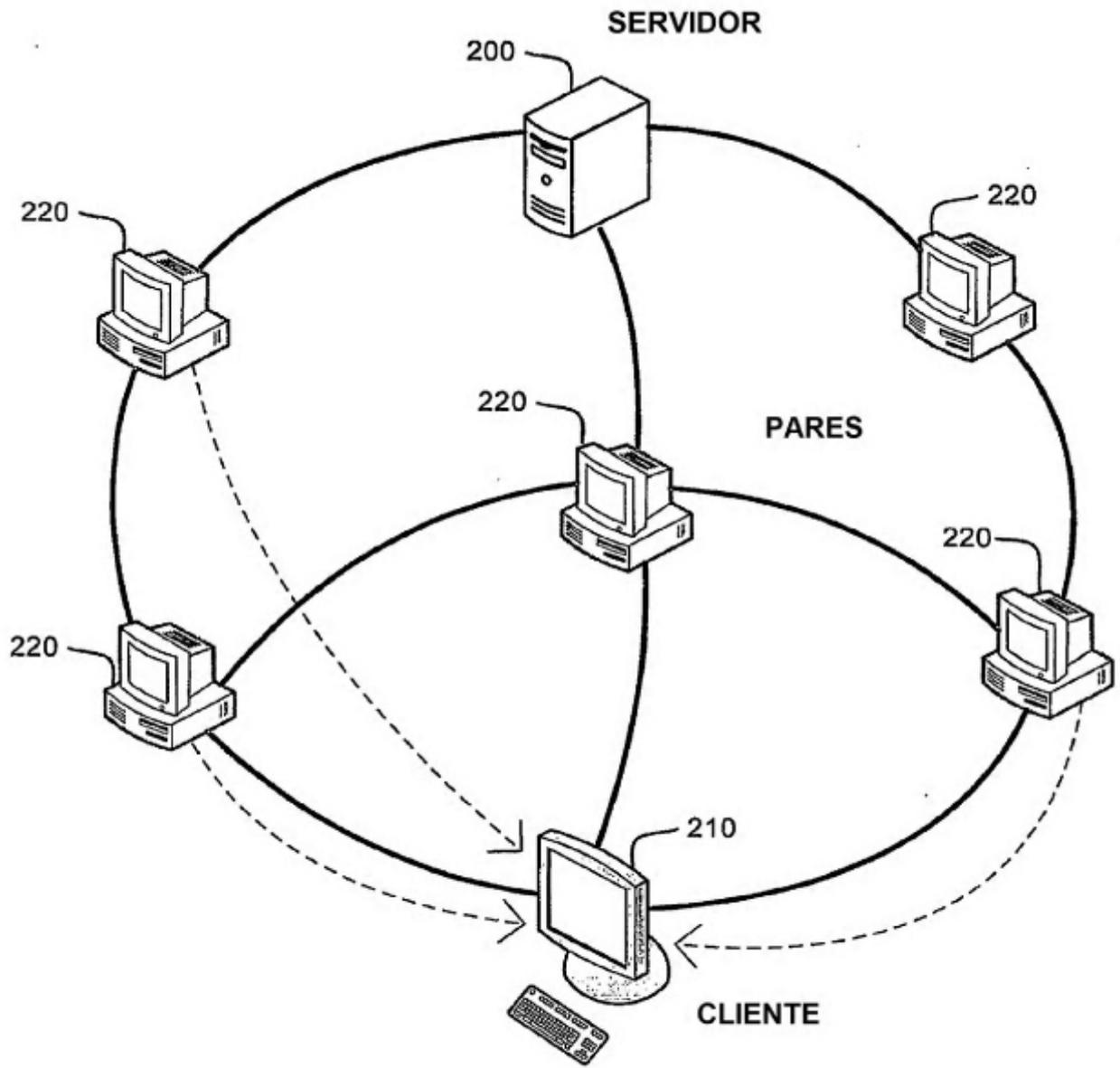
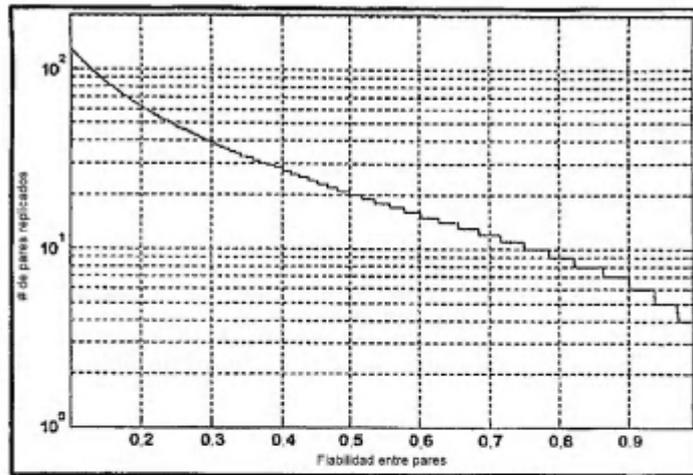


