

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 635 646**

51 Int. Cl.:

G06F 9/50 (2006.01)

G06F 17/30 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **18.03.2003 E 03006146 (9)**

97 Fecha y número de publicación de la concesión europea: **10.05.2017 EP 1351141**

54 Título: **Procedimiento y sistema de gestión de registros de datos en una red informática**

30 Prioridad:

27.03.2002 US 107287

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

04.10.2017

73 Titular/es:

**MICROSOFT TECHNOLOGY LICENSING, LLC
(100.0%)
ONE MICROSOFT WAY
REDMOND, WA 98052, US**

72 Inventor/es:

**THEIMER, MARVIN M. y
JONES, MICHAEL B.**

74 Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 635 646 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

Procedimiento y sistema de gestión de registros de datos en una red informática

Campo técnico

5 La presente invención se refiere en general a gestión de registros de datos y, más particularmente, a la gestión de los registros de datos de una red, en la que copias de los registros de datos se organizan en jerarquías lógicas de estructura de árbol y en la que registros de datos que están en gran demanda pueden replicarse para satisfacer la demanda.

Antecedentes de la invención

10 Los servicios de nombres se han usado en redes informáticas durante muchos años. En general, la principal función de un servicio de nombres es correlacionar un nombre, tales como un nombre de fichero o nombre de dominio de red, a algún registro de datos arbitrario, tales como un fichero o una dirección de red. Un servicio de nombres puede, por ejemplo, recibir una "solicitud de consulta" que incluye un nombre, tales como un nombre textual de un sitio web, de un cliente solicitante y devolver información asociada con el nombre, tal como la dirección IP del sitio web, a la parte solicitante. Otras funciones realizadas por un servicio de nombres incluyen correlacionar un nombre a un conjunto restringido de objetos de datos. Por ejemplo, un servicio de nombres de "páginas blancas" informatizado para una red de telefonía podría correlacionar un nombre de persona a múltiples piezas de información, tales como el número de teléfono de la persona y la dirección postal de la persona. Uno de los servicios de nombres más populares en uso hoy es el Servicio de Nombres de Dominio (DNS).

20 Un servicio de nombres, como otros tipos de sistemas de gestión de datos, pueden implementarse usando una base de datos distribuida. Es decir, algunos o todos los registros de datos del servicio de nombres pueden distribuirse y almacenarse en múltiples ubicaciones por toda una red. En un servicio de nombres de este tipo, siempre existirán múltiples copias o "réplicas" de cualquier registro de datos dado. Un desafío que existe en la implementación de un servicio de nombres de esta manera es proporcionar actualizaciones puntuales a las diversas réplicas de un registro de datos. Otro desafío es tratar con los así llamados "aumentos repentinos de visitas." Un aumento repentino de visitas es un gran número de usuarios que repentinamente se interesan en un único nombre o conjunto limitado de nombres administrado por el servicio de nombres y posteriormente pierden interés tras un periodo de tiempo relativamente corto. En el contexto del DNS y la Internet, por ejemplo, un aumento repentino de visitas se forma cuando un sitio web repentinamente se vuelve popular y muchos usuarios repentinamente y simultáneamente entran en el nombre de dominio del sitio web (por ejemplo www.foo.com), provocando de este modo que numerosas solicitudes para la dirección IP of www.foo.com se envíen a los servidores DNS. Un aumento repentino de visitas puede interrumpir los servicios de nombres tales como DNS porque únicamente unos relativamente pocos de los servidores de nombres en una red realmente tendrán la información solicitada por el aumento repentino de visitas y por lo tanto se desbordarán.

35 En la patente de Estados Unidos US 6.256.675 B1, se desvela un servicio de replicación que comprende una jerarquía de servidores de replicación. Esta jerarquía de replicación puede usarse para encontrar servidores de alojamiento con la mayor o menor carga. Los servidores de alojamiento pueden indicar la carga a los replicadores de nivel más bajo que eligen los servidores con la carga más alta y más baja e informar los padres en consecuencia. El procedimiento de información a los replicadores subordinados continúa hasta que alcanza la raíz del árbol que identificará los servidores de alojamiento con la mayor o menor carga y redistribuirá la carga entre los mismos. Además, el documento US 6.256.675 B1 desvela que un replicador en cada nivel puede emplearse para distribuir la carga entre conjuntos de alojamiento de sus subordinados.

Sumario de la invención

45 La invención se dirige en general a un procedimiento y sistema para la gestión de registros de datos en una red informática, en la que copias de registros de datos se distribuyen entre diversos servidores en una estructura jerárquica en árbol y en la que servidores que experimentan un número excesivo de solicitudes para un registro de datos particular transmiten réplicas de ese registro de datos a otros servidores para distribuir la carga.

50 De acuerdo con una realización de la invención, los nombres de redes se organizan y almacenan en registros de datos, denominados como "directorios," de acuerdo con su temática. La responsabilidad para el mantenimiento de copias de cada uno de los directorios se distribuye entre los servidores, que se implementan como servidores de nombres.

En diversas realizaciones de la invención, los servidores se comunican a través de una red superpuesta. En otras palabras, los servidores se organizan en una topología de encaminamiento que opera por encima de una topología de encaminamiento existente, tal como una topología de encaminamiento de IP convencional. Los servidores pueden crear y destruir dinámicamente copias de registros de datos.

55 De acuerdo con diversas realizaciones de la invención, las copias de un registro de datos particular se organizan lógicamente en una estructura jerárquica en árbol para permitir rápidas actualizaciones de las copias.

Adicionalmente, las copias se almacenan a lo largo de rutas de una topología de red superpuesta que dirige a la copia maestra del registro de datos para minimizar el número de reenvíos necesarios para que una solicitud de consulta alcance un servidor de nombres que tiene una copia del registro de datos (ya sea una copia maestra o una copia hijo).

5 De acuerdo con una realización de la invención, los servidores de la red se comunican a través de una topología de encaminamiento superpuesta en la que cada servidor tiene un sustancialmente único número asociado con el mismo, denominado en el presente documento como una ID de Nodo. La responsabilidad para el mantenimiento de la información de un registro de datos particular se determina calculando un troceo del nombre del registro de datos y asignando la responsabilidad para ese registro de datos al servidor cuya ID de Nodo es la más cercana al troceo calculado. Esto ayuda a garantizar que la responsabilidad para los diversos registros de datos se distribuye equitativamente entre los diversos servidores. En esta realización, cuando un ordenador cliente en la red requiere información de uno de los registros de datos, envía una solicitud para la consulta a uno de los servidores. La solicitud incluye un número de clave asociado con el registro de datos. El número de clave se deriva del propio nombre del registro de datos. El servidor que recibe la solicitud determina a continuación si tiene una copia del registro de datos que contiene la información que es el objeto de la solicitud de consulta. Si no tiene una copia del registro de datos, encamina la solicitud hacia el servidor que tiene responsabilidad para el registro de datos solicitado. Este encaminamiento puede implicar el reenvío de la solicitud a través de múltiples servidores intermediarios. Cada uno de estos servidores intermediarios comprueba para ver si tiene una copia del registro de datos solicitado. Si es así, a continuación responde a la solicitud. De otra manera, continúa reenviando la solicitud. La solicitud continuará siendo reenviada de servidor a servidor hasta que finalmente alcance un servidor con una copia del registro de datos solicitado (que responderá a la solicitud) o hasta que alcance el servidor que es responsable para el mantenimiento de la copia maestra del registro de datos (que asimismo responderá a la solicitud).

25 De acuerdo con diversas realizaciones de la invención, cuando un servidor detecta que el número de solicitudes de información de un registro de datos particular ha excedido un nivel de umbral, el servidor desplaza una copia de ese registro de datos a otro servidor que se encuentra en una trayectoria a lo largo de la que se ha recibido el mayor número de solicitudes de información de ese registro de datos. El servidor puede repetir este procedimiento tantas veces como sea necesario para satisfacer el aumento en la demanda. El nivel de umbral puede ser estático o puede cambiar de acuerdo con condiciones de red cambiantes.

30 De acuerdo con diversas realizaciones de la invención, un servidor que recibe una solicitud de consulta de un ordenador cliente mide la cantidad de tiempo que transcurre entre el momento que reenvía la solicitud al siguiente servidor y el momento que recibe una respuesta a la solicitud. Si el tiempo que ha transcurrido es lo suficientemente grande, entonces el servidor genera una solicitud de una copia del registro de datos que contiene la información que es el objeto de la solicitud de consulta y almacena la misma localmente. Esto ayuda a compensar los cuellos de botella de comunicación que pueden producirse en diversos puntos en la red.

35 En realizaciones aún más adicionales de la invención, actualizaciones a un registro de datos se encaminan a la copia maestra del registro de datos. El servidor que mantiene la copia maestra ejecuta la actualización y a continuación reenvía la solicitud de actualización a todas las copias hijo que ha creado del registro de datos. Esto aplica la actualización y a continuación, a su vez, reenvía la actualización a todas sus copias hijo. Esto continúa hasta que la actualización se ha reenviado a todas las copias del registro de datos en las que debe aplicarse.

Características y ventajas adicionales de la invención se harán evidentes a partir de la siguiente descripción detallada de realizaciones ilustrativas que prosigue con referencia a las figuras adjuntas.

Breve descripción de los dibujos

45 Mientras las reivindicaciones adjuntas exponen las características de la presente invención con particularidad, la invención, junto con sus objetos y ventajas, puede entenderse mejor a partir de la siguiente descripción detallada tomada en conjunción con los dibujos adjuntos de los cuales:

la Figura 1 muestra un ejemplo de una red informática en la que la invención puede practicarse;

la Figura 2 muestra un ejemplo de un ordenador en el que al menos algunas partes de la invención pueden implementarse;

50 la Figura 3 muestra un ejemplo de una red informática que incluye servidores en los que la invención puede implementarse;

la Figura 3a muestra un ejemplo de una estructura de árbol en la que esos servidores de la Figura 3 que tienen una copia de un directorio alojado en el nodo 150 pueden organizarse lógicamente;

55 la Figura 4 muestra un ejemplo de un procedimiento que puede seguirse mediante un servidor que opera de acuerdo con una realización de la invención;

la Figura 5 muestra ejemplos de cómo pueden encaminarse mensajes entre múltiples servidores de acuerdo con una realización de la invención;

la Figura 6 muestra ejemplos de estructuras de datos que pueden usarse en una realización de la invención; y

la Figura 7 muestra un ejemplo de cómo pueden crearse copias de un registro de datos de acuerdo con una

realización de la invención.

