

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 697 874**

51 Int. Cl.:

H04L 29/08 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **23.07.2009 PCT/EP2009/059497**

87 Fecha y número de publicación internacional: **27.01.2011 WO11009489**

96 Fecha de presentación y número de la solicitud europea: **23.07.2009 E 09780985 (9)**

97 Fecha y número de publicación de la concesión europea: **29.08.2018 EP 2457355**

54 Título: **Seguidor en sistemas P2P para optimizar funcionalidades de DVD**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
29.01.2019

73 Titular/es:
TELEFÓNICA, S.A. (100.0%)
Gran Vía, 28
28013 Madrid, ES

72 Inventor/es:
YANG, XIAOYAUN;
GJOKA, MINAS;
RODRÍGUEZ, PABLO y
CHHABRA, PARMINDER

74 Agente/Representante:
CARPINTERO LÓPEZ, Mario

ES 2 697 874 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

Seguidor en sistemas P2P para optimizar funcionalidades de DVD

Campo de la invención

5 La presente invención se refiere a sistemas de igual a igual (P2P), en particular, al diseño de un seguidor para soportar características de DVD.

Estado de la técnica

10 Un seguidor (tracker) es un componente de red que proporciona a iguales (peers) información para construir conjuntos de vecinos de alta calidad que hace posible compartir datos sin tener que basarse en el servidor excepto sólo como último recurso. Un seguidor también proporciona a cada igual un mecanismo para iniciar el suministro de datos a un igual solicitante.

15 La mayoría de las aplicaciones de compartición de archivos siguen el modelo descargar y luego consumir. En este caso, el seguidor está diseñado para hacer una selección aleatoria de iguales (como un conjunto de vecinos de un igual solicitante). Esta selección aleatoria funciona bien porque los iguales están interesados en todo el archivo. Así, el seguidor no se ocupa del suministro en secuencia de segmentos de archivos. Dos de los algoritmos más comunes usados en tales seguidores son la selección aleatoria de vecinos y la selección óptima de vecinos.

20 La selección aleatoria de vecinos (RNS) es el más simple de todos los algoritmos de selección de vecinos. En la RNS, el seguidor selecciona n iguales aleatoriamente de una lista de iguales activos y los devuelve al igual solicitante. Éste es el algoritmo convencional usado por la mayoría de los sistemas de seguidores P2P hoy en día. Éste es el más ajustable a escala de los algoritmos de seguidores ya que no requiere ninguna estructura de datos sofisticada para mantener la lista de iguales candidatos.

En la selección óptima de vecinos (ONS) se asume que el seguidor tiene un conocimiento global de cada segmento en cada igual y realiza una búsqueda exhaustiva por todos los iguales para devolver siempre a los iguales los segmentos más relevantes. Esto es muy bueno para la formación de conjuntos de vecinos de iguales. Sin embargo, es prohibitivamente caro en cuanto a la carga de procesamiento del seguidor.

25 Las demandas puestas sobre un seguidor de vídeo bajo demanda (VoD) P2P son completamente diferentes. La demanda de un usuario para el siguiente segmento está guiada por su punto de reproducción actual. Por tanto, es necesaria una elección cuidadosa del igual de un vecino para garantizar un buen rendimiento en todo el sistema. Además, la respuesta de latencia de cada petición también pasa a ser crítica porque afecta directamente a la experiencia del usuario de usar operaciones de salto, lo que requeriría una rápida selección de un conjunto de
30 vecinos para garantizar una rápida recuperación para la reproducción.

Los sistemas P2P del estado de la técnica actual ignoran la funcionalidad de DVD o saltos (avance rápido, salto hacia atrás, pausa etc.). El diseño es rígido dado que las operaciones de DVD reducen las oportunidades de compartición de archivos P2P. En un sistema ideal, el servidor debería usarse sólo como último recurso para extraer datos. Así, planificar datos en el tiempo y encontrar el conjunto correcto de iguales para compartir datos son
35 cuestiones no triviales que deben tratarse como una primera etapa para el diseño de iguales que soportan una funcionalidad de DVD y proporcionan una buena experiencia al usuario.

El documento US2009/172179 A1 divulga un sistema de transmisión en red que comprende una red, un servidor de transmisión y una serie ordenada de grupos de intercambio cooperativo desde el primer hasta el n -ésimo. El servidor de transmisión transmite uno o más clips de audio/video al nodo líder de los primeros grupos de intercambio cooperativo, que transmite el clip de audio/video recibido actualmente a los terminales de usuario dentro del mismo grupo de intercambio cooperativo, y pasa el audio/video recibido al nodo líder del siguiente grupo de intercambio cooperativo después de un intervalo de tiempo.

45 El documento US2008/256255 A1 divulga un proceso para transmitir datos de medios en una red P2P (de igual a igual) que incluye la etapa de enviar una solicitud a través de la red P2P para reproducir un segmento de tiempo de un archivo de medios. Un ordenador local está conectada a través de la red P2P a un ordenador de transmisión que tiene el segmento de tiempo deseado. El segmento de tiempo se transmite desde el ordenador de transmisión a la computadora local comenzando con un byte de datos inicial. El segmento de tiempo se almacena en el ordenador local para su reproducción a través del reproductor multimedia correspondiente.

Descripción de la invención

50 La invención se define por un procedimiento según la reivindicación 1 y un sistema según la reivindicación 4. Otras realizaciones se definen mediante las reivindicaciones dependientes. El objeto de la presente invención es proporcionar un diseño óptimo de un seguidor para soportar características de DVD (capacidad para activar la pausa, saltar hacia delante y hacia atrás en el tiempo) en los iguales en un sistema de vídeo bajo demanda (VoD) mientras que se proporciona a los usuarios una buena experiencia de visionado.

El procedimiento y dispositivo del diseño de la invención se basan en los dos principios de guiado siguientes:

- (1) Los usuarios que están en (o cerca de) el mismo punto de reproducción pueden beneficiarse mutuamente mediante colaboración.
- (2) Entre saltos, los usuarios reproducen el vídeo secuencialmente.

