



19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

11 Número de publicación: **2 206 132**

51 Int. Cl.:
G06F 9/50 (2006.01)
G06F 17/30 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA MODIFICADA

T5

96 Número de solicitud europea: **00128346 .4**
96 Fecha de presentación : **09.02.1999**
97 Número de publicación de la solicitud: **1143337**
97 Fecha de publicación de la solicitud: **10.10.2001**

54 Título: **Localización optimizada de recursos de red.**

30 Prioridad: **10.02.1998 US 21506**

45 Fecha de publicación de la mención y de la traducción de patente europea: **16.05.2004**

45 Fecha de la publicación de la mención de la patente europea modificada BOPI: **06.10.2009**

45 Fecha de publicación de la traducción de patente europea modificada: **06.10.2009**

73 Titular/es: **Level 3 CDN International, Inc.**
1025 Eldorado Blvd.
Broomfield, Colorado 80021, US

72 Inventor/es: **Farber, David A.;**
Greer, Richard E.;
Swart, Andrew D. y
Balter, James A.

74 Agente: **Carvajal y Urquijo, Isabel**

ES 2 206 132 T5

DESCRIPCIÓN

Localización optimizada de recursos de red.

5 1. Campo de la invención

Esta invención se refiere a la replicación de recursos en redes informáticas.

10 2. Antecedentes de la invención

10 La llegada de redes informáticas globales, tales como Internet, ha conducido a formas completamente nuevas y diferentes de obtener información. Ahora, un usuario de Internet puede acceder a información desde cualquier lugar del mundo, sin tener en cuenta la verdadera localización del usuario o de la información. Un usuario puede obtener información simplemente conociendo una dirección de red para la información y proporcionando esa dirección a un programa de aplicación adecuado, tal como un navegador de red.

El rápido crecimiento en popularidad de Internet ha impuesto una pesada carga de tráfico en toda la red. Las soluciones a los problemas de la demanda (por ejemplo, mejor capacidad de acceso y enlaces de comunicación más rápidos) sólo aumentan la tensión en el suministro. Los sitios Web de Internet (a los que se hace referencia aquí como “editores”) deben soportar anchos de banda crecientes, acomodar cambios dinámicos en la carga, y mejorar el rendimiento para los clientes que navegan desde lugares distantes, especialmente los que se encuentran en el extranjero. La adopción de aplicaciones de mucho contenido, tales como vídeo y audio en directo, ha exacerbado adicionalmente el problema.

25 Para tratar las necesidades básicas del crecimiento del ancho de banda, un editor Web normalmente se suscribe a un ancho de banda adicional desde un proveedor de servicios de Internet (ISP), bien en forma de “conductos” o canales adicionales o más largos desde el proveedor de servicios de Internet hasta los establecimientos del editor, o en forma de asignaciones de ancho de banda amplio en un conjunto de servidores de alojamiento remoto del proveedor de servicios de Internet. Estos incrementos no siempre se perfeccionan de forma tan fina como las necesidades del editor, y con bastante frecuencia los tiempos de ejecución pueden hacer que la capacidad de la página Web del editor quede rezagada en relación con la demanda.

35 Para tratar problemas más serios respecto al crecimiento del ancho de banda, los editores pueden desarrollar soluciones a medida más complejas y costosas. La solución a la necesidad más común, la capacidad creciente, normalmente se basa en la replicación de recursos de hardware y del contenido de los sitios (conocido como “mirroring”, duplicación) y la duplicación de los recursos de ancho de banda. Sin embargo, estas soluciones son difíciles y costosas de utilizar y aplicar. Como resultado, sólo pueden permitírselas los editores, dado que sólo esos editores pueden amortizar los gastos mediante muchos clientes (y accesos a sitios Web).

40 Se ha desarrollado un número de soluciones para promover la replicación y la duplicación (mirroring). En general, estas tecnologías están diseñadas para que sólo las utilice un único sitio Web y no incluyen elementos que permitan que sus componentes sean compartidos por muchos sitios Web al mismo tiempo.