Descripción detallada de la invención

Antes de proceder con una descripción de las diversas realizaciones de la invención, se proporcionará una descripción del entorno informático y de red en el que pueden practicarse diversas realizaciones de la invención.

5 Aunque no se requiere, la presente invención puede implementarse mediante módulos de programa que se ejecutan mediante un ordenador. En general, módulos de programa incluyen rutinas, objetos, componentes, estructuras de datos y similares que realizan tareas particulares o implementan tipos de datos abstractos particulares. El término "programa" como se usa en el presente documento puede connotar un único módulo de programa o múltiples módulos de programa actuando en concierto. La invención puede implementarse en una diversidad de tipos de ordenadores. Por consiguiente, los términos "ordenador," "cliente" y "servidor" como se usan en el presente documento incluyen ordenadores personales (PC), dispositivos de mano, sistemas multiprocesador, electrónica de consumo programable basada en microprocesador, PC de red, miniordenadores, ordenadores centrales y similares. La invención también puede emplearse en entornos informáticos distribuidos, en los que las tareas se realizan mediante dispositivos de procesamiento remotos que están enlazados a través de una red de comunicaciones. En un entorno informático distribuido, los módulos pueden ubicarse tanto en dispositivos de almacenamiento de memoria locales como remotos.

Ahora se describirá un ejemplo de un entorno de red en el que puede usarse la invención con referencia a la Figura 1. La red de ejemplo incluye varios ordenadores 100 que se comunican entre sí en una red 102, representada por una nube. La red 102 puede incluir muchos componentes bien conocidos, tales como encaminadores, pasarelas, concentradores, etc. y puede permitir que los ordenadores 100 se comuniquen a través de medios por cable y/o inalámbricos.

Haciendo referencia a la Figura 2, se muestra un ejemplo de una configuración básica para un ordenador en el que puede implementarse el sistema descrito en el presente documento. En su configuración más básica, el ordenador 100 habitualmente incluye al menos una unidad 112 de procesamiento y memoria 114. Dependiendo de la configuración exacta y del tipo del ordenador 100, la memoria 114 puede ser volátil (tal como RAM), no volátil (tales como ROM o memoria flash) o alguna combinación de las dos. Esta configuración más básica se ilustra en la Figura 2 mediante la línea 106 discontinua. Adicionalmente, el ordenador también puede tener características/funcionalidades adicionales. Por ejemplo, el ordenador 100 también puede incluir almacenamiento adicional (extraíble y/o no extraíble) que incluye, pero sin limitación, discos o cintas magnéticos u ópticos. El medio de almacenamiento informático incluye memoria volátil y no volátil, extraíble y no extraíble implementada en cualquier procedimiento o tecnología para el almacenamiento de información tales como instrucciones legibles por ordenador, estructuras de datos, módulos de programa u otros datos. El medio de almacenamiento informático incluye, pero sin limitación, RAM, ROM, EEPROM, memoria flash u otra tecnología de memoria, CD-ROM, disco versátil digital (DVD) u otro almacenamiento óptico, casetes magnéticas, cinta magnética, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético o cualquier otro medio que pueda usarse para almacenar la información deseada y que puede accederse mediante el ordenador 100. Cualquier medio de almacenamiento informático tal puede ser parte del ordenador 100.

El ordenador 100 también puede contener conexiones de comunicaciones que permiten que el dispositivo se comunique con otros dispositivos. Una conexión de comunicación es un ejemplo de un medio de comunicación. El medio de comunicación habitualmente incorpora instrucciones legibles por ordenador, estructuras de datos, módulos de programa u otros datos en una señal de datos modulada tales como una onda reenviadora u otro mecanismo de transporte e incluye cualquier medio de distribución de información. A modo de ejemplo, y no como limitación, medio de comunicación incluye medio por cable tales como una red por cable o conexión por cable directa y medio inalámbrico tales como acústico, RF, infrarrojo y otro medio inalámbrico. La expresión medio legible por ordenador como se usa en el presente documento incluye tanto medio de almacenamiento como medio de comunicación.

El ordenador 100 también puede tener dispositivos de entrada tales como un teclado, ratón, lápiz, dispositivo de entrada de voz, dispositivo de entrada táctil, etc. También pueden incluirse dispositivos de salida tales como una pantalla 116, altavoces, una impresora, etc. Todos estos dispositivos son bien conocidos en la técnica y no necesitan analizarse en profundidad en este punto.

50 La operación de la invención de acuerdo con diversas realizaciones se analizará ahora en el contexto de una red 140, mostrada en la Figura 3, que incluye una pluralidad de servidores, mostrados con números 150-182 de referencia, y una pluralidad de ordenadores cliente, mostrados con números 184-188 de referencia. Los servidores 150-182 mantienen registros de datos en la red 140. En los ejemplos que se contienen en el presente documento, los ordenadores 184-188 cliente solicitan información del uno o más de los servidores 150-182. Sin embargo, se entiende que los propios ordenadores cliente también pueden actuar como servidores en algunas realizaciones. Por lo tanto, la expresión "ordenador cliente" como se usa en este contexto pretende transmitir el hecho que los ordenadores 184-188 dependen de los servidores 150-182 para el acceso a los registros de datos y no pretende limitar la implementación de los ordenadores 184-188 como clientes permanentes. De hecho, los ordenadores cliente 184-188 pueden, de hecho, ser pares de los servidores 150-182 o incluso actuar como servidores para los servidores 150-182 para otras funciones.

Los servidores 150-182 son responsables para el mantenimiento de una base de datos de nombres y registros de datos asociados con cada uno de los nombres. La base de datos se distribuye - es decir, diferentes partes de la misma se almacenan en diferentes medios legible por ordenador ubicados por toda la red 140. En algunas realizaciones de la invención, la base de datos se organiza en directorios por temática. Por ejemplo, el servidor 150 puede ser responsable del almacenamiento y mantenimiento de un directorio de nombres de redes en el encabezamiento de asunto "Compañías automovilísticas," que puede incluir direcciones de red de sitios web de todas las compañías automovilísticas más importantes. En otras realizaciones de la invención, los registros de datos no se organizan en directorios, sino que se almacenan, mantienen y replican individualmente. En general, las expresiones "registro" y "registro de datos" como se usan en el presente documento se refieren a la estructura de datos que tiene algún nombre o etiqueta y en la que se almacena algunos datos asociados con el nombre o etiqueta. El término "directorio" como se usa en el presente documento se refiere a un tipo particular de registro de datos que contiene otros registros de datos y actúa como una unidad organizacional para los otros registros de datos.

De acuerdo con diversas realizaciones de la invención, los servidores 150-182 operan todos en el contexto de una red superpuesta. Es decir, existe una topología de encaminamiento entre los servidores que sale y opera por encima de una topología de encaminamiento subyacente (por ejemplo encaminamiento IP). Existe una diversidad de posibles topologías superpuestas que pueden usarse. Estas topologías superpuestas incluyen, entre otras: PASTRY, descrita en el Acta de la 18ª Conferencia Internacional sobre Plataformas de Sistemas Distribuidos (noviembre 2001); Red de Contenido Direccional (CAN) y CHORD, que se describieron ambas en las actas de SIG COMM'01, 27-31 de agosto de 2001; y TAPESTRY, descrita en Informe N.º UCB/CSD-01-1141, publicado por la Universidad de California en Berkeley.

Para cada registro de datos de la red 140, existe al menos un servidor que tiene una copia del registro. Cuando un ordenador en la red 140 necesita realizar una consulta, presenta una solicitud de consulta a uno de los servidores 150-182. El servidor que recibe la solicitud primero determina si tiene una copia del registro de datos que tiene la información que es el objeto a la solicitud (es decir tiene acceso a un medio legible por ordenador en el que se almacena una copia del registro de datos). Si lo tiene, responde a la solicitud proporcionando información del registro de datos (por ejemplo si el nombre es un nombre de red, podría responder con la correspondiente dirección IP). Si no lo tiene, entonces el servidor reenvía la solicitud de acuerdo con algoritmos de reenvío de un programa de topología de encaminamiento que se ejecuta en el servidor. El programa de topología de encaminamiento elige el destino de reenvío a base de su conocimiento de otros servidores y su determinación de cuáles de estos servidores conocidos está lógicamente "más cerca" al servidor que tiene el registro de datos correcto. La noción del programa de topología de encaminamiento de qué servidores están lógicamente "más cerca" al servidor de destino apropiado depende de qué topología de encaminamiento superpuesta se está usando.

El resultado de este esquema de reenvío es que, para cada registro de datos, un subconjunto de los servidores (o posiblemente todos los servidores) forman una jerarquía a base de lo lógicamente "cerca" que están de un servidor que tiene una copia del registro de datos. Por ejemplo, la Figura 3a muestra como el servidor 150, que almacena y mantiene una copia del directorio de dirección de red llamado "Compañías Automovilísticas," podrían referirse a alguno de los otros servidores de la red en términos de "cercanía" lógica. En este ejemplo, se requieren dos actos de reenvío para que una solicitud de información contenida en el directorio "Compañías Automovilísticas" viaje desde el servidor 182 al servidor 150, mientras se requiere únicamente una etapa de reenvío para conseguir la misma solicitud desde el servidor 154 al servidor 150. En general, esta jerarquía virtual puede rediseñarse dinámicamente en respuesta a condiciones de red cambiantes, tales como fallos de nodos o enlaces.