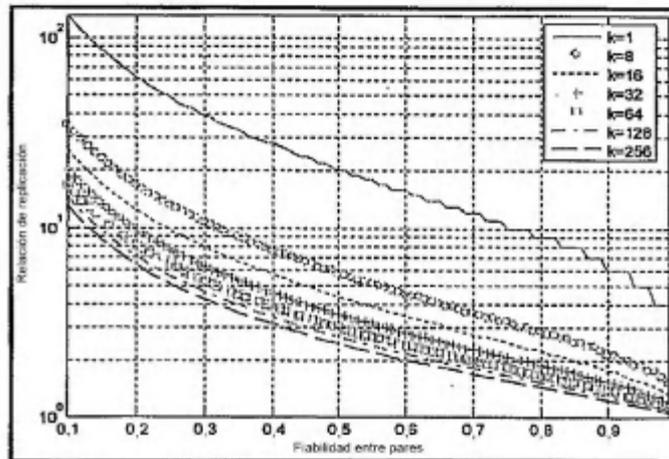
FIG. 1



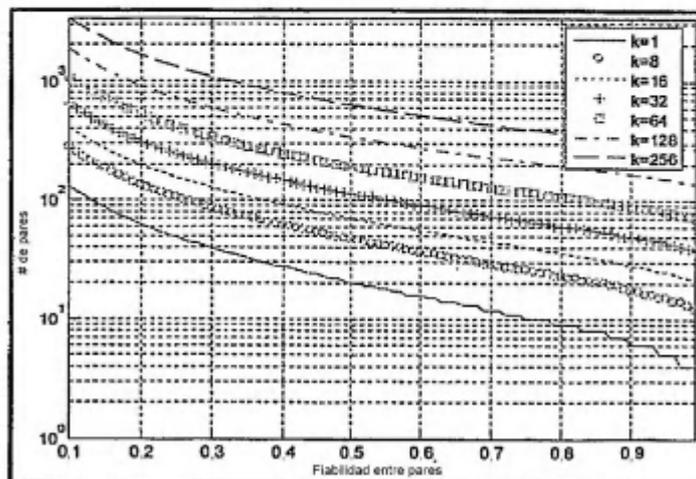