45 Algunos mecanismos de solución ofrecen software de replicación que ayuda a mantener actualizados los servidores duplicados. Estos mecanismos normalmente funcionan realizando una copia completa de un sistema de archivos. Un sistema de este tipo opera manteniendo, de forma transparente, múltiples copias de un sistema de archivos en sincronización. Otro sistema proporciona mecanismos para copiar, de forma explícita y regular, archivos que han cambiado. Los sistemas de bases de datos son particularmente complicados de replicar puesto que están cambiando continuamente. Varios mecanismos permiten la replicación de bases de datos, aunque no existen propuestas estándar para llevarla a cabo. Diversas empresas que ofrecen proxy-cachés los describen como herramientas de replicación. Sin embargo, los proxy-cachés difieren de éstas, puesto que se hacen funcionar por parte de los clientes, en lugar de hacerlo los editores.

55 Una vez que se haya suministrado un sitio Web por medio de múltiples servidores, un desafío consiste en garantizar que la carga se distribuya de forma apropiada o está equilibrada entre dichos servidores. La determinación de direcciones de asignación cíclica de recursos basada en el nombre de dominio del servidor hace que diferentes clientes se dirijan a diferentes réplicas.

60 Otra solución, compensar la carga, tiene en cuenta la carga de cada servidor (medida de diversas formas) para seleccionar qué servidor podría procesar una petición particular.

Los equilibradores de carga utilizan una variedad de técnicas para encaminar la petición al servidor adecuado. La mayoría de dichas técnicas de compensación de la carga requieren que cada servidor sea una réplica exacta del sitio Web primario. Los equilibradores de carga no tienen en cuenta la “distancia de red” entre el cliente y los servidores réplica candidato.

65 Suponiendo que los protocolos de los clientes no puedan cambiar fácilmente, existen dos problemas principales en el empleo de recursos duplicados. El primero de ellos es cómo seleccionar qué copia del recurso utilizar. Es decir,

cuando se realiza una petición para un recurso a un servidor individual, cómo se realizaría la elección de una réplica del servidor (o de esos datos). Denominamos a este problema “rendezvous”. Existe un número de formas de obtener clientes para rendezvous (reencontrarse) en servidores réplica distantes. Estas tecnologías, como los equilibradores de carga, deben encaminar una petición a un servidor adecuado, pero, a diferencia de los equilibradores de carga, tienen en cuenta el rendimiento y la topología de la red al realizar la determinación.

Ciertas empresas ofrecen productos que mejoran el rendimiento de red al priorizar y filtrar el tráfico de la red.

Los proxy-cachés proporcionan una forma para que los agregadores de cliente reduzcan el consumo de recursos de red almacenando copias de recursos populares cerca de los usuarios finales. Un agregador de cliente es un proveedor de servicios de Internet u otra organización que coloca en Internet un gran número de clientes que operan con navegadores. Los agregadores de cliente pueden utilizar proxy-cachés para reducir el ancho de banda necesario para suministrar contenido Web a dichos navegadores. Sin embargo, los proxy-cachés tradicionales se hacen funcionar por parte de los clientes Web, en lugar de por parte de los editores Web.

Los proxy-cachés almacenan los recursos más populares a partir de todos los editores, lo que significa que deben ser muy grandes para conseguir una eficacia razonable de la memoria caché. (La eficacia de una memoria caché se define como el número de peticiones de recursos que ya están almacenadas en la memoria caché, divididas por el número total de peticiones).

Los proxy-cachés dependen de pistas para el control de la memoria caché suministradas con los recursos para determinar cuando deberían reemplazarse los recursos. Estas pistas son indicadoras y, necesariamente, a menudo son incorrectas, de manera que los proxy-cachés con frecuencia suministran datos obsoletos. En muchos casos, los operadores de los proxy-cachés le ordenan a su proxy que ignore las pistas para hacer que la memoria caché sea más eficiente, incluso si esto ocasiona que ofrezca con mayor frecuencia datos obsoletos.

Los proxy-cachés ocultan la actividad de los clientes a los editores. Una vez que se ha almacenado un recurso en la memoria caché, el editor no tiene forma de saber con qué frecuencia se ha accedido a éste desde la memoria caché.

El documento WO 97/29423A describe la compensación de la carga a través de una red de servidores de acuerdo con la carga de los servidores.

Sumario de la invención

De acuerdo con la presente invención, se proporciona un sistema como el definido en la reivindicación 1.

De acuerdo con un segundo aspecto de la presente invención, se proporciona un método como el definido en la reivindicación 6.

Esta invención proporciona una forma para que los servidores en una red informática descarguen su procesamiento de solicitudes de recursos seleccionados, mediante determinar un servidor diferente (un “repetidor”) para procesar tales solicitudes. La selección del repetidor puede realizarse dinámicamente, en función de información sobre posibles repetidores.