Ahora se describirá una realización más específica de la invención, en la que la topología de encaminamiento superpuesta se implementa de acuerdo con PASTRY. Haciendo referencia de nuevo a la Figura 3, cada servidor en la red 140 tiene un identificador único, denominado en el presente documento como una ID de Nodo. Aunque la ID de Nodo puede tener cualquier longitud, se asume que en los ejemplos contenidos en el presente documento cada ID de Nodo es un número de 128 bits. El sistema de numeración de la ID de Nodo se envuelve sobre sí mismo. En otras palabras, cuando se usan números de 128 bits, el orden de las ID de Nodo va desde cero hasta $2^{128} - 1$, y a continuación comienza de nuevo en cero.

Cuando un servidor nuevo se une a la red 140, se asigna y calcula una ID de Nodo aleatoria al mismo. Existe una diversidad de formas de calcular una ID de Nodo. En algunas realizaciones, la ID de Nodo se calcula como un troceo criptográfico de la clave pública del servidor y su dirección IP. Cada mensaje enviado y devuelto entre los servidores de la red 140 incluye un número de clave que identifica el registro de datos en el que se ubica la información requerida para responder a la solicitud de consulta. El número de clave se calcula como un troceo del nombre de registro de datos y se toma del sistema de numeración de ID de Nodo usado por los servidores.

Cuando un servidor de la red 140 recibe un mensaje, que puede incluir tales cosas como una solicitud de una consulta de nombre o una solicitud de una copia de un registro de datos particular y el número de clave asociado con el registro de datos, el servidor determina si tiene una copia del registro de datos necesaria para servir la solicitud. Si determina que no tiene una copia del registro de datos, compara el número de clave con las ID de Nodo de otros servidores de los que es consciente y determina cuáles de las ID de Nodo tiene un valor que es más cercano al del número de clave, ya sea en términos de valor o en términos de dígitos compartidos. El servidor a continuación envía el mensaje al servidor que tiene esa ID de Nodo.

Haciendo referencia al diagrama de flujo de la Figura 4, ahora se describirá un ejemplo de cómo un servidor determina si y dónde reenviar una solicitud de acuerdo con una realización de la invención. En esta descripción, el servidor que ejecuta el procedimiento se denominará como el "primer servidor," mientras el servidor al que se reenvía el mensaje se denominará como el "segundo servidor." Se asume en esta descripción que el servidor de nombres mantiene una lista de "nodos hoja," que son un conjunto de otros servidores que están lógicamente más cerca, en términos de la topología de encaminamiento superpuesta a los del primer servidor, en comparación a todos los otros servidores que pertenecen a la red. El número total de nodos hoja se representa en lo sucesivo mediante la letra L .

En la etapa 200, el primer servidor recibe una solicitud que incluye un número de clave de un registro de datos. En la etapa 202, el primer servidor determina si tiene una copia del registro de datos que está asociada con el número de clave. Si lo está, entonces el procedimiento continúa a la etapa 204. En la etapa 204 el propio primer servidor procesa la solicitud. En la etapa 206, el primer servidor determina si conoce un segundo servidor en su conjunto de hojas cuya ID de Nodo está más cerca al número de clave que la del primer servidor. Si es así, el primer servidor envía la solicitud al segundo servidor en la etapa 212. De otra manera el procedimiento continúa en la etapa 208. En la etapa 208, el primer servidor determina si conoce a un segundo servidor cuya ID de Nodo comparte con el número de clave un prefijo que es al menos un dígito más largo que el prefijo compartido con el primer servidor. Si es así, entonces el procedimiento continúa a etapa 212, el primer servidor envía la solicitud al segundo servidor. Si no es así, entonces el procedimiento continúa a etapa 210. En la etapa 210, el primer servidor elige un segundo servidor cuya ID de Nodo comparte con el número de clave un prefijo que tiene el mismo número de dígitos que el prefijo compartido con la ID de Nodo del primer servidor y está más cerca en valor que ID de Nodo del primer servidor al número de clave. De acuerdo con esta realización, el procedimiento esbozado en la Figura 4 tendrá éxito, a menos que exista un fallo simultáneo de al menos $L/2$ servidores que (a) son vecinos inmediatos del primer servidor y (b) tienen ID de Nodo están entre la ID de Nodo del primer servidor y el número de clave.

Ahora se describirá un ejemplo de cómo se traducen nombres de redes en direcciones de redes de acuerdo con una realización de la invención. Haciendo referencia de nuevo a la Figura 3, se asume que cada uno de los servidores de la red 140 es un servidor de nombres que implementa un servicio de nombres y que el servidor 156 es responsable del alojamiento de la copia maestra de un directorio cuyo nombre es "Ordenadores" y que contiene múltiples registros de datos, correlacionando cada registro de datos un nombre textual de un sitio en la red, tal como un sitio web, que tiene información respecto a ordenadores. El directorio "Ordenadores" también puede contener nombres de subdirectorios que pertenecen a ordenadores, tales como "Ordenadores - análisis" u "Ordenadores - fabricantes." Se asume en este ejemplo que el directorio "Ordenadores" incluye el registro de datos que correlaciona el nombre "ACME Ordenadores" a la dirección IP del sitio web de ACME Ordenadores.

Haciendo referencia a la Figura 5, el ordenador 188 cliente hace una solicitud para la dirección IP de "ACME Ordenadores." En la formación de la solicitud, el ordenador 188 cliente genera un número de clave a base de un troceo del nombre en texto del directorio - "Ordenadores." La solicitud, que incluye el número de clave, el nombre de directorio "Ordenadores" y el nombre de red "ACME Ordenadores" se transmite mediante el ordenador 188 cliente al servidor 182 de nombres (Flecha A). Obsérvese que un cliente puede enviar la solicitud a cualquiera de los servidores. Preferentemente, el cliente envía la solicitud al servidor más cercano físicamente para minimizar el coste de comunicación. De acuerdo con una realización, algunos o todos los servidores se registran a sí mismos con un nombre bien conocido en DNS y el cliente localiza un servidor apropiado a través de una consulta DNS.

Continuando con el ejemplo, el servidor 182 ejecuta el procedimiento descrito en conjunción con la Figura 4 y determina que la solicitud debería enviarse al servidor 174. El servidor 182 por lo tanto reenvía la solicitud al servidor 174 (Flecha B). En esta implementación de ejemplo, cada vez que un servidor reenvía la solicitud al siguiente servidor, anota la solicitud con su propia dirección IP como el "último reenviador". Por lo tanto, para cuando la solicitud alcance su destino previsto, incluye un historial de la dirección del servidor original que recibió la solicitud (servidor 182, en este ejemplo) y la dirección del "último reenviador", que se usa para determinar dónde enviar una réplica de un directorio si se necesita.

Continuando con el ejemplo, el nodo 174 recibe la solicitud y también ejecuta el procedimiento descrito en conjunción con la Figura 4 y determina que el siguiente servidor al que la solicitud debería enviarse es el servidor 164. El servidor 174 por lo tanto envía la solicitud al servidor 164 (Flecha C). El servidor 164 repite este procedimiento y pasa la solicitud al servidor 156 (Flecha D). El servidor 156 ejecuta el procedimiento de la Figura 4 y determina que tiene el directorio correcto. El servidor 156 localiza el directorio "Ordenadores" así como la entrada en ese directorio que contiene la dirección IP de "ACME ordenadores" en una memoria a la que tiene acceso y, usando el historial de navegación incluido en la solicitud, el servidor 156 descubre que el servidor 182 fue el primer nodo de servidor en reenviar la solicitud, y por lo tanto transmite una respuesta a la solicitud al servidor 182 usando la subyacente arquitectura de encaminamiento de IP de la red 140 (Flecha E). La respuesta incluye la dirección IP de "ACME Ordenadores." El servidor 182 a continuación proporciona la dirección IP de "ACME Ordenadores" al ordenador 188 cliente.

De acuerdo con una realización de la invención, cada servidor de una red es capaz de hacer copias de registros de datos que aloja y enviar las copias a otros servidores. Para mayor facilidad de referencia, la copia del registro de datos que mantiene el servidor que aloja el registro de datos se denominará como la copia "maestra". Por lo tanto,

las copias enviadas a otros servidores son réplicas de la copia maestra. Enviar copias de un registro de datos a otros servidores ayuda a un servidor a distribuir su carga. Cuando existen múltiples copias de un registro de datos, cada servidor que tiene una copia mantiene un enlace a la copia del registro de datos "padre" y mantiene enlaces a cualquier copia "hijo" que anteriormente envió a otros servidores. Estos enlaces se usan para propagar actualizaciones a las diversas copias del registro de datos. Por lo tanto, esos servidores de la red que mantienen copias de un registro de datos particular de una jerarquía de árbol virtual con respecto al otro, con el servidor que aloja la copia maestra del registro de datos actuando como la "raíz" del árbol. El resultado de este esquema es que, para cada registro de datos que se mantiene en la red, puede existir un árbol distinto de servidores. Árboles de servidores para diferentes registros de datos pueden, sin embargo, solaparse e incluso hacerse del mismo conjunto de servidores. Se ha de observar que el árbol jerárquico formado mediante un conjunto de servidores que se encarga del mantenimiento de copias de un registro de datos particular no es necesariamente el mismo "árbol" que se forma mediante cualquier topología de encaminamiento superpuesta usada por la red.

Cada servidor en esta realización tiene una tasa de solicitud de umbral. La tasa de solicitud de umbral puede ser un valor relativamente estático preestablecido mediante, por ejemplo, un administrador de sistemas o ser un valor dinámico, tal como un valor que es una función de tales cosas como la hora del día, carga de servidor actual, carga de red actual o alguna combinación de las mismas. Si el número de solicitudes que se reciben por el servidor para un registro de datos particular excede la tasa de solicitud de umbral, el servidor responde desplazando copias de ese registro de datos a uno o más de los otros servidores de los que es consciente y habitualmente prefiere servidores desde los que se han recibido la mayoría de solicitudes para una consulta del registro de datos particular. Adicionalmente, cada servidor que recibe una copia de un registro de datos de otro servidor es capaz de borrar la copia siempre que determina que la tasa de las solicitudes que recibe para entradas en el registro de datos copiado cae por debajo de un cierto nivel de umbral. Esta habilidad para desplazar copias durante periodos de alta demanda y para borrar copias durante periodos de baja demanda ayuda al servicio de nombres a tratar con los así denominados "aumentos repentinos de visitas" que a menudo aparecen en las redes. Por ejemplo, a finales de 2001, el país de Afganistán repentinamente cautivó el interés de los usuarios de Internet por todo el mundo, aumentando de este modo el número de consultas de dirección IP de sitios web relacionados con Afganistán.