5 El seguidor usa (1) para agrupar usuarios que están en el mismo punto de reproducción y (2) para estimar la posición de reproducción actual de usuarios en el sistema. Juntos, (1) y (2) permiten a un seguidor encontrar y notificar iguales adecuados que pueden compartir contenido con iguales solicitantes de una manera ajustable a escala.

10 El diseño del seguidor es un proceso de dos etapas. En la primera etapa, se propone una técnica eficaz para agrupar iguales por el punto de reproducción. En la segunda etapa, se diseñan dos algoritmos inteligentes, cualquiera de los cuales puede implementarse en un seguidor; selección inteligente del conjunto de vecinos (SNS) o selección basada en el historial del conjunto de vecinos (HNS). Ambos algoritmos ayudan a un igual solicitante a formar un conjunto de vecinos de iguales con los que puede intercambiar datos. Si es necesario, tanto la selección inteligente del conjunto de vecinos como la selección basada en el historial pueden usarse para seleccionar el conjunto de vecinos, usándose el primero en primer lugar para producir un subconjunto de iguales.

15 Se usan las siguientes abreviaturas:

P2P: Sistema de igual a igual. En este caso, cada nodo actúa tanto como cliente (petición de datos) como como servidor (poniendo datos a disposición para otros iguales).

K: Número de fragmentos en un archivo de medios.

20 W: Tamaño de ventana

B: Tamaño de memoria intermedia

P_j : Punto de reproducción actual del igual j (tal como se notifica al seguidor).

L_i : Lista de iguales que reproducen el fragmento i (del total de K fragmentos).

N_B : Número de vecinos del igual B.

25 H_i : Lista de iguales que han reproducido el fragmento i en el pasado. A medida que los iguales reproducen un medio, conociendo la velocidad de reproducción y el tamaño del segmento de un igual, periódicamente, los segmentos reproducidos por un igual se actualizan en la tabla de historial mantenida por el seguidor.

30 La invención proporciona por tanto un procedimiento para agrupar iguales según la reivindicación 1. El procedimiento comprende las etapas de (a) determinar para el igual j, en el tiempo de vídeo T, el punto de salto P_j , (b) generar una clave hash HK_j . La clave hash tiene un valor $HK_j = (P_j - T)/C$, donde $C \geq 1$ es la granularidad del sistema y finalmente (c) agrupar iguales con la misma clave hash.

35 El valor de clave hash para un igual se actualiza cuando se realiza la siguiente operación de salto. En una segunda etapa, un conjunto de vecinos de un igual solicitante se selecciona del grupo de iguales con la misma clave hash que el igual solicitante. Si el número de iguales en el conjunto de vecinos (iguales que tienen la misma clave hash) es menor que n, en una etapa siguiente los iguales restantes se seleccionan de los vecinos con una clave hash que es mayor que, pero está cerca de la clave hash del igual solicitante, es decir de vecinos que ya han reproducido el segmento de reproducción k del igual solicitante. Si el número de iguales en el conjunto de vecinos es todavía menor que n, en una etapa siguiente los iguales restantes se seleccionan aleatoriamente.

Breve descripción de los dibujos

40 Para completar la descripción y con el fin de proporcionar una mejor comprensión de la invención, se proporciona un conjunto de dibujos. Dichos dibujos forman una parte integrante de la descripción e ilustran una realización preferida de la invención, que no debe interpretarse como restrictiva del alcance de la invención, sino sólo como un ejemplo de cómo puede realizarse la invención. Los dibujos comprenden las siguientes figuras:

45 La figura 1 describe un ejemplo que demuestra la agrupación de iguales por el punto de reproducción en el seguidor.

La figura 2 es un diagrama de flujo del algoritmo SNS de la invención.

La figura 3 es un diagrama de flujo del algoritmo HNS de la invención.

La figura 4(a) muestra cómo se usan las claves hash para agrupar iguales para el ejemplo en la figura 1.

50 La figura 4(b) muestra la imagen instantánea del contenido de la tabla de historial para el ejemplo de la figura 1 para el algoritmo HNS.

La figura 5 muestra una comparación del rendimiento de los algoritmos de selección de iguales en el seguidor.

Descripción detallada de la invención

El seguidor es un componente clave en cualquier sistema P2P. En un sistema de VoD P2P, tiene un papel particularmente clave dado que los iguales establecerán un contacto con el seguidor en:

55 (1) Tiempo de inicio, cuando se hace una petición de contenido.

(2) Cada operación de DVD dado que el igual necesita encontrar otros iguales con los que pueda intercambiar datos sin tener que ir al servidor para cada segmento.

(3) Cada vez que un igual determina que su conjunto de vecinos no contiene iguales de alta calidad para intercambiar datos.

5 La presente invención proporciona un procedimiento para agrupar iguales por el punto de reproducción para su uso en sistemas de VoD que soportan una funcionalidad de DVD. Entonces son posibles dos maneras de selección del conjunto de vecinos en el seguidor, ambas basadas en esta primera agrupación. Para que sea eficaz, es necesario que el seguidor agrupe iguales que puedan compartir datos. Cada vez que un igual necesita una lista de vecinos, contacta al seguidor con su punto de reproducción actual. Basándose en el punto de reproducción, el algoritmo de selección del conjunto de vecinos en el seguidor invoca dos mecanismos: (1) Encontrar el punto de reproducción del igual solicitante. (2) Selección del conjunto de vecinos que selecciona una lista de iguales que están en el mismo punto de reproducción que el punto de reproducción del igual solicitante y contiene la información de vídeo necesaria.

15 En primer lugar, se muestra cómo agrupar iguales por el punto de reproducción. Entonces se propondrán dos algoritmos de selección del conjunto de vecinos, la selección inteligente del conjunto de vecinos (SNS) y la selección basada en el historial del conjunto de vecinos (HNS) para ayudar a un igual a intercambiar datos con su conjunto de vecinos.