**FIG. 2**



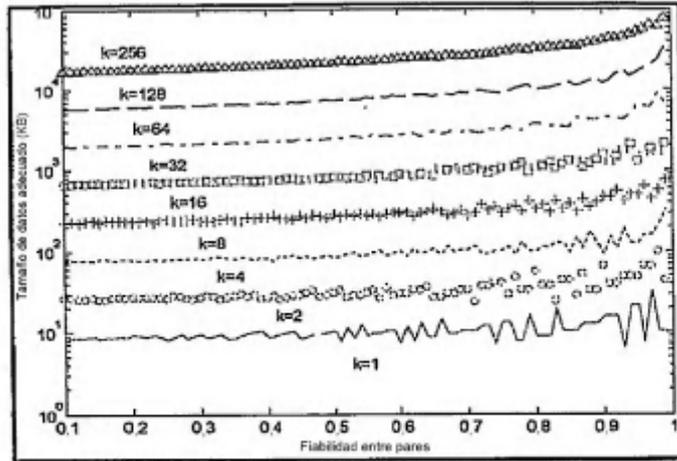
**FIG. 3**



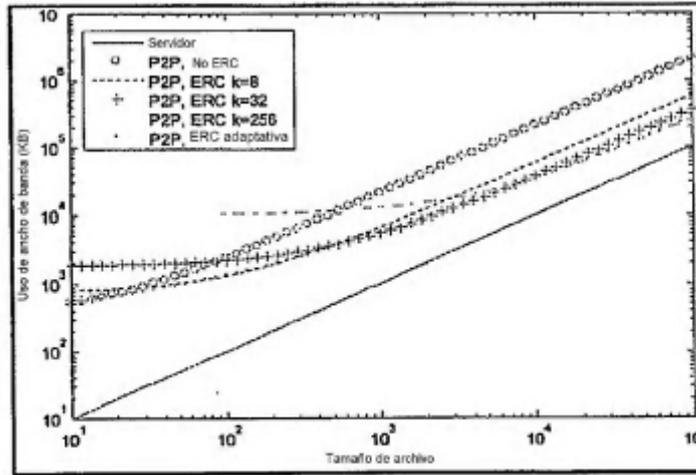