De acuerdo con una realización de la invención, cada servidor en una red determina si replicar o no uno de sus registros de datos a base de la tasa a la que recibe solicitudes de información que reside en ese registro de datos. Una vez que la tasa excede un valor umbral, el servidor desplaza una copia del registro de datos a otro servidor de los que es consciente. Para determinar cuál de los otros servidores debe recibir la copia, el servidor que está haciendo el copiado (el servidor local) compara la tasa a la que se están recibiendo las solicitudes para la entrada desde cada uno de los otros servidores de los que es consciente. El servidor local a continuación elige (de esos otros servidores que todavía no tienen una copia del registro de datos) el servidor desde el que se están recibiendo las solicitudes de información de ese registro de datos a la tasa más alta. Esto garantiza que la copia que el servidor local está haciendo se ubicará a lo largo de una trayectoria a través de la que es probable que se encaminen solicitudes de información desde ese registro de datos. El servidor determina el origen de cada solicitud analizando la "información de último reenviador" contenido en la solicitud.

Haciendo referencia a la Figura 7, se muestra un ejemplo de cómo se consigue una replicación de acuerdo con una realización de la invención. En este ejemplo, se asume que los servidores de la red 140 son servidores de nombres que implementan un servicio de nombres. Se asume que el servidor 152 ha recibido solicitudes del directorio desde otros cuatro servidores: servidor 150, servidor 160, servidor 162 y servidor 154. De manera similar, se asume que el servidor 162 ha recibido solicitudes del directorio desde otros cuatro servidores: servidor 170, servidor 172, servidor 174 y servidor 164. En este ejemplo, el servidor 152 aloja el directorio "Inversión - Noticias," que incluye entradas que correlacionan nombres de texto de sitios web de noticias de inversión a sus direcciones IP. Por lo tanto, el servidor 152 mantiene la copia maestra del directorio "Inversión - Noticias". En este ejemplo, la compañía HighFlyer.com anuncia que tendrá una oferta pública inicial de acciones. El servidor 152 se configura de modo que si recibe 500 solicitudes por hora o más para un directorio particular, replica el directorio. El día que HighFlyer.com anuncia su oferta pública, el servidor 152 recibe 1.500 solicitudes de consulta de los nombres de sitio web que se ubican en el directorio "Inversión - Noticias". Determina que, de esas solicitudes, 800 vienen del servidor 162, 400 vienen del servidor 160, 275 vienen del servidor 150 y 25 vienen del servidor 154. Ya que el servidor 162 tiene la tasa mayor de solicitudes, el servidor de alojamiento, servidor 162, replica el directorio "Inversión - Noticias" y desplaza una copia del directorio al servidor 162 (flecha A).

Asumiendo que el servidor 162 también se configura para replicar esos directorios para los que se han recibido más de 500 solicitudes por hora, el servidor 162 puede experimentar pronto el mismo problema de sobrecarga. Por ejemplo, si durante la siguiente hora, el servidor 162 recibe 800 nombres de solicitudes de consultas cuyas entradas están listadas en el directorio "Inversión - Noticias", tomará la decisión de replicar su copia del directorio. Si, por ejemplo, recibe 410 de esas solicitudes del servidor 170 y 130 de esas solicitudes de cada uno de los servidores 172, 174 y 164, el servidor 162 desplaza una copia del directorio al servidor 170 (flecha B). Durante la misma hora, el servidor 152 recibe 700 entradas de solicitudes de consulta en el directorio "Inversión - Noticias," cuatrocientas de las cuales vienen a través del servidor 160, 200 de las cuales vienen a través del servidor 150 y 100 de las cuales vienen a través del servidor 154. En respuesta, el servidor 152 desplaza una copia de la entrada al servidor 160 (flecha C).

5 Cuando se experimentan unas condiciones de alta carga sostenidas desde todas partes para un registro de datos particular, un servidor finalmente replica ese registro de datos a lo largo de todas las trayectorias de reenvío entrantes al mismo. Una vez que existe una copia del registro de datos a lo largo de cada una de las trayectorias de reenvío entrantes de la topología superpuesta, el servidor parará de recibir solicitudes reenviadas y finalizará sirviendo únicamente aquellas solicitudes que recibe directamente desde clientes. Para un registro de datos extremadamente popular, cada servidor en la red puede finalmente poseer una copia del mismo. En un caso de este tipo, cada servidor únicamente necesitará servir las solicitudes al mismo desde clientes conectados directamente y no tendrá que procesar las solicitudes reenviadas desde otros servidores.

10 En diversas realizaciones de la invención, servidores descartarán una copia de un registro de datos cuando la tasa de solicitudes de información desde la misma caiga por debajo de un cierto umbral más bajo. Para descartar una copia dada de un registro de datos, un servidor informa al servidor que tiene la copia "padre" del registro de datos (el registro de datos (servidores a los que se recibió la copia del registro de datos) e informa a los servidores que tienen copias "hijo" del registro de datos (servidores a los que ha enviado una copia del registro de datos). Se informa a los servidores que tienen copias hijo que su nuevo padre es el servidor que tiene la copia padre. De manera similar, se dice al servidor que tiene la copia padre (por el servidor que está borrando su copia) que añada a los servidores que tienen las copias hijo a su propia lista de réplicas hijo para un registro de datos dado.

15 Por ejemplo, si el servidor 162 en la Figura 7 determina que debería borrar su copia de un directorio "Inversión - Noticias," informa al servidor que tiene la copia padre, el servidor 152, y proporciona al servidor 152 la identidad de todos los servidores que tienen copias hijo del directorio. En este caso, existe únicamente uno - el servidor 170. El servidor 152 a continuación "adopta" el servidor 170 con respecto a ese directorio. El servidor 162 también informa al servidor 170 de este hecho.

20 Además de cargar en servidores individuales, otra limitación en la capacidad total de un servicio de registro de datos es la congestión de red. Un lugar en el que puede producirse la congestión de red es en los enlaces de red que interconectan los servidores. Para compensar la congestión de red, de acuerdo con una realización de la invención, los servidores que están recibiendo solicitudes de consulta directamente desde ordenadores cliente miden los tiempos de ida y vuelta de solicitudes que pasan a otros servidores. En otras palabras, cuando un servidor recibe una solicitud de consulta de un cliente, mide el tiempo que transcurre desde cuando reenvía la solicitud hasta cuando recibe una respuesta a la solicitud. Cuando se observa que los tiempos de ida y vuelta de solicitudes de información de un registro de datos dado son más grandes que un umbral dado, el servidor (en su capacidad como un intermediario para el cliente) envía una solicitud de una copia del registro de datos al servicio de nombres a través de la misma trayectoria de enrutamiento por la que se envían las solicitudes de consulta. El umbral puede ser un valor estático o puede cambiar de acuerdo con un algoritmo que representa la red y otras condiciones. La solicitud viaja hacia el servidor que aloja la copia maestra del registro de datos y o bien alcanza el servidor de alojamiento, que responde transmitiendo una copia del registro de datos al servidor solicitante o alcanza algún otro servidor que resulta que tiene una copia del registro de datos y asimismo respondería enviando una copia del registro de datos al servidor solicitante. Como alternativa y preferentemente, el servidor solicitante envía la solicitud de la copia directamente al servidor que proporcionó la respuesta a la solicitud original (es decir el servidor que tiene el registro de datos) a través de la estructura de encaminamiento de IP de la red. Por lo tanto, pueden "atraerse" copias de un registro de datos a otros servidores de nombres a través de enlaces de red congestionados, además de "desplazarse" a otros servidores desde nodos sobrecargados.

25 Para evitar la contribución a congestión de red adicional en una realización de la invención, los servidores que actúan como intermediarios de cliente también pueden almacenar en caché registros de datos para los que observan altos tiempos de ida y vuelta. Mientras estas entradas pueden almacenarse en caché únicamente durante breves periodos de tiempo antes de ser descartadas, esta forma de almacenamiento en caché puede todavía ser útil si las tasas de solicitudes de clientes son considerablemente mayores que un periodo de espera por caché (tal como cuando se forma un aumento repentino de visitas). Si los servidores fueran a solicitar copias inmediatamente en respuesta a altos tiempos de ida y vuelta observados, entonces muchos menos registros de datos populares podrían terminar replicándose así como los populares que crean la congestión de red. Esto es porque la congestión de red afecta a todo el tráfico de solicitudes que se está encaminando a través de enlaces congestionados. Para evitar esta replicación innecesaria en esta realización, los servidores únicamente inician solicitudes de copias de registros de datos para las que ellos mismos observan altas tasas de solicitudes de clientes.

30 En diversas realizaciones de la invención, se aplican, por defecto, actualizaciones a múltiples copias de un registro de datos en una manera débilmente consistente. En otras palabras, mientras una actualización a un registro de datos se está difundiendo a múltiples servidores que tienen copias del registro de datos, algunas copias reflejarán la actualización mientras otras aún tienen que recibirlas. En algunas aplicaciones, actualizaciones se aplicarán mejor en una manera fuertemente consistente, de modo que todos los clientes de la red verían vistas idénticas de, por ejemplo, los nombres administrados por un servicio de nombres en un momento dado. Sin embargo, el coste del mantenimiento de semánticas con consistencia fuerte a través de múltiples copias puede ser prohibitivo para el número de copias que pueden producirse para registros de datos populares.