Agrupar iguales por el punto de reproducción: Para implementar los algoritmos de selección inteligente del conjunto de vecinos y selección basada en el historial, es necesario que el seguidor conozca la parte del vídeo que está reproduciendo cada igual cuando recibe una petición de igual para una lista de vecinos. Dado que el punto de reproducción aumenta con el tiempo, un seguidor convencional tendría que volver a calcular el punto de reproducción de cada igual en cada petición, incurriendo así en una sobrecarga muy alta. Para evitar tener que hacer un cálculo por cada igual, se creó una tabla hash que mantiene en cada entrada todos los iguales que están reproduciendo la misma parte del vídeo. Se usa una clave hash que una función del tiempo de vídeo cuando se produjo el salto y el punto del salto final. Se usa una operación de módulo simple como función hash.

25 Esto se ilustra usando un ejemplo. Para una operación de salto en el tiempo de vídeo T , el igual j notifica el punto de salto, P_j , al seguidor. El seguidor genera una clave hash HK_j tal como sigue: $HK_j = (P_j - T)/C$, donde $C \geq 1$ es la granularidad del mecanismo para predecir el punto de reproducción de iguales. La granularidad se elige según la tolerancia de los usuarios. Debe ser lo más pequeña posible, siendo el límite inferior de 1 segundo. Sin embargo, una granularidad muy baja puede conducir a interpretaciones erróneas. Se ha calculado que la C óptima debe ajustarse a no menos de 5 segundos. Entonces se elimina el igual de la entrada hash antigua y se inserta en la nueva entrada con clave como HK_j . Obsérvese que la clave hash HK_j es estática y no es necesario que el seguidor actualice el igual en la tabla hash hasta la siguiente operación de salto. Ésta es una de las ventajas de la invención.

30 En la figura 1, se demuestra este enfoque usando un ejemplo numérico. El igual A comienza a ver un vídeo en el tiempo 2 y en el tiempo 6 salta al punto de reproducción 7. Asumiendo que $C = 1$, la clave hash para el igual A en el salto es 1. La clave hash del igual A es la proyección de su punto de reproducción en el tiempo de salto. Se asume que el igual B ha comenzado la reproducción en el tiempo 4. En el tiempo 8, el igual B salta al punto de reproducción 9. En el tiempo 9, el igual B también tiene una clave hash 1 y el seguidor devolverá A como un igual en el mismo grupo que el igual B. Si la duración de los medios es muy grande, el tamaño de la tabla hash puede alcanzar potencialmente el número de iguales en el seguidor. La figura 4(a) proporciona una imagen instantánea de las claves hash antes y después del salto.

35 Para delimitar el tamaño de la tabla hash y para garantizar que los iguales que abandonan el sistema se eliminan, cada igual establece periódicamente contacto con el seguidor para informarle de su posición de reproducción actual. Este tiempo de mantenimiento en estado activo, (t_a), se ajusta a 15 minutos. Con notificaciones periódicas, el número de claves no es más de $(L + t_a)/C$. En este caso, L es la duración del vídeo. Para una película de 120 minutos, $t_a = 15$ minutos y $C = 1$ minuto, sólo hay 135 diferentes claves activas posibles en ese seguidor en cualquier tiempo. Al asignar 135 entradas en la tabla, pueden evitarse colisiones y conseguirse una complejidad de $O(1)$ para acceder a/insertar en una tabla hash.

40 Dado que ahora se tiene un mecanismo ligero para agrupar iguales por el punto de reproducción y hacer actualizaciones rápidas en operaciones de salto, se comenta el diseño de dos algoritmos de selección del conjunto de vecinos.

Selección inteligente del conjunto de vecinos:

El algoritmo SNS se muestra a continuación.

55 **Algoritmo 1** Selección inteligente del conjunto de vecinos (SNS)

Requisito: El archivo de medios se divide en K fragmentos.

Requisito: Para cada fragmento de medios i , el seguidor mantiene una lista de iguales L_i que reproducen el fragmento i .

Resultado: El seguidor devuelve n vecinos a un igual solicitante.

1: Un igual A pone en contacto el seguidor con un punto de reproducción P al que quiere saltar.
 2: El conjunto de vecinos de A, $N_A = \Phi$ (inicialmente, ajustado a conjunto vacío)
 3: El seguidor estima el fragmento j del punto de reproducción P, $1 \leq j \leq K$, $P \in$ fragmento j y calcula su clave hash HK_j .
 5 4: El seguidor coge aleatoriamente n iguales de la lista L_j y los añade a la lista N_A
 5: recuento $\leftarrow j$
 6: suma \leftarrow Tamaño(L_j)
 7: **mientras** suma < n **hacer**
 8: recuento \leftarrow recuento + 1
 10 9: suma \leftarrow suma + Tamaño($L_{recuento}$)
 10: añadir el $L_{recuento}$ ajustado a N_A
 11: **fin mientras**
 12: Añadir el igual A a la lista L_j
 13: **devolver** el conjunto de vecinos de A, N_A

15 La lógica de este algoritmo para seleccionar n vecinos sigue las etapas siguientes, una vez que una lista de iguales con la misma clave hash que el igual solicitante se ha generado según la etapa anterior:

1. Si no pueden encontrarse n vecinos en el mismo punto de reproducción del igual solicitante, se intenta obtener los iguales restantes de una lista de iguales que están cerca del punto de reproducción del igual solicitante y tienen una clave hash mayor. El fundamento detrás de esta elección es que los iguales cercanos en el tiempo al igual solicitante pueden haber reproducido ya el segmento y por tanto, pueden proporcionarlo al igual solicitante. Así, para un igual que ha saltado a un punto de reproducción en el fragmento j, si el seguidor no puede encontrar n iguales en el punto de reproducción j, el seguidor consultará iguales en el punto de reproducción j+1, j+2, ... K etc. en ese orden.
 20 2. Si el seguidor llega al último fragmento K y todavía no encuentra n vecinos, cogerá aleatoriamente los vecinos restantes de puntos de reproducción [1, j].
 25

El algoritmo consiste, por tanto, en dos etapas: (1) calcular el punto de reproducción de un igual solicitante es una operación de tiempo constante tal como se comentó anteriormente y (2) encontrar n vecinos es también una operación de tiempo constante ya que el número de fragmentos K en un archivo de medios es pequeño, mucho más pequeño que el número de iguales en el sistema.