**FIG. 4**



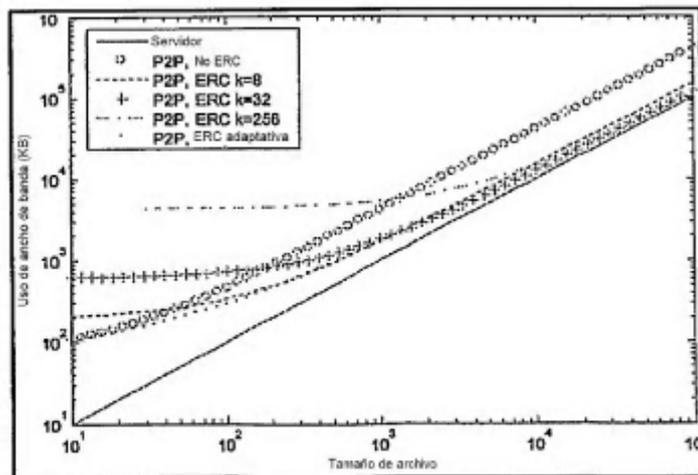
**FIG. 5**



**FIG. 6**

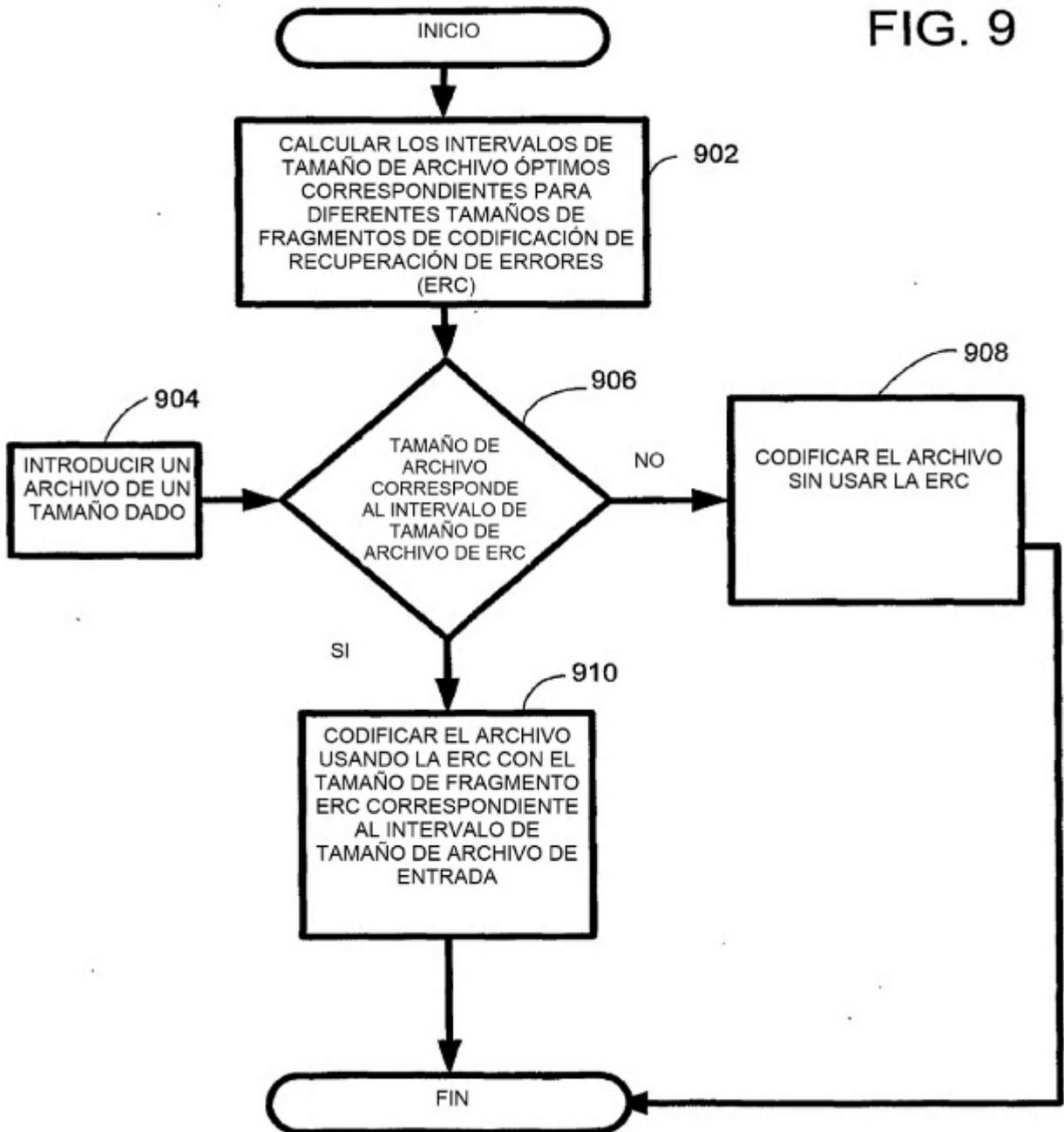