Si un recurso solicitado contiene referencias a otros recursos, parte o la totalidad de estas referencias pueden ser sustituidas por referencias a repetidores.

En primer lugar, un cliente realiza una petición de un recurso particular desde un servidor origen, incluyendo la petición un localizador de recurso para el recurso concreto, incluyendo el localizador de recurso en algunos casos una indicación del servidor origen. Las peticiones que llegan al servidor origen no siempre incluyen una indicación del servidor origen; dado que se envían al servidor origen, no es necesario que lo nombren. Un mecanismo al que se hace referencia como a un reflector, dispuesto junto con el servidor origen, intercepta la petición desde el cliente al servidor origen y decide si reflejar la petición o procesarla de forma local. Si el reflector decide procesar la petición de forma local, la envía al servidor origen; de lo contrario, selecciona el “mejor” repetidor para procesar la petición. Si la petición se refleja, se le proporciona al cliente un localizador de recursos modificado que designa al repetidor.

El cliente obtiene el localizador de recursos modificado desde el reflector y realiza una petición para el recurso particular desde el repetidor designado en el localizador de recursos modificado.

Cuando el repetidor obtiene la petición del cliente, éste responde presentándole al cliente el recurso solicitado. Si el repetidor tiene una copia local del recurso, entonces muestra esa copia; de lo contrario, envía la petición al servidor origen para obtener el recurso, y guarda una copia local del recurso para suministrar peticiones posteriores.

La selección por parte del reflector de un repetidor adecuado para procesar la petición puede realizarse de varias formas. En la realización preferida, se realiza dividiendo previamente la red en “grupos de coste” y luego determinando en qué grupo de coste se encuentra el cliente. A continuación, se selecciona un conjunto de repetidores a partir de una pluralidad de repetidores de la red, de modo que los miembros del conjunto tienen un bajo coste en relación con el grupo de coste en el que se encuentra el cliente. Para determinar el coste más bajo, se guarda una tabla y se actualiza

regularmente para definir el coste entre cada grupo y cada repetidor. Entonces, se selecciona un miembro del conjunto, preferiblemente de forma aleatoria, como el mejor repetidor.

Si el recurso particular solicitado puede contener en sí mismo localizadores de los otros recursos, entonces el recurso puede describirse (antes de proporcionársele al cliente). En particular, el recurso se describe para reemplazar al menos algunos de los localizadores de recursos contenidos en él por localizadores de recursos modificados que hacen referencia a un repetidor, en lugar de hacer referencia al servidor origen. Como una consecuencia de este proceso de reescritura, cuando el cliente solicita otros recursos basándose en localizadores del recurso particular solicitado, el cliente realizará esas peticiones directamente al repetidor seleccionado, evitando por completo al reflector y al servidor origen.

La reescritura de los recursos debe realizarse mediante reflectores. También puede llevarse a cabo por medio de repetidores, en la situación en la que los repetidores “se escrutan” entre sí y realizan copias de los recursos que incluyen localizadores de recursos rescritos que hacen referencia a un repetidor.

En una realización preferida, la red es Internet y el localizador de recursos es un Localizador de Recursos Uniforme (URL) para designar recursos en Internet, y el localizador de recursos modificado es un localizador URL que hace referencia al repetidor y que indica el servidor origen (como se indica posteriormente en el paso B3), y al cliente se le proporciona el localizador de recursos modificado mediante un mensaje REDIRECT. Obsérvese que sólo cuando el reflector está “reflejando” una petición se suministra el localizador de recursos modificado mediante un mensaje REDIRECT.

Breve descripción de los dibujos

Los objetos y ventajas anteriores y otros resultarán evidentes al considerar la siguiente descripción detallada, tomada en combinación con los dibujos adjuntos, en los que los caracteres de referencia hacen referencia a partes iguales en todos ellos y en los que:

La figura 1 muestra una parte de un entorno de red según la presente invención; y
las figuras 2-6 son diagramas de flujo del funcionamiento de la presente invención.