30 No hay directrices rígidas para la elección del valor de n, el número mínimo de otros iguales a los que puede estar conectado un igual. El valor de n depende de la aplicación de VoD. Dependiendo de las condiciones de red y la estabilidad de los iguales, la aplicación P2P determina una elección adecuada de n. Normalmente se elige un valor de $n > 10$.

35 Tal como se observa en la discusión anterior, el algoritmo SNS en el seguidor es de tiempo constante e intenta encontrar rápidamente iguales que pueden colaborar. Un diagrama de flujo del algoritmo SNS se muestra en la figura 2.

Usando la figura 4(a), se demuestra que para el ejemplo en la figura 1, la clave hash del igual A (HK_A) era inicialmente -2. De manera similar, la clave hash del igual B (HK_B) era inicialmente -4. En el tiempo = 6, el igual A salta dando como resultado una nueva $HK_A = 1$. De manera similar, cuando el igual B salta en el tiempo = 8, la nueva $HK_B = 1$. Para la SNS, el igual A se devolverá al igual B como un vecino para intercambiar datos. Si hay menos de n iguales con la misma clave hash, el algoritmo intentará en primer lugar encontrar los iguales restantes con una clave hash mayor y si no se encuentran suficientes iguales, intentará y encontrará iguales aleatoriamente.

Selección basada en el historial del conjunto de vecinos: El algoritmo HNS se muestra a continuación.

Algoritmo 2: Selección basada en el historial del conjunto de vecinos (HNS)

45 **Requisito:** El archivo de medios se divide en K fragmentos.

Requisito: Para cada fragmento de medios i, el seguidor mantiene una lista de iguales L_i que reproducen el fragmento i.

Requisito: El seguidor devuelve n vecinos a un igual solicitante.

50 **Requisito:** Para cada operación de salto, el seguidor genera un historial de segmentos, H_i , reproducido por el igual. Así, cada punto de reproducción $1 \leq j \leq K$ contiene una lista de iguales que reprodujeron ese segmento.

1: Un igual B pone en contacto el seguidor con un punto de reproducción P al que quiere saltar.
 2: El conjunto de vecinos de B, $N_B = \Phi$ (el conjunto de vecinos de B es un conjunto vacío).
 3: El seguidor estima el fragmento j del punto de reproducción P, $1 \leq j \leq K$, $P \in$ fragmento j.
 4: El seguidor coge n iguales aleatoriamente de la lista L_j y los añade a la lista N_B
 55 5: **si** Tamaño(L_j) < n **entonces**
 6: Coger los n - Tamaño(L_j) iguales restantes aleatoriamente de la lista H_j y añadirlos a N_B
 7: **fin si**
 8: Coger n - Tamaño(L_j) iguales de las listas L_i , $j \leq i \leq K$
 9: Devolver vecinos para B, N_B

El algoritmo de selección basado en el historial del conjunto de vecinos difiere de la SNS en un aspecto clave. Además de mantener una lista de iguales L_j para cada punto de reproducción j ($1 \leq j \leq K$), el algoritmo también mantiene H_j , un historial de todos los iguales que han reproducido el segmento j . Conociendo la velocidad de reproducción de los medios, el historial de fragmentos reproducidos por un igual se actualiza periódicamente. Obsérvese que al menos para el contenido popular, el número de iguales que han reproducido el segmento j y están activos en la nube (*swarm*) (y por tanto están en la tabla H_j) es probable que sea mayor que el número de iguales que están reproduciendo el segmento j (y forman parte de la lista L_j).

La lógica de este algoritmo para seleccionar n vecinos sigue las etapas siguientes:

1. Como en SNS, se identifica en primer lugar el punto de reproducción del igual solicitante y se obtiene una lista de iguales que están en el mismo punto de reproducción (y que, por tanto, tienen la misma clave hash). Si no está ya actualizada, se actualiza la tabla de historial con todos los fragmentos reproducidos desde la última actualización.
2. Si no pueden encontrarse n vecinos en el mismo punto de reproducción que el igual solicitante, se intenta obtener los iguales restantes de la tabla de historial. El fundamento detrás de esta elección es que se mantiene un historial H_i de cada fragmento i visionado por cada igual.
3. Si el seguidor todavía no encuentra n vecinos, cogerá en primer lugar los vecinos restantes de puntos de reproducción $[j, K]$.
4. Si todavía no encuentra n vecinos, cogerá los iguales restantes de puntos de reproducción $[1, j)$. Un motivo por el que esto puede funcionar es que los iguales usan el algoritmo de los más raros primero en el planificador de cargas, lo que permite a los iguales descargar fragmentos que no se han reproducido aún localmente. Un diagrama de flujo del algoritmo HNS se muestra en la figura 3.

Para generar la tabla de historial del contenido de cada igual, no es necesario que los iguales notifiquen periódicamente su contenido al seguidor. En su lugar, al estimar su punto de reproducción actual (al conocer la velocidad de reproducción de los medios que están reproduciendo) y el tiempo desde el último contacto (ya sea el tiempo de inicio, o la última operación de salto o la última petición de nuevos vecinos), el seguidor puede deducir el contenido de cada igual y renovar por consiguiente la tabla de historial. La renovación depende tanto del número de iguales como del periodo de tiempo de renovación. Por tanto la renovación, que puede ejecutarse como una tarea de fondo, tiene una complejidad de $O(n)$, mientras que la consulta de iguales es $O(1)$.

La segunda parte de la selección del conjunto de vecinos (HNS) requiere una tabla de historial separada que contiene los segmentos que tiene cada igual. Mantener el historial sobre un igual del orden de segundos requiere mucha memoria y hacer búsquedas en el mismo también requiere muchos recursos. En esta implementación, se dividen vídeos en fragmentos de tiempo de tamaño C . La figura 4(b) muestra imágenes instantáneas de la tabla de historial en diferentes puntos de tiempo para el ejemplo en la figura 1.

El requisito de memoria es $N_p * L * \text{tamaño}(P_{id})/C$, donde L es la duración de la película, P_{id} es el id de igual único de cada igual y N_p es el número total de iguales. Cada igual tiene un id único de 32 bits. Por tanto, el seguidor requerirá aproximadamente 18 MB de memoria para construir el historial para 20K iguales, para una duración de la película de 120 minutos y con $C = 30$ segundos. Así, los requisitos de memoria son muy moderados para mantener una tabla de historial de este tipo.