35 En una realización de la invención, los clientes en una red que desean semánticas de actualización fuertemente consistentes del servicio de nombres de red pueden solicitar las mismas cuando se crea un registro de datos, pero al

hacer eso deshabilitarán la replicación para ese registro de datos. Esto es en general aceptable para registros de datos cuyos creadores saben que únicamente serán accedidos por un limitado número de clientes. El límite de escalabilidad para tales registros de datos se determinará mediante la capacidad de un único servidor para procesar solicitudes. Las máquinas modernas habitualmente son capaces de procesar unos pocos miles de solicitudes por segundo. Por lo tanto, en la práctica, todos menos los registros de datos más populares podrían probablemente declararse que son fuertemente consistentes sin afectar las latencias de solicitudes de clientes actuales en ninguna manera perceptible. Tales registros de datos, sin embargo, pueden tener que renunciar a la oportunidad de convertirse en populares. Adicionalmente, un registro de datos puede conmutarse dinámicamente entre tener semánticas de actualización débilmente y fuertemente consistentes. Sin embargo, clientes puede necesitar tener en cuenta la posibilidad que esto puede suceder comprobando en qué estado está un registro de datos cuando reciben una contestación a una solicitud que presentaron. También, cuando un registro de datos se conmuta de semánticas de consistencia débil a fuerte, todas las copias del registro de datos excepto la copia maestra deben borrarse como parte del procedimiento.

De acuerdo con una realización de la invención, todas las solicitudes de actualizaciones a un registro de datos débilmente consistente se encaminan al servidor de alojamiento para ese registro de datos. El servidor de alojamiento a continuación reenvía cada actualización a todos los servidores que tienen copias hijo de ese registro de datos. Estos, a su vez, reenvían cada actualización a sus copias hijo, hasta que finalmente cada actualización se propaga a todas las copias del registro de datos. Por ejemplo, haciendo referencia a la Figura 5, se asumirá que los servidores de la red 140 son servidores de nombres que implementan un servicio de nombres. También se asumirá que el ordenador 188 cliente es el "propietario" de la dirección de ACME Ordenadores (cuya entrada de servicio de nombres se mantiene en el directorio "Inversión - Noticias" en el servidor 152). Si la dirección IP de ACME Ordenadores ha cambiado, el ordenador 188 cliente puede transmitir un mensaje de "actualización" a su servidor 182 local. El mensaje de actualización incluye el número de clave del directorio en el que se ubica ACME Ordenadores así como la nueva dirección IP. El mensaje a continuación se reenvía al servidor 152 a través de las flechas F-K, como se muestra en la Figura 5. El servidor 152 a continuación difunde la actualización a los servidores 160 y 162, de modo que estos servidores pueden actualizar sus respectivas copias del directorio con los cambios a la entrada de ACME Ordenadores. El servidor 162, a su vez, envía la actualización al servidor 170. También, el servidor 152 envía una confirmación de la actualización de vuelta al ordenador 188 cliente a través de las flechas L y M. Como alternativa, el servidor 152 podría enviar la confirmación directamente al ordenador 188 cliente.

En diversas realizaciones de la invención, se envían mensajes de actualización usando protocolo de mensajería de solicitud-respuesta fiable. Como consecuencia, cuando los servidores que tienen una copia de un registro de datos contestan a mensajes de actualización, el emisor de la actualización será informado cuando se haya recibido con éxito una actualización, en la ausencia de fallos. Un servidor que tienen una copia del registro de datos contesta a un mensaje de actualización después de que: (1) ha reenviado el mensaje de actualización a servidores que tienen copias hijo y (2) o bien ha escuchado de vuelta de los servidores que tienen las copias hijo o las contestaciones han expirado. Por lo tanto, después de que el servidor de alojamiento de un registro de datos escucha una contestación de cada uno de los servidores que tienen las copias hijo del registro de datos o ha expirado, sabe que la actualización se ha propagado con éxito a todas las copias alcanzables del registro de datos en la red. Asumiendo que los servidores que tienen copias pueden inferir cuando no son alcanzables y ponerse ellos mismos fuera de servicio, el servidor de alojamiento, puede, en ese punto, contestar a una solicitud de actualización con una indicación de que la actualización se ha propagado con éxito a todas las copias visibles del registro de datos.

Para habilitar que los servidores que tienen una copia de un registro de datos dado detecten que no son alcanzables de acuerdo con diversas realizaciones de la invención, los servidores realizan un seguimiento de la última vez que escucharon de su servidor padre para ese registro de datos. Si un servidor no ha escuchado de su padre dentro de un intervalo de tiempo especificado, entonces asume que el padre no es alcanzable y borra su copia del registro de datos. Preferentemente, el servidor de alojamiento de un registro de datos o bien reenvía una actualización o envía un mensaje de "latido" a cada uno de los servidores que tienen copias hijo al menos una vez en cada intervalo de tiempo. Como alternativa, una copia de un registro de datos puede ponerse fuera de línea de modo que el servidor no usará más la misma para servir las solicitudes de clientes. Si el servidor escucha de su padre de nuevo, trae la copia de nuevo en línea y pregunta a su padre por cualquier actualización que se haya producido durante el periodo de fuera de línea. Esta implementación alternativa ayuda a reducir costes de recuperación.

En diversas realizaciones de la invención, cada servidor en una red mantiene estructuras de datos que implementan una red superpuesta que permite a la misma encaminar mensajes, tales como solicitudes de consulta, a otros servidores. Ejemplos de tales estructuras de datos incluyen una tabla de encaminamiento, tal como la tabla 250 de encaminamiento de la Figura 6. La tabla de encaminamiento de un servidor en diversas realizaciones de la invención se organiza en $\log_{2^b} N$ filas, en las que N es el número de servidores en la red superpuesta y b es un parámetro de configuración que habitualmente se establece en 4. Cada fila tiene 2^{b-1} entradas. Las filas se enumeran comenzando por la fila superior, que es la fila 0. Cada una de las entradas en una fila l dada de la tabla de encaminamiento se refiere a un servidor cuyo prefijo de ID de Nodo comparte l dígitos (un dígito contiene b bits) con el prefijo de ID de Nodo del servidor en el que la reside tabla de encaminamiento, que se denominará como el "servidor local."

El formato de cada entrada de la tabla de encaminamiento en una realización de la invención es como sigue: prefijo común con primer servidor - siguiente dígito - resto de ID de Nodo. Por ejemplo, en la tabla 250 de encaminamiento de la Figura 6, la ID de Nodo del servidor local es 10233102. La primera entrada de fila 0 es (NULO) -0-2212102 y representa un servidor cuya ID de Nodo tiene cero dígitos de prefijo en común con la ID de Nodo del servidor local (es decir 02212102 no tiene dígitos de prefijo en común con 10233102). La entrada sombreada (la columna 2 de fila 0) representa el dígito de inicio de la ID de Nodo del servidor local (es decir la "0^{ésima}" posición de 10233102 es 1). Como otro ejemplo, la tercera entrada de la fila cinco es 10233-2-32. Esto representa un nodo cuya ID de Nodo es 10233232, que tiene cinco dígitos de prefijo en común con la ID de Nodo del primer servidor (10233102).

Una tabla de encaminamiento en esta realización de la invención también se dispone en columnas de acuerdo con el valor del correspondiente dígito de la ID de Nodo de cada nodo. Por ejemplo, columna 0 representa servidores que tienen ID de Nodo con cero en el lugar señalado por el número de fila. Por lo tanto, fila 0, columna 0 contiene una entrada en la que la ID de Nodo es 02212102 y tiene un 0 en la 0^{ésima} posición. Fila 3, columna 0, en contraste, contiene una entrada cuya ID de Nodo es 10200230, que tiene un 0 en la posición número 3.

Para habilitar encaminamiento adecuado usando la tabla de encaminamiento recién descrita, cada servidor de una red mantiene a conjunto de hojas L . El conjunto de hojas L representa servidores cuyas ID de Nodo están numéricamente más cerca al del servidor local. Obsérvese que estas no están necesariamente "cerca" al servidor local de acuerdo con una métrica de proximidad. La métrica de proximidad se elige a base de tales factores como distancia geográfica y el número de saltos de encaminamiento de IP. Como se muestra en la Figura 6, un conjunto 252 de hojas de ejemplo se divide en dos partes: (1) entradas que representan los servidores $L/2$ cuyas ID de Nodo están numéricamente más cerca de y son más largas que la ID de Nodo del primer servidor y (2) entradas que representan los servidores $L/2$ cuyas ID de Nodo están numéricamente más cerca de y son más pequeñas que la ID de Nodo del primer servidor.

Volviendo a la descripción de la tabla de encaminamiento, cada entrada en la tabla de encaminamiento se indexa mediante la ID de Nodo del servidor que representa y también contiene la dirección IP del servidor. Cada servidor representado en una tabla de encaminamiento es potencialmente uno de muchos servidores que comparten los dígitos de prefijo apropiados con el servidor local. En principio, cualquier servidor cuya ID de Nodo coincide con la ID de Nodo del servidor local para el número apropiado de dígitos de prefijo puede usarse en la tabla de encaminamiento. En la práctica, sin embargo, para ser elegido para entrada en la tabla de encaminamiento, se requiere que un servidor esté lo suficientemente "cerca," de acuerdo con la métrica de proximidad, al servidor local.

Un ejemplo de un procedimiento que puede llevarse a cabo por cada servidor en una red para encaminar mensajes a otros servidores en una realización de la invención se muestra en forma de pseudocódigo en la Tabla 1. El procedimiento se ejecuta siempre que un mensaje con número de clave D llega a un servidor (denominado de nuevo como el "servidor local") con una ID de Nodo de A .

R_i^j : la entrada en la tabla de encaminamiento R en la columna i , $0 \leq i < 2^b$ y fila j , $0 \leq j < \lfloor 128/b \rfloor$.

L_i : la i ésima ID de Nodo más cercana en el conjunto de hojas L , $-\lfloor |L|/2 \rfloor \leq i \leq \lfloor |L|/2 \rfloor$, en el que índices negativos/positivos indican ID de Nodo más pequeñas/más grandes que la ID de Nodo presente, respectivamente.

$D(i)$: el valor del dígito i ésimo en la clave D .

$sh1(A, D)$: la longitud del prefijo compartido entre A y D , en dígitos.