Descripción detallada de las realizaciones a modo de ejemplo preferidas actualmente

Perspectiva general

La figura 1 muestra una parte de un entorno 100 de red según la presente invención, en el que un mecanismo (reflector 108, descrito posteriormente de forma detallada) en un servidor (aquí, servidor 102 origen) mantiene y sigue la pista de un número de servidores o repetidores 104a, 104b, y 104c parcialmente duplicados. Cada repetidor 104a, 104b y 104c duplica algo de o toda la información disponible en el servidor 102 origen, así como la información disponible en otros servidores origen de la red 100. El reflector 108 está conectado a un repetidor particular, conocido como su repetidor “de contacto” (“repetidor B” 104b, en el sistema mostrado en la figura 1). Preferiblemente, cada reflector está conectado con un único repetidor, conocido como su contacto, y cada repetidor está conectado con un repetidor especial, conocido como su repetidor maestro (por ejemplo, el repetidor 104m para los repetidores 104a, 104b y 104c de la figura 1).

De esta manera, puede considerarse un repetidor como un servidor proxy dedicado que mantiene una réplica parcial o escasa del servidor 102 origen, mediante la implementación de una memoria caché coherente distribuida del servidor origen. Un repetidor puede mantener una réplica (parcial) de más de un servidor origen. En algunas realizaciones, la red 100 es Internet y los repetidores duplican recursos seleccionados proporcionados por servidores origen como respuesta a peticiones http (Protocolo de Transferencia de Hipertexto) y FTP (Protocolo de Transferencia de Archivos) de clientes.

Un cliente 106 conecta, a través de la red 100, con un servidor 102 origen y, posiblemente, con uno o más repetidores 104a, etc.

El servidor 102 origen es un servidor en el que se originan los recursos. De forma más general, el servidor 102 origen es cualquier proceso o grupo de procesos que proporciona recursos como respuesta a peticiones de un cliente 106. El servidor 102 origen puede ser cualquier servidor Web ya creado. En una realización preferida, el servidor 102 origen es normalmente un servidor Web tal como el servidor Apache o el servidor Netscape Communications Corporation's Enterprise™.

El cliente 106 es un procesador que solicita recursos a partir del servidor 102 origen dirigidos a un usuario final. Normalmente, el cliente 106 es un agente de usuario (por ejemplo, un navegador Web tal como Netscape Communications Corporation's Navigator™) o un proxy para un agente de usuario. Los componentes distintos del reflector 108 y de los repetidores 104a, 104b, etc., pueden implementarse utilizando programas de software disponibles de forma habitual. En concreto, esta invención trabaja con cualquier cliente http (por ejemplo, un navegador Web), proxy-caché, y servidor Web. Además, el reflector 108 podría estar completamente integrado en el servidor 112 de datos (por

ejemplo, en un servidor Web). Estos componentes podrían estar integrados de forma flexible, basándose en el uso de mecanismos de extensión (tales como los denominados módulos de adición), o integrados de forma fija, mediante la modificación del componente de servicio de forma específica para soportar los repetidores.

5 Los recursos que se originan en el servidor 102 origen pueden ser estáticos o dinámicos. Es decir, los recursos pueden fijarse o crearse por el servidor 102 origen específicamente como respuesta a una petición. Obsérvese que los términos “estático” y “dinámico” son relativos, puesto que un recurso estático puede cambiar a algún intervalo regular, aunque largo.

10 Las peticiones de recursos desde el cliente 106 al servidor 102 origen son interceptadas por el reflector 108, que, para una determinada petición, o bien envía la petición al servidor 102 origen o, de forma condicional, la refleja a algún repetidor 104a, 104b, etc. de la red 100. Es decir, en función de la naturaleza de la petición del cliente 106 al servidor 102 origen, el reflector 108 o bien suministra la petición de forma local (en el servidor 102 origen), o selecciona uno de los repetidores (preferiblemente el mejor repetidor para el trabajo) y refleja la petición al repetidor
15 seleccionado. En otras palabras, el reflector 108 hace que las peticiones de recursos desde el servidor 102 origen, realizadas por el cliente 106, o bien se suministren localmente por medio del servidor 102 origen o bien se reflejen de forma transparente al mejor repetidor 104a, 104b, etc. La noción del mejor repetidor y la manera en que se selecciona el mejor repetidor se describen posteriormente de forma detallada.