**FIG. 7**



**FIG. 8**

FIG. 9



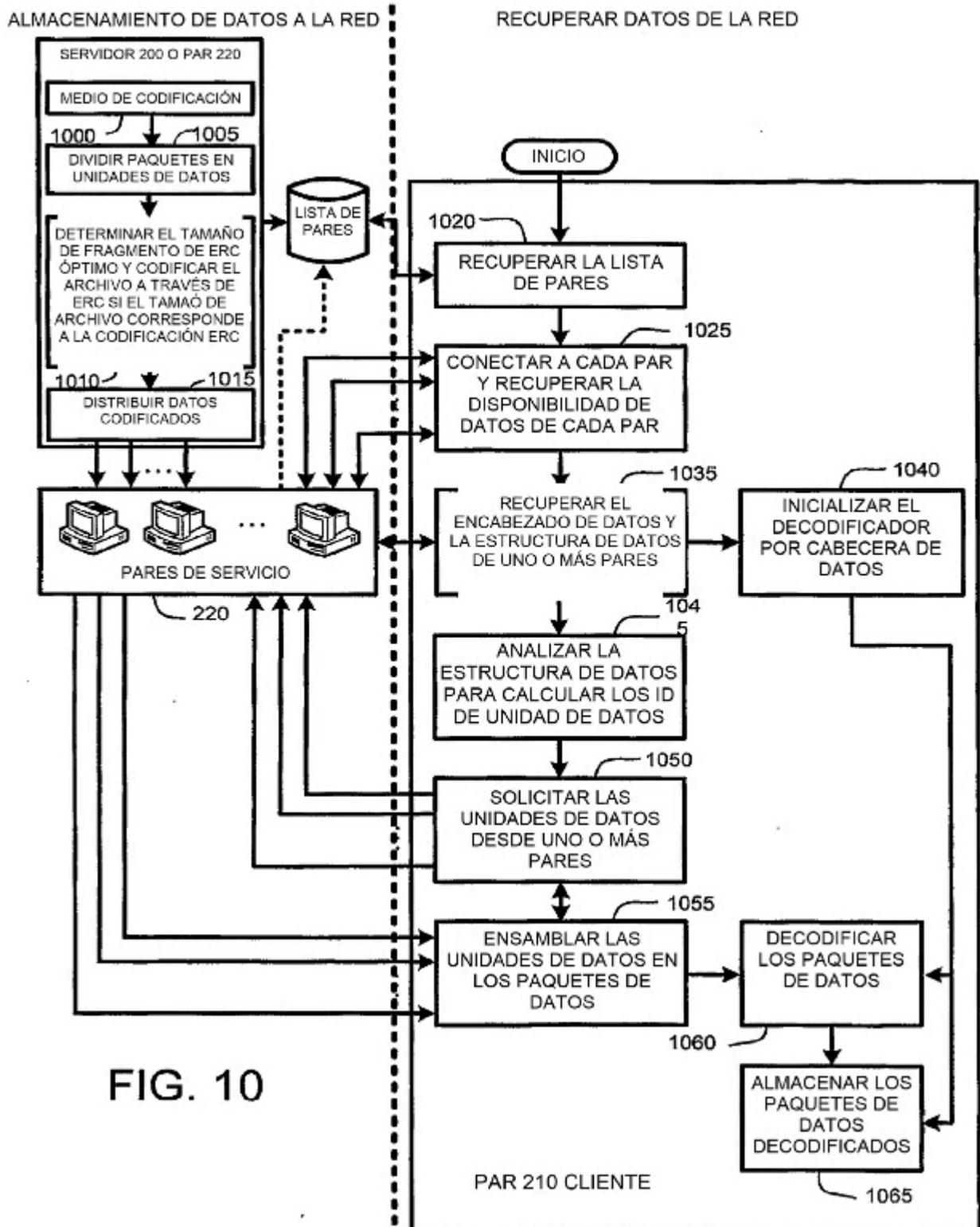