Impacto de los algoritmos de correspondencia de iguales: Para evaluar el rendimiento de diversos algoritmos de asignación de iguales, se evaluaron HNS y SNS frente a RNS y ONS. Se mostrará que una selección aleatoria de tales iguales puede dar como resultado conjuntos de vecinos en los que los iguales no tienen datos para compartir, dando como resultado un mal rendimiento del sistema.

La figura 5 compara el rendimiento de diferentes algoritmos, concretamente SNS y HNS, frente a un algoritmo de selección óptima del conjunto de vecinos (ONS) y el más simple entre todos, RNS. Dado que ONS tiene un conocimiento global de cada segmento en cada igual y realiza una búsqueda exhaustiva en todos los iguales, siempre devuelve iguales con los segmentos más relevantes. Si hay N iguales en el seguidor y un archivo de medios de M fragmentos, cada petición al seguidor es $O(N)$. A partir de la figura 5, se observa que con una carga de trabajo diferente con un mayor número de saltos (a partir de una traza de VoD activa), las estrategias simples de selección del conjunto de vecinos tienen un mal rendimiento. Inicialmente, los algoritmos tanto HNS como SNS tienen aproximadamente el mismo porcentaje de vecinos útiles. Sin embargo, con el tiempo, a medida que los usuarios saltan, la fracción de vecinos útiles devuelta por el seguidor es bastante pequeña tanto para SNS como para RNS, pero es muy alta y cercana a la estrategia de selección óptima de iguales para HNS. En general, SNS tiene un rendimiento mucho mejor que RNS dado que SNS selecciona iguales que están cercanos al punto de reproducción del igual solicitante y por tanto, tiene una mayor probabilidad de formar conjuntos de vecinos útiles. HNS tiene un rendimiento casi óptimo y mucho mejor que SNS dado que HNS devuelve iguales que han reproducido el segmento solicitado en el pasado. Los iguales devueltos por SNS, por otro lado, pueden no tener siempre el segmento solicitado dado que no todos los iguales devueltos pueden estar dentro de una ventana del punto de reproducción del igual solicitante (porque los usuarios saltan).

Se ha mostrado que la HNS tiene un rendimiento casi óptimo y es de tiempo constante. El requisito de espacio de la HNS es mínimo. Aunque SNS es mejor que RNS, no tiene un rendimiento tan bueno como HNS principalmente

debido al hecho de que no aprovecha el hecho de que otros iguales activos que no comparten el punto de reproducción con el igual solicitante pueden haber visionado el mismo contenido en el historial reciente. Tanto HNS como SNS han mostrado ser eficaces en un diseño de seguidor para soportar operaciones de DVD en sistemas de VoD P2P.

5 Adicionalmente, pruebas de ajuste a escala con el seguidor mostraron que con patrones de salto con hasta 16.000 iguales simultáneos, el seguidor puede responder con menos de 0,1 segundos de retraso. La utilización de memoria máxima observada fue de 32 Mbytes con un pico de utilización de la CPU del 74% para el algoritmo SNS. Éste es un valor insignificante en comparación con el tiempo de transmisión de los primeros segmentos del vídeo (2 segundos para un segmento de 64 KB en un enlace de 1 Mbps) y las potentes máquinas disponibles hoy en día.

10 Algunas aplicaciones del seguidor son:

- Un subsistema dentro de un sistema P2P para coordinar todos los iguales (o un subconjunto de iguales si se usan múltiples seguidores).
- Como un gestor de conjunto de vecinos de iguales en un sistema P2P para ayudar a los iguales a encontrar conjuntos de vecinos de alta calidad para intercambiar datos.

15 En este texto, el término “comprende” y sus derivaciones (tal como “que comprende”, “comprendiendo”, etc.) no deben entenderse en un sentido excluyente, es decir, estos términos no deben interpretarse como que excluyen la posibilidad de que lo que se describe y define pueda incluir elementos, etapas, etc. adicionales.

20 Por otro lado, la invención obviamente no se limita a la(s) realización(es) específica(s) descrita(s) en el presente documento, sino que también abarca cualquier variación que pueda considerar un experto en la técnica (por ejemplo, en lo que respecta a la elección de componentes, configuración, etc.), dentro del alcance general de la invención tal como se define en las reivindicaciones.

REIVINDICACIONES

1. Un procedimiento para agrupar iguales en un sistema P2P en el que pueden realizarse operaciones de salto de DVD, comprendiendo el procedimiento las etapas de:
- 5 a. determinar para cada igual y operación de vídeo un tiempo T y un punto de salto P_j , en el que el punto de salto P_j es el punto de reproducción al que dicho igual salta en el tiempo T ;
 - b. generar una clave hash HK_j para cada igual, teniendo la clave hash un valor $HK_j = (P_j - T)/C$, donde $C \geq 1$ es la granularidad del sistema
 - c. agrupar los iguales con la misma clave hash,
 - 10 d. seleccionar un grupo de vecinos de un igual solicitante del grupo de iguales con la misma clave hash que el igual solicitante, en el que tras determinar que el número de iguales en la vecindad del igual solicitante es menor que n , siendo n el número mínimo de otros iguales al que un igual se puede conectar, seleccionándose los iguales restantes del grupo de vecinos desde una lista de iguales con una clave hash mayor que y cerca de la clave hash del igual solicitante o de una lista de iguales que ya han reproducido el segmento de reproducción j del igual solicitante.
- 15 2. Un procedimiento según la reivindicación 1, en el que el valor de clave hash se actualiza cuando se realiza la siguiente operación de salto.
3. Un procedimiento según las reivindicaciones 1 o 2, en el que, si el número de iguales en el conjunto de vecinos es menor que n , en una etapa siguiente los iguales restantes se seleccionan aleatoriamente.
- 20 4. Un sistema P2P para aplicaciones de vídeo, comprendiendo el sistema un seguidor adaptado para soportar operaciones de DVD de salto, en el que el seguidor está adaptado para realizar cualquiera de los procedimientos según las reivindicaciones 1-3.