20 Los repetidores 104a, 104b, etc. son procesadores intermedios utilizados para suministrar peticiones de clientes, mejorando con ello el rendimiento y reduciendo los costes de la forma aquí descrita. Dentro de los repetidores 104a, 104b, etc., se encuentra cualquier proceso o conjunto de procesos que suministre recursos al cliente 106 por parte del servidor 102 origen. Un repetidor puede incluir una memoria 110 caché de repetición, utilizada para evitar transacciones innecesarias con el servidor 102 origen.

25 El reflector 108 es un mecanismo, preferiblemente un programa de software, que intercepta peticiones que normalmente se enviarían directamente al servidor 102 origen. A pesar de que en los dibujos se muestran como componentes separados, el reflector 108 y el servidor 102 origen normalmente están situados juntos, por ejemplo, en un sistema particular, tal como un servidor 112 de datos. (Como se explicará posteriormente, el reflector 108 puede ser incluso
30 un módulo “de conexión” que forma parte del servidor 102 origen).

La figura 1 sólo muestra una parte de una red 100 según esta invención. Una red operativa completa se compone de cualquier número de clientes, repetidores, reflectores, y servidores origen. Los reflectores se comunican con la red de repetidores, y los repetidores de la red se comunican entre sí.

35 *Localizadores de Recursos Uniformes*

Cada localización en una red informática tiene una dirección que normalmente puede especificarse como una serie de nombres o números. Para acceder a la información debe conocerse una dirección para esa información. Por ejemplo, en la World Wide Web (“la Web”), que es un subconjunto de Internet, la forma en que se proporcionan localizaciones de direcciones de información se ha normalizado en Localizadores de Recursos Uniformes (URL). Los localizadores URL especifican la localización de los recursos (información, archivos de datos, etc.) en la red.

La noción de los localizadores URL resulta incluso más útil cuando se utilizan documentos de hipertexto. Un documento de hipertexto es un documento que incluye, dentro del propio documento, vínculos (indicadores o referencias) al propio documento o a otros documentos. Por ejemplo, en un sistema de búsqueda legal en línea, cada caso puede estar presentado como un documento de hipertexto. Al citar otros casos, pueden proporcionarse vínculos a dichos casos. De esta manera, si una persona esta leyendo un caso, puede seguir los vínculos citados para leer las partes apropiadas de los casos citados.

50 En el caso de Internet en general y de la World Wide Web en particular, los documentos pueden crearse mediante una forma normalizada conocida como el Lenguaje de Marcado de Hipertexto (HTML). En el lenguaje HTML, un documento se compone de datos (texto, imágenes, sonidos, y similares), incluyendo vínculos a otras secciones del mismo documento o a otros documentos. Los vínculos normalmente se proporcionan como localizadores URL, y
55 pueden estar en forma absoluta o en forma relativa. Los localizadores URL relativos simplemente omiten las partes del localizador URL que son las mismas que para el documento que incluye el vínculo, tal como la dirección del documento (cuando está vinculado al mismo documento), etc. En general, un programa del navegador completará las partes omitidas de un localizador URL utilizando las partes correspondientes del documento actual, formando con ello un localizador URL completamente formado que incluye un nombre de dominio totalmente capacitado,
60 etc.

Un documento de hipertexto puede contener cualquier número de vínculos a otros documentos, y cada uno de esos otros documentos puede estar en un servidor diferente en una parte diferente del mundo. Por ejemplo, un documento puede contener vínculos a documentos en Rusia, África, China y Australia. Un usuario que visualiza ese documento
65 en un cliente particular puede seguir cualquiera de los vínculos de forma transparente (es decir, sin saber dónde se encuentra realmente el documento a cuyo vínculo se está accediendo). De forma correspondiente, el coste (en términos de tiempo, dinero o asignación de recursos) de seguir un vínculo frente a otro puede ser bastante significativo.

ES 2 206 132 T5

Los localizadores URL normalmente tienen la siguiente forma (definida de forma detallada en T. Berners-Lee *et al.*, Uniform Resource Locators (URL), Network Working Group, Request for Comments: 1738, Category: Standards Track, Diciembre de 1994, localizado en

“http://ds.internic.net/rfc/rfc1738.txt”):

esquema://anfitrión[:puerto]/ruta-url

donde “esquema” puede ser un símbolo tal como “archivo” (para un archivo del sistema local), “ftp” (para un archivo en un servidor de archivos FTP anónimo), “http” (para un archivo en un servidor Web), y “telnet” (para una conexión de un servicio basado en Telnet). También pueden utilizarse otros esquemas y actualmente y en el futuro siguen añadiéndose nuevos esquemas. El número de puerto es opcional, de modo que si no se proporciona ninguno el sistema sustituye un número de puerto por defecto (en función del esquema). El campo “anfitrión” representa una dirección de red particular para un ordenador particular. La “ruta-url” está relacionada con el ordenador especificado en el campo “anfitrión”. Una ruta-url es normalmente, aunque no necesariamente, el nombre de ruta de un archivo en un directorio del servidor Web.