- (1) si $(L_{-\lfloor |L|/2 \rfloor} \leq D \leq L_{\lfloor |L|/2 \rfloor})$ {
- (2) // D está dentro de intervalo de nuestro conjunto de hojas
- (3) reenviar a L_i , s.th $|D - L_i|$ es mínima
- (4) } si no {
- (5) // usar la tabla de rutina
- (6) Sea $1 = sh1(D, A)$;
- (7) si $(R_i^{D(1)} \neq \text{nulo})$ {
- (8) reenviar a $R_i^{D(1)}$;
- (9) }
- (10) si no {
- (11) // caso raro
- (12) reenviar a $T \in L \cap R$, s.th
- (13) $sh(T, D) \geq 1$
- (14) $? T - D ? < ? A - D ?$
- (15) }
- (16) }

Tabla 1

Dado un mensaje, el servidor local primero comprueba para ver si el número de clave pertenece al intervalo de ID de Nodo cubiertas por su conjunto de hojas (línea 1). Si es así, el mensaje se reenvía directamente al servidor de destino, a saber el servidor en el conjunto de hojas cuya ID de Nodo está más cerca de la clave (posiblemente el

servidor local) (línea 3).

5 Si el número de clave no se cubre mediante el conjunto de hojas, entonces se usa la tabla de encaminamiento y el mensaje se reenvía a un servidor que comparte un prefijo común con el número de clave por al menos un dígito más que el servidor local (líneas 6-8). En ciertos casos, es posible que la entrada aproximada en la tabla de encaminamiento esté vacía o el servidor asociado no sea alcanzable (línea 11-14), en cuyo caso el mensaje se reenvía a un servidor que comparte un prefijo con el número de clave que es al menos tan largo como el compartido con el servidor local y está numéricamente más cerca al número de clave que la ID de Nodo del servidor local. Un servidor de este tipo debería estar en la unión del conjunto de hojas y tabla de encaminamiento a no ser que el mensaje ya haya llegado al servidor con ID de Nodo numéricamente más cercana. Y, a no ser que servidores adyacentes [L|2] en el conjunto de hojas hayan fallado simultáneamente, al menos uno de esos servidores debería mantenerse conectado.

10 El procedimiento de encaminamiento descrito en la Tabla 1 normalmente converge, porque cada etapa toma el mensaje a un servidor cuya ID de Nodo o bien (1) comparte un prefijo más largo con el número de clave que el del servidor local o (2) comparte con un prefijo tan largo, pero está numéricamente más cercano al número de clave que el del servidor local.

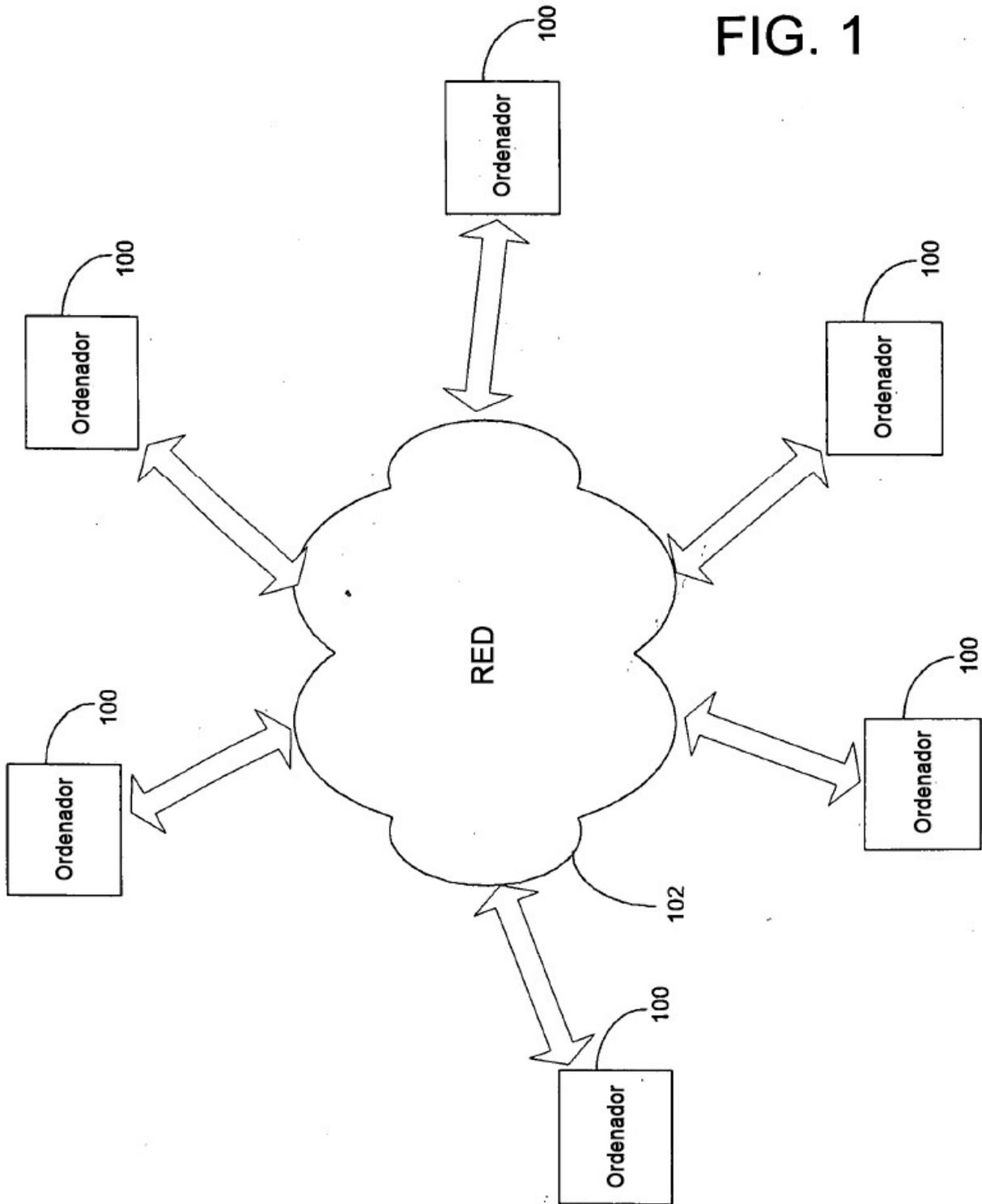
15 Por lo tanto puede verse que se ha proporcionado un procedimiento y sistema nuevo y útil para la gestión de registros de datos en una red informática. En vista de las muchas posibles realizaciones a las cuales pueden aplicarse los principios de la presente invención, debería reconocerse que las realizaciones descritas en el presente documento con respecto a las figuras de los dibujos pretenden ser únicamente ilustrativas y no deberían tomarse como limitantes del alcance de invención. Por ejemplo, los expertos en la materia reconocerán que los elementos de las realizaciones ilustradas mostradas en software pueden implementarse en hardware y viceversa o que las realizaciones ilustradas pueden modificarse en disposición y detalle. Por lo tanto, la invención como se describe en el presente documento contempla todas tales realizaciones como que pueden estar dentro del alcance de las siguientes reivindicaciones.

25

REIVINDICACIONES

1. Un procedimiento de administración de una pluralidad de registros de datos en una red (140) informática, comprendiendo cada registro de datos de la pluralidad de registros de datos información asociada con un nombre de una pluralidad de nombres, comprendiendo la red informática una pluralidad de servidores (150, 152, 154, 156, 158, 160, 162, 164, 169, 170, 172, 174, 176, 178, 180, 182), comprendiendo el procedimiento:
- 5 proporcionar, a cada uno de un subconjunto (150, 152, 154, 156, 158, 160, 162, 164, 172, 176, 180) de la pluralidad de servidores (150, 152, 154, 156, 158, 160, 162, 164, 169, 170, 172, 174, 176, 178, 180, 182), una copia de un registro de datos de la pluralidad de registros de datos, teniendo cada uno de la pluralidad de servidores (150, 152, 154, 156, 158, 160, 162, 164, 169, 170, 172, 174, 176, 178, 180, 182) una tasa de solicitud de umbral;
- 10 organizar los servidores del subconjunto en un árbol para la diseminación de actualizaciones al registro de datos, en el que uno de los servidores del subconjunto mantiene una copia maestra del registro de datos y actúa como la raíz del árbol, en el que el árbol comprende una pluralidad de bordes, cada borde definiendo una trayectoria de actualización entre un par de servidores del subconjunto, siendo la trayectoria de actualización usable para transmitir una actualización del registro de datos desde un servidor del par de servidores al otro servidor del par de servidores;
- 15 **caracterizado por**
determinar si un primer servidor del subconjunto está recibiendo un número excesivo de solicitudes de información del registro de datos y si se excede la tasa de solicitud de umbral del primer servidor;
- 20 basándose la etapa de determinación en:
- identificar, por el primer servidor, un segundo servidor de la pluralidad de servidores (150, 152, 154, 156, 158, 160, 162, 164, 169, 170, 172, 174, 176, 178, 180, 182) que no pertenece al subconjunto comparando tasas a las que se están recibiendo las solicitudes de información del registro de datos desde servidores de los que el primer servidor es consciente y elegir el servidor como segundo servidor desde el que se están recibiendo solicitudes de información del registro de datos a la tasa más alta;
- 25 transmitir una copia del registro de datos desde el primer servidor al segundo servidor para añadir el segundo servidor al subconjunto; y
añadir un nuevo borde al árbol, definiendo el nuevo borde una nueva trayectoria de actualización entre el segundo servidor y el primer servidor, siendo la nueva trayectoria de actualización usable para transmitir actualizaciones del registro de datos desde el primer servidor al segundo servidor
- 30 en el que los servidores del subconjunto tienen una relación de padre-hijo entre sí con respecto a la actualización de sus respectivas copias del registro de datos y en el que cada servidor del subconjunto tiene una ID de Nodo que está más cerca de un número de clave que cualquiera de sus hijos.
2. El procedimiento de la reivindicación 1, comprendiendo además la diseminación de la actualización del registro de datos al subconjunto de la pluralidad de servidores junto con la pluralidad de bordes.
- 35 3. Un medio legible por ordenador que tiene almacenadas en el mismo instrucciones ejecutables por ordenador para la realización del procedimiento de la reivindicación 1.