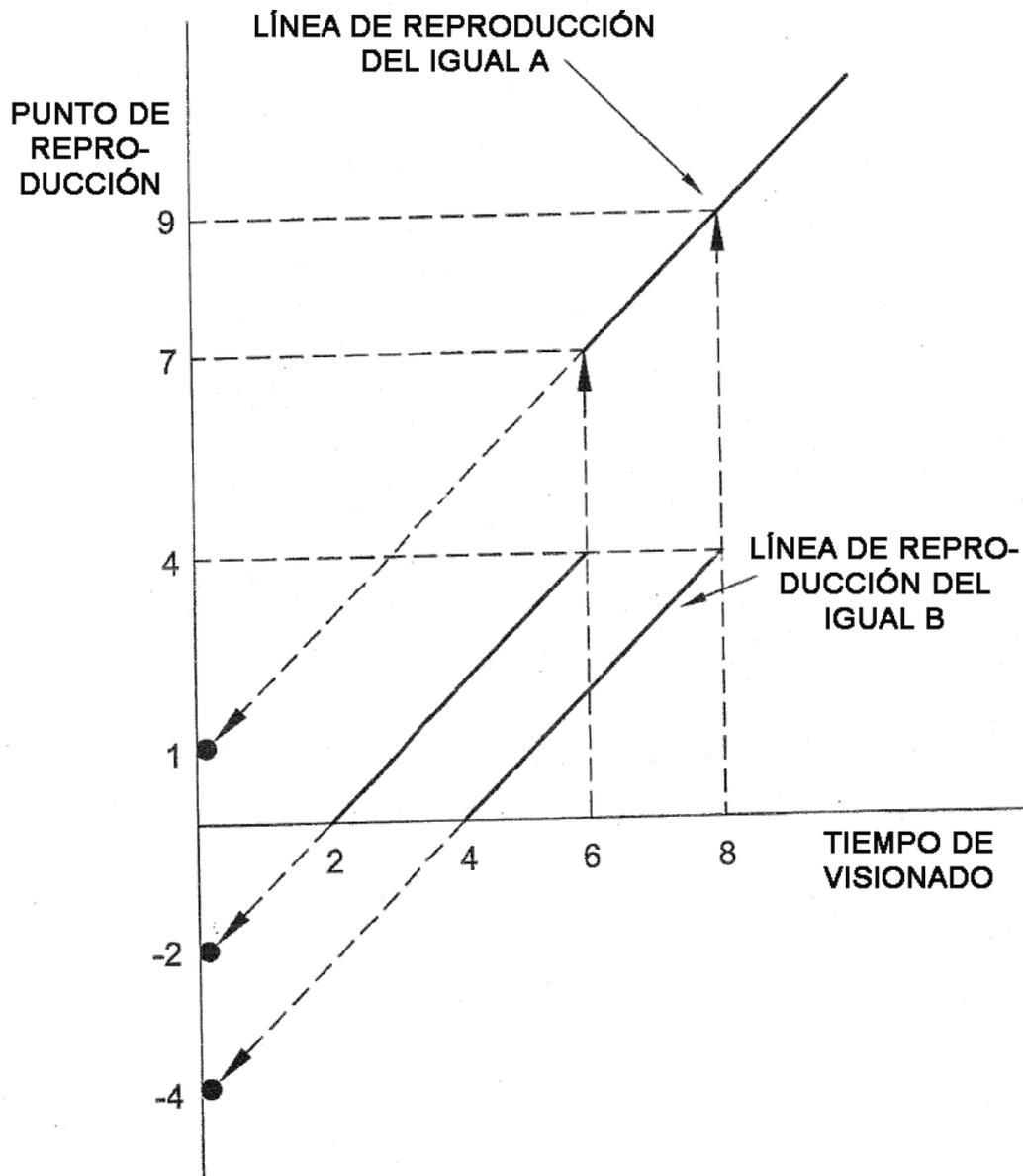


FIG. 1

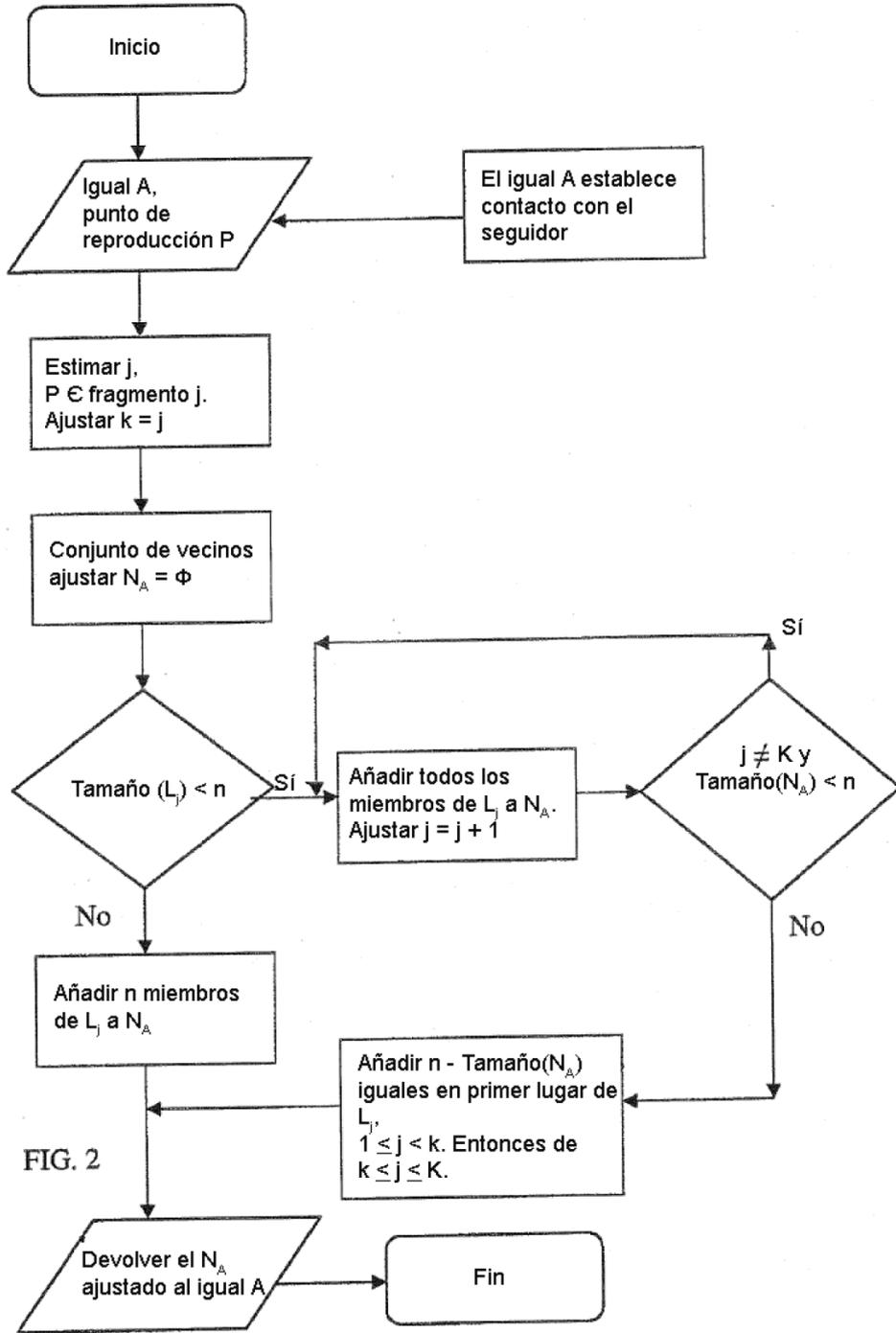


FIG. 2

FIG. 2