Por ejemplo, el siguiente es un archivo “F” que identifica un localizador URL en la ruta “A/B/C” de un ordenador en “www.uspto.gov”:

http://www.uspto.gov/A/B/C/F

Para acceder al archivo “F” (el recurso) especificado por el localizador URL anterior, un programa (por ejemplo, un navegador) que se ejecuta en un ordenador del usuario (es decir, un ordenador cliente) primero tendría que localizar el ordenador (es decir, un ordenador servidor) especificado por el nombre de anfitrión, es decir, el programa tendría que localizar al servidor “www.uspto.gov”. Para hacer esto, accedería a un Servidor de Nombres de Dominio (DNS), proporcionándole al DNS el nombre del anfitrión (“www.uspto.gov”). El DNS actúa como un tipo de directorio centralizado para definir direcciones a partir de nombres. Si el DNS determina que existe un ordenador (servidor remoto) que corresponde al nombre “www.uspto.gov”, le proporcionará al programa una dirección real de la red informática para ese ordenador servidor. En Internet esto se denomina una dirección de Protocolo de Internet (o IP) y tiene la forma “123.345.456.678”. El programa en el ordenador del usuario (cliente) utilizaría entonces la verdadera dirección para acceder al ordenador remoto (servidor).

El programa abre una conexión al servidor HTTP (servidor Web) en el ordenador remoto “www.uspto.gov” y utiliza la conexión para enviar un mensaje de petición al ordenador remoto (empleando el esquema HTTP). Normalmente, el mensaje es una petición HTTP GET que incluye la ruta URL del recurso solicitado, “A/B/C/F”. El servidor HTTP recibe la petición y la utiliza para acceder al recurso especificado por la ruta URL “A/B/C/F”. El servidor presenta el recurso por la misma conexión.

De esta manera, convencionalmente las peticiones de cliente HTTP para recursos Web en un servidor 102 origen se procesan de la siguiente manera (véase la figura 2). (Ésta es una descripción del proceso cuando no hay instalado ningún reflector 108):

- A1. Un navegador (por ejemplo, Netscape’s Navigator) en el cliente recibe un localizador de recursos (es decir, un localizador URL) desde un usuario.
- A2. El navegador extrae el nombre del anfitrión (servidor origen) a partir del localizador de recursos, y utiliza un servidor de nombres de dominio (DNS) para buscar la dirección de red (IP) del servidor correspondiente. El navegador también extrae un número de puerto, si hay uno presente, o utiliza un número de puerto por defecto (el número de puerto por defecto para las peticiones http es 80).
- A3. El navegador utiliza la dirección de red del servidor y el número de puerto para establecer una conexión entre el cliente 106 y el anfitrión o servidor 102 origen.
- A4. Entonces, el cliente 106 envía una petición (GET) por la conexión que identifica al recurso solicitado.
- A5. El servidor 102 origen recibe la petición y
- A6. localiza o compone el recurso correspondiente.
- A7. El servidor 102 origen envía de vuelta entonces al cliente 106 una respuesta que contiene el recurso solicitado (o alguna forma de indicación de error si el recurso no está disponible). La respuesta se le envía al cliente por la misma conexión por la que se recibió la petición del cliente.
- A8. El cliente 106 recibe la respuesta desde el servidor 102 origen.

ES 2 206 132 T5

Existen muchas variaciones de este modelo básico. Por ejemplo, en una variación, en lugar de proporcionarle el recurso al cliente, el servidor origen puede solicitarle al cliente que vuelva a solicitar el recurso mediante otro nombre. Para hacerlo, en el paso A7, el servidor 102 envía de vuelta al cliente 106 una respuesta denominada un “REDIRECT”, que contiene un nuevo localizador URL que indica el otro nombre. El cliente 106 repite entonces toda la secuencia, normalmente sin ninguna intervención del usuario, solicitando esta vez el recurso identificado por el nuevo localizador URL.

Funcionamiento del sistema

En esta invención, el reflector 108 reemplaza eficazmente a un servidor Web ordinario o al servidor 102 origen. El reflector 108 hace esto tomando la dirección IP del servidor origen y el número de puerto. De esta manera, cuando un cliente trata de conectar con el servidor 102 origen, realmente conectará con el reflector 108. El servidor Web original (o servidor 102 origen) debe aceptar entonces las peticiones en una dirección de red (IP) diferente, o en la misma dirección IP, pero en un número de puerto diferente. De esta manera, mediante esta invención, el servidor al que se hace referencia en los pasos A3-A7 anteriores es realmente un reflector 108.

Obsérvese que también es posible dejar la dirección de red del servidor origen tal como es y permitir que el reflector se ejecute en una dirección diferente o en un puerto diferente. De esta manera, el reflector no intercepta las peticiones enviadas al servidor origen, pero aún se le pueden enviar peticiones dirigidas específicamente al reflector. De este modo, puede comprobarse y configurarse el sistema sin interrumpir su funcionamiento normal.

El reflector 108 soporta el procesamiento de la siguiente manera (véase la figura 3):

Al recibir una petición,

B1. El reflector 108 analiza la petición para determinar si reflejar o no la petición. Para hacer esto, el reflector determina primero si el emisor (cliente 106) es un navegador o un repetidor. Las peticiones formuladas por los repetidores deben suministrarse de forma local por el servidor 102 origen. Esta determinación puede realizarse buscando la dirección de red (IP) del emisor en una lista de direcciones de red (IP) de repetidores conocida. De forma alternativa, esta determinación podría realizarse adjuntando información a una petición para indicar que la petición es de un repetidor específico, o los repetidores pueden solicitar recursos desde un puerto especial diferente del utilizado para los clientes ordinarios.

B2. Si la petición no procede de un repetidor, el reflector busca el recurso solicitado en una tabla (denominada la “base de reglas”) para determinar si el recurso solicitado puede “repetirse”. Basándose en esta determinación, el reflector o bien refleja la petición (paso B3 descrito a continuación) o suministra la petición de forma local (paso B4 descrito a continuación). La base de reglas es una lista de expresiones regulares y atributos asociados. (Las expresiones regulares son bien conocidas en el área de la ciencia informática. Una pequeña bibliografía sobre su uso se encuentra en *Abo, et al.*, “Compilers, Principles, techniques and tools”, Addison-Wesley, 1986, pp. 157-158): El localizador de recursos (URL) para una petición determinada se busca en la base de reglas haciéndolo corresponder secuencialmente con cada una de las expresiones regulares. La primera correspondencia identifica los atributos para el recurso, concretamente, si puede repetirse o es local. Si no se produce ninguna correspondencia en la base de reglas, se utiliza un atributo determinado por defecto. Cada reflector tiene su propia base de reglas, que se configura de forma manual por medio del operador del reflector.

B3. Para reflejar una petición, (para suministrar una petición de forma local, diríjase al paso B4), como se muestra en la figura 4, el reflector determina (pasos B3-1) el mejor repetidor para reflejar la petición, como se describe posteriormente de forma detallada. Entonces, el reflector crea (pasos B3-2) un nuevo localizador de recursos (URL) (utilizando el localizador URL solicitado y el mejor repetidor) que identifica el mismo recurso en el repetidor seleccionado.

Es necesario que el paso de reflexión cree un único localizador URL que contenga el localizador URL del recurso original, así como la identidad del repetidor seleccionado. Se crea una forma especial de localizador URL para proporcionar esta información. Esto se hace creando un nuevo localizador URL de la siguiente manera:

D1. Dado un nombre de repetidor, esquema, nombre del servidor origen y ruta, crear un nuevo localizador URL. Si el esquema es “http”, la realización preferida utiliza el siguiente formato:

`http://<repetidor>/<servidor>/<ruta>`

Si la forma empleada es diferente del esquema “http”, la realización preferida emplea el siguiente formato:

`http://<repetidor>/<servidor>@proxy=<esquema>@/<ruta>`

ES 2 206 132 T5

El reflector también puede adjuntar un tipo MIME a la petición, para hacer que el repetidor proporcione el resultado a ese tipo MIME. Esto es útil puesto que muchos protocolos (tales como el protocolo FTP) no proporcionan una forma de adjuntar un tipo MIME a un recurso