FIG. 10

FIG. 11

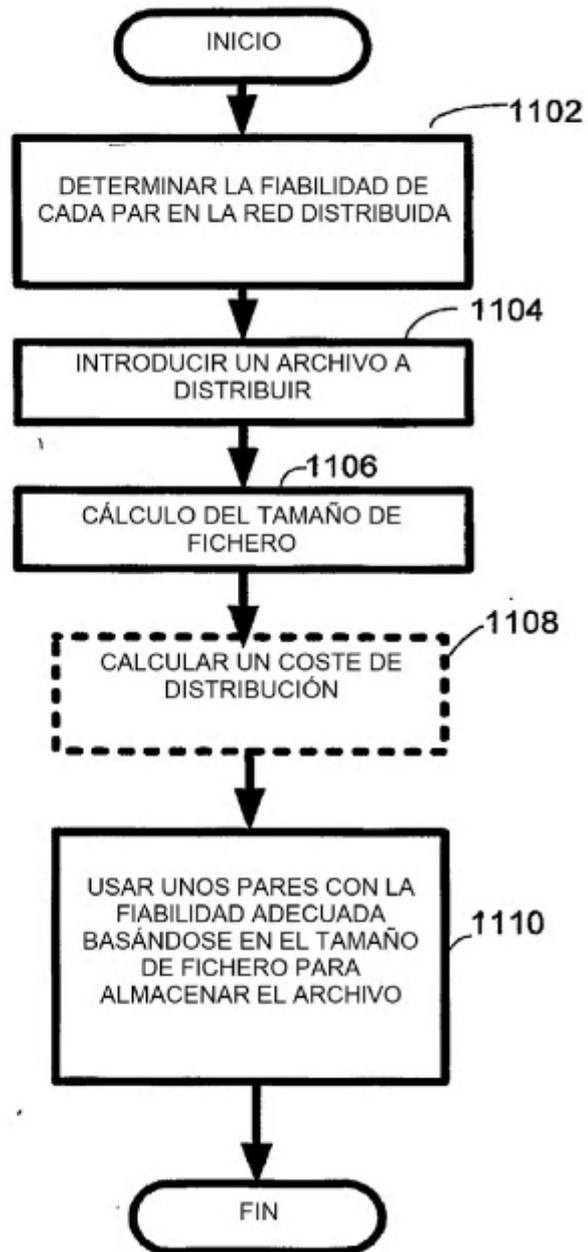
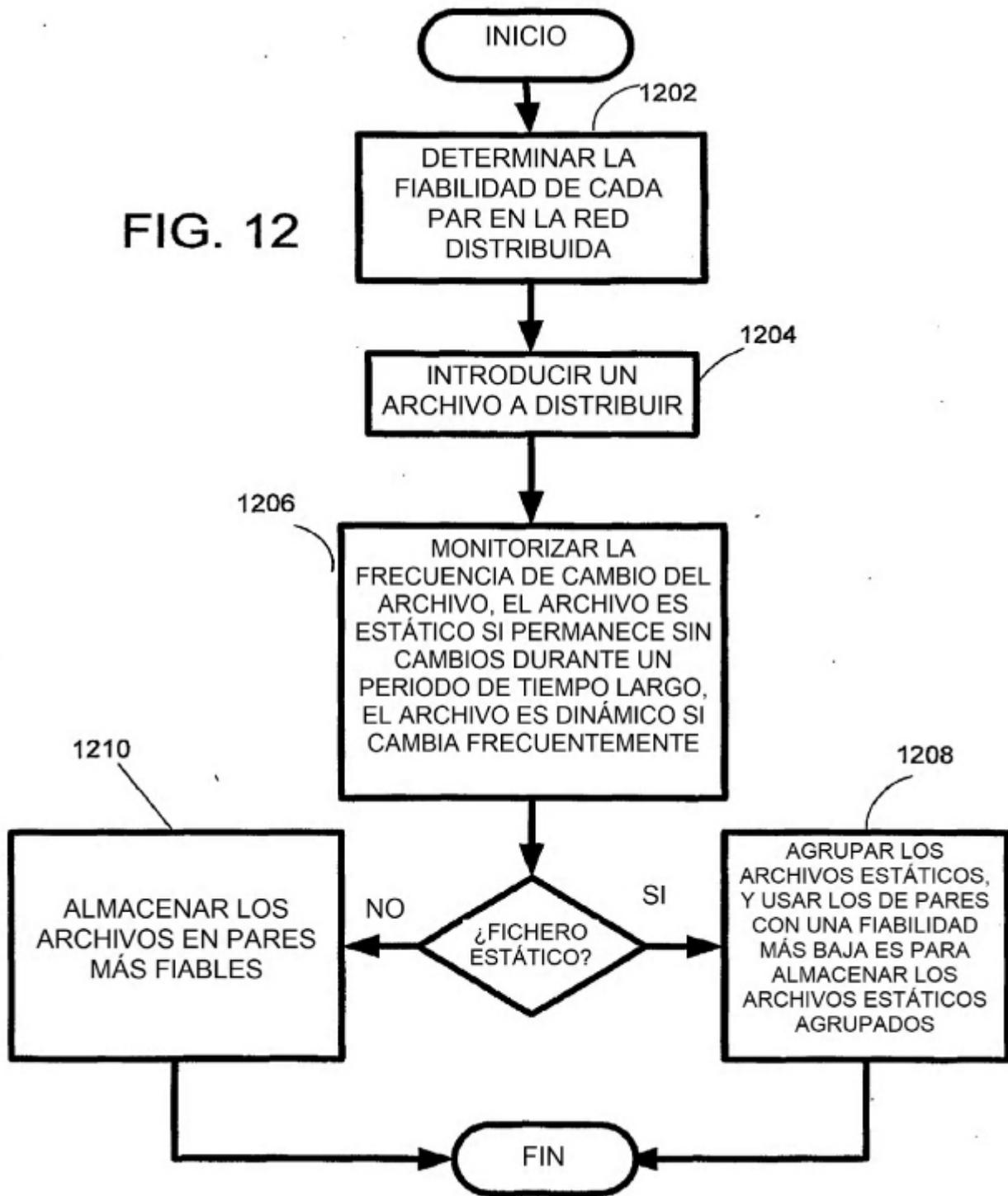