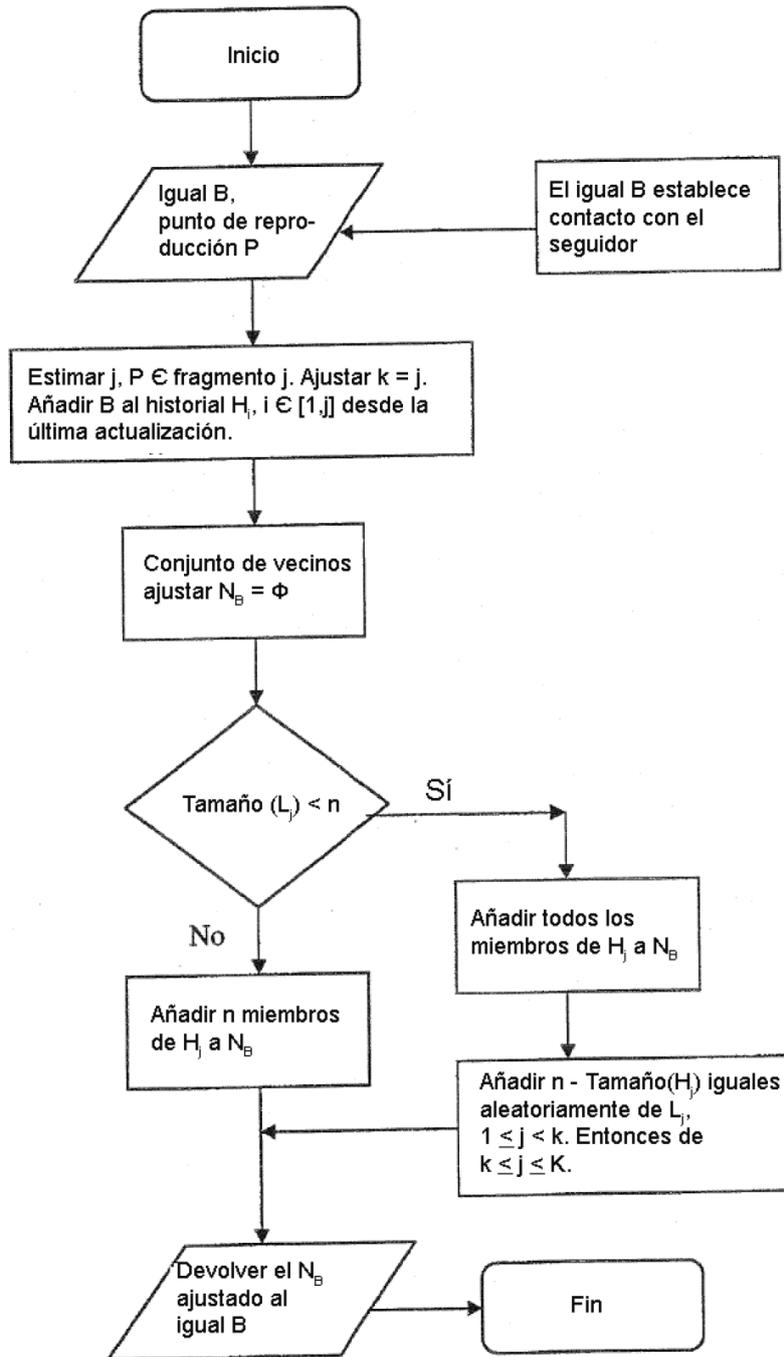


FIG. 3

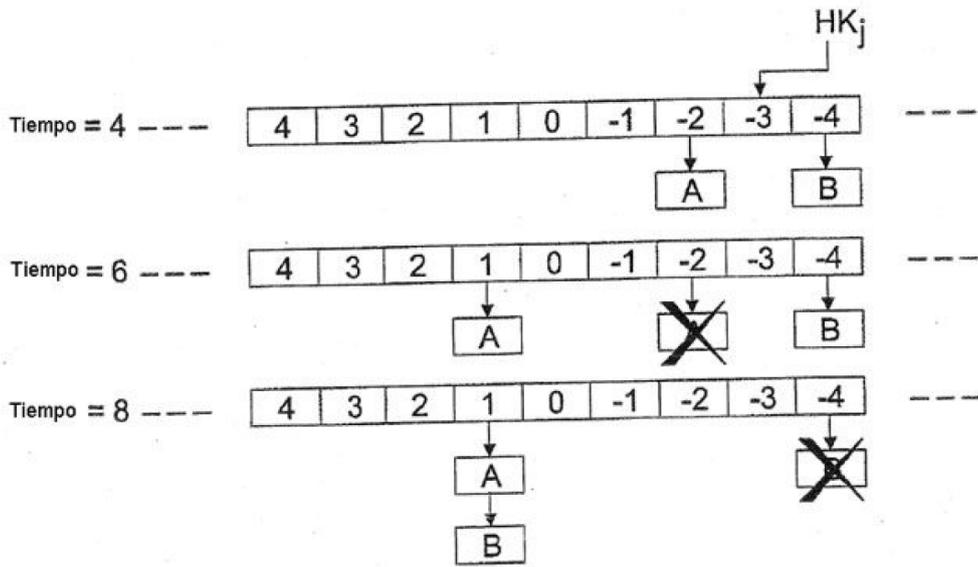


FIG. 4(a)