FIG. 1



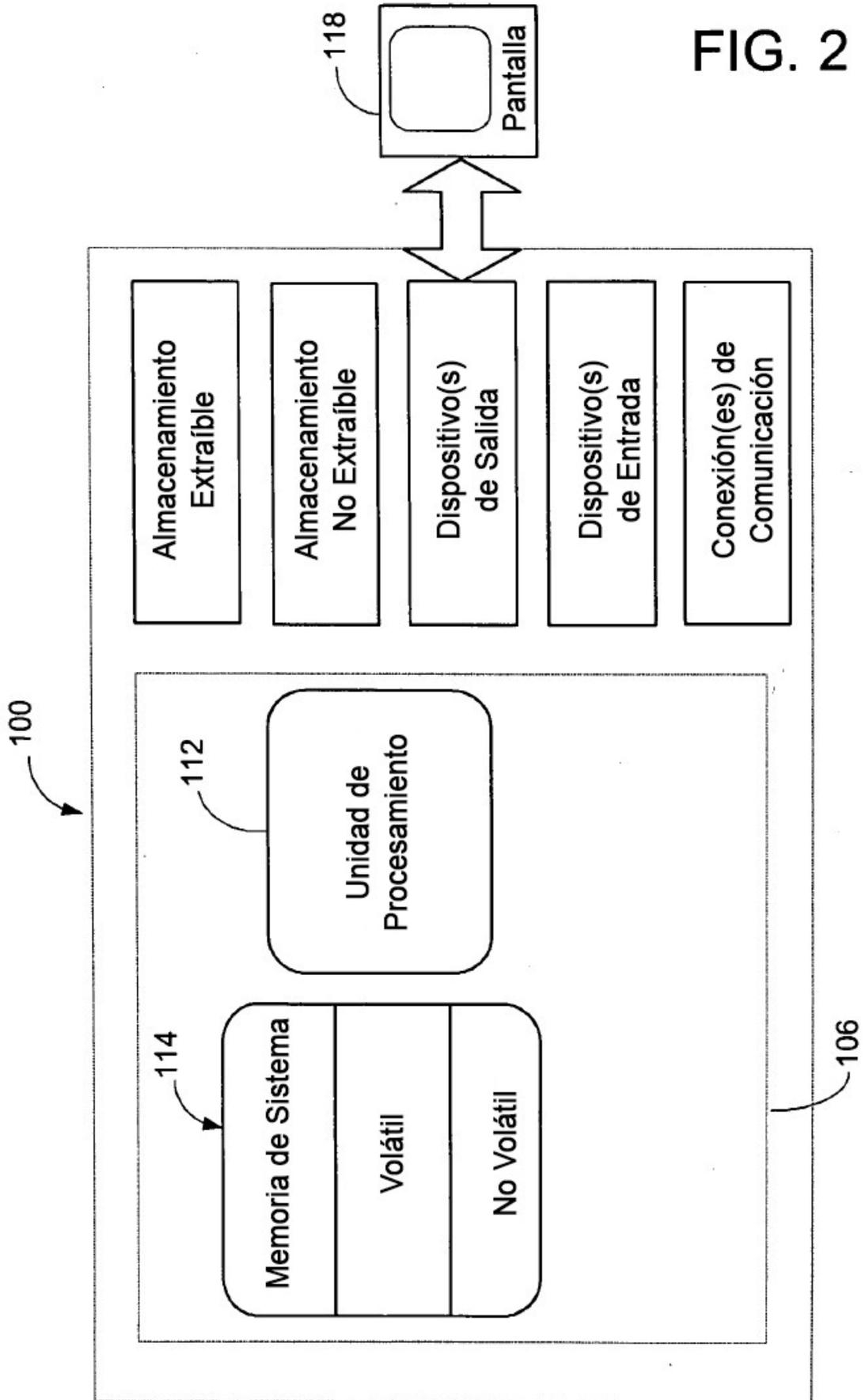


FIG. 2

FIG. 3

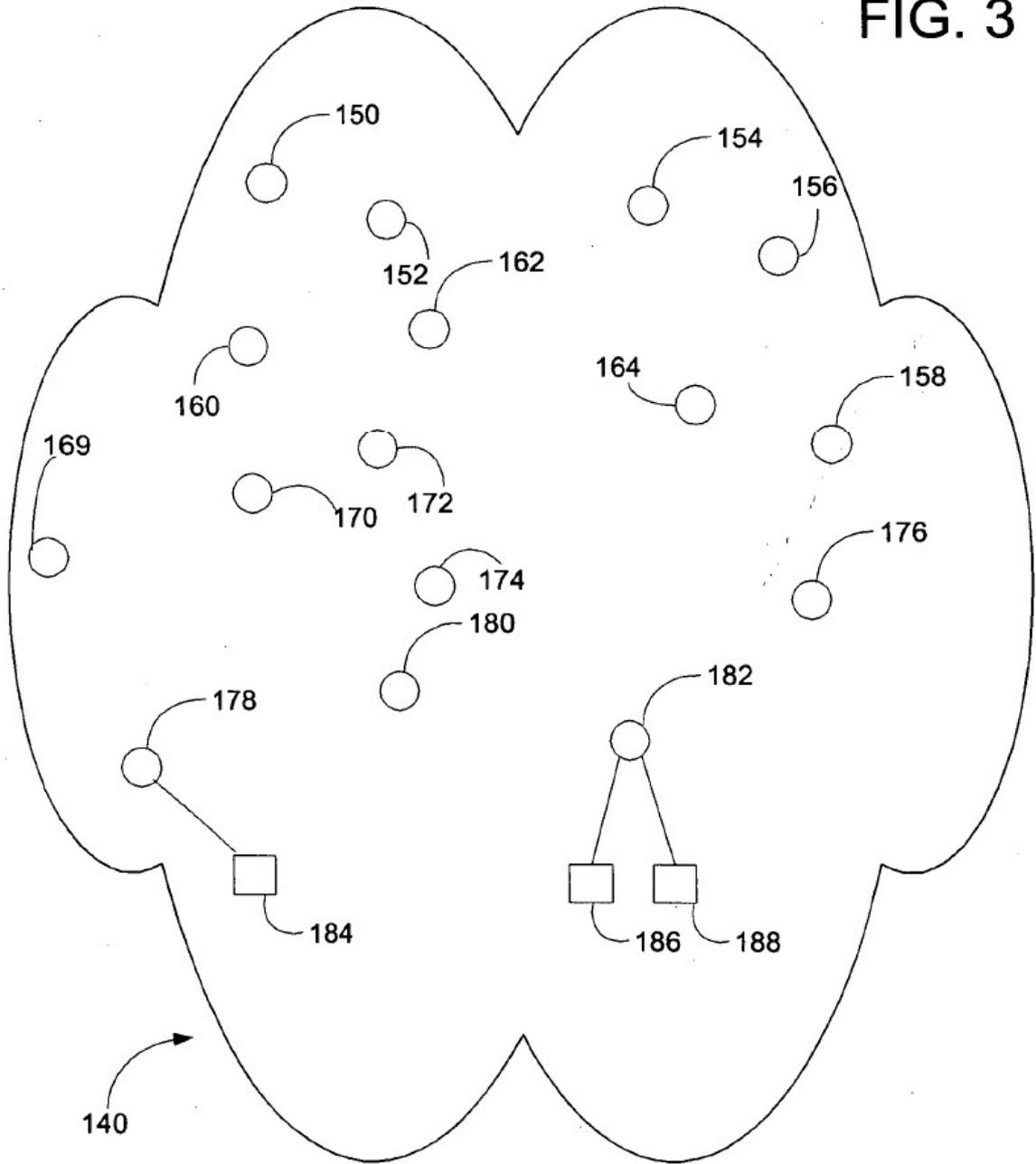


FIG. 3a

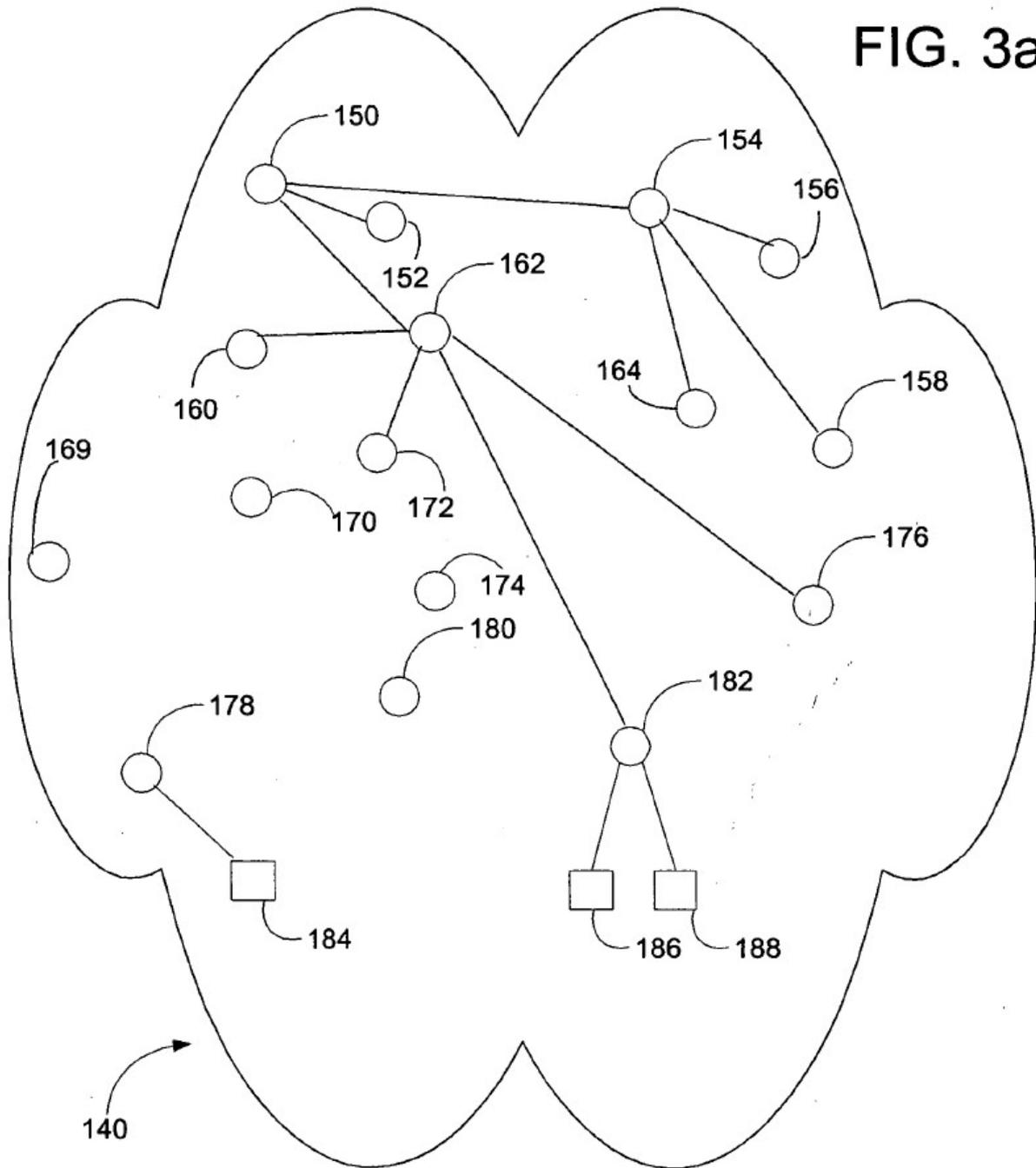


FIG. 4

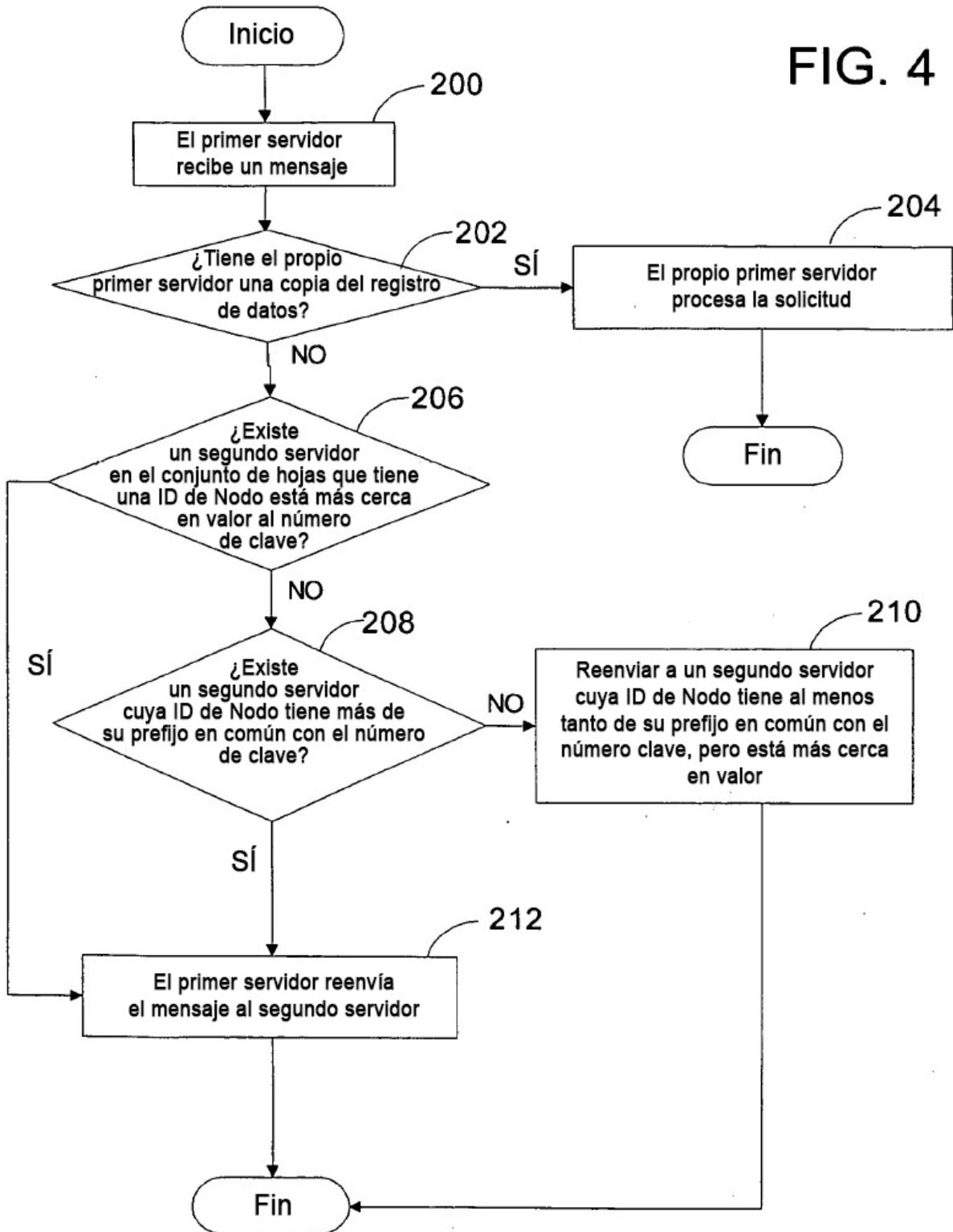


FIG. 5

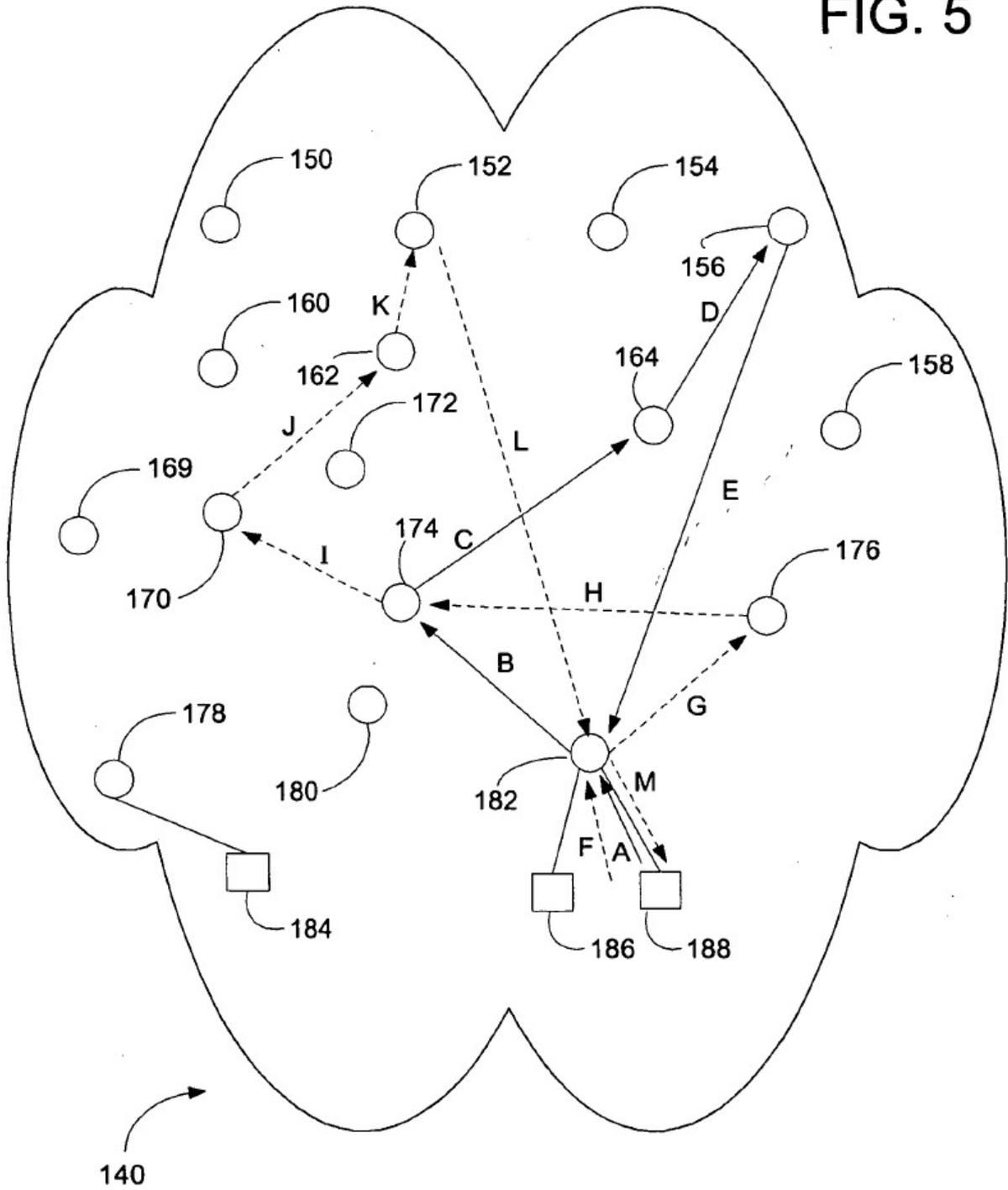


FIG. 6

Id de Nodo 10233102			
Conjunto Hoja		MAS PEQUEÑA	MAS GRANDE
10233033	10233021	10233120	10233122
Tabla de encaminamiento			
columna 0	columna 1	columna 2	columna 3
0-2212102	1	2-2301203	3-1203203
0	1-1-301233	1-2-230203	1-3-021022
10-0-31203	10-1-32102	2	10-3-23302
102-0-0230	102-1-1302	102-2-2302	3
1023-0-322	1023-1-000	1023-2-121	3
10233-0-01	1	10233-2-32	
0		102331-2-0	
		2	

FIG. 7