FIG. 12



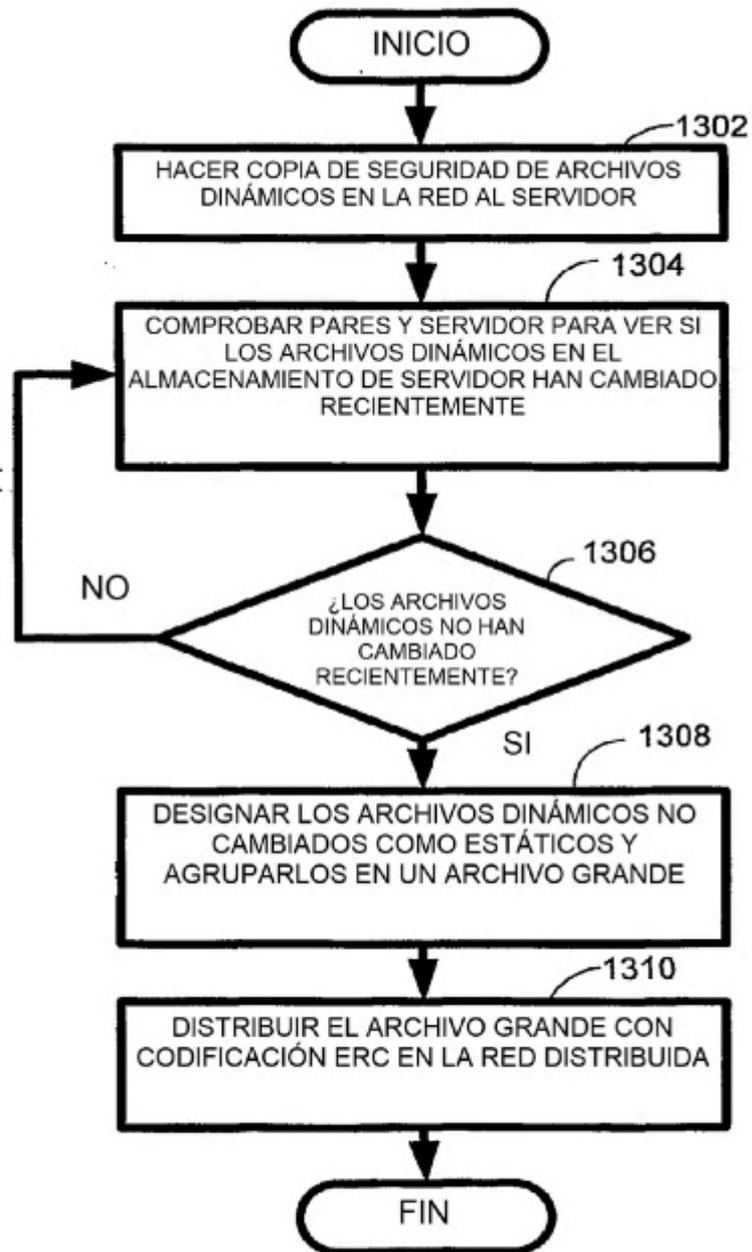


FIG. 13