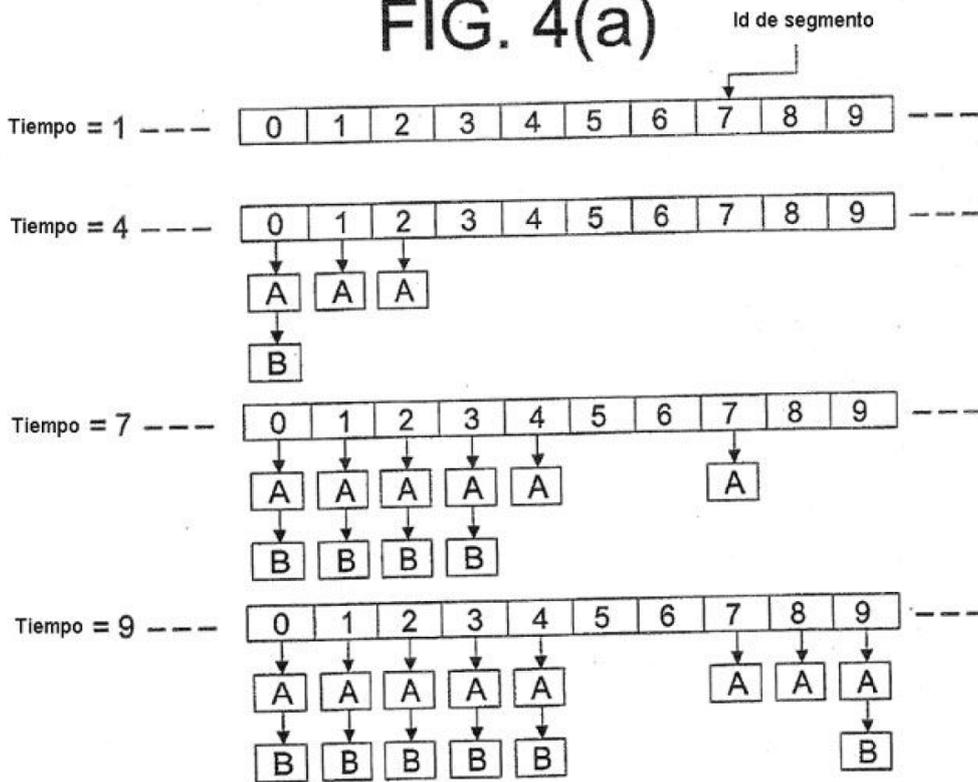


FIG. 4(b)

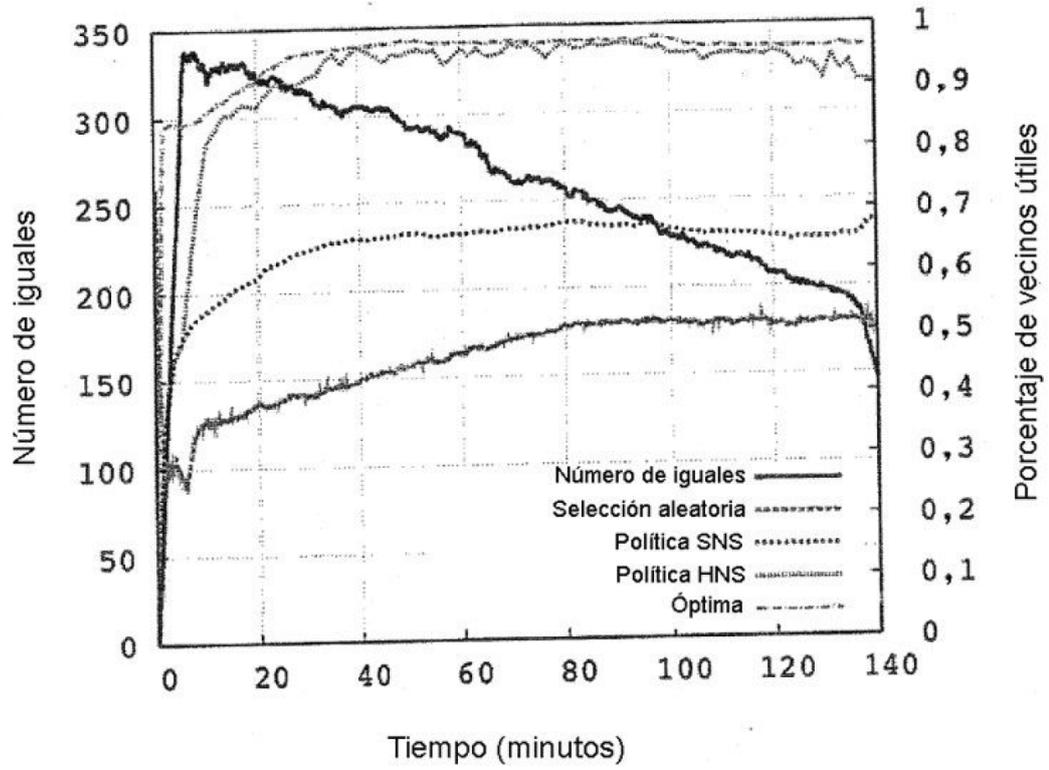


FIG. 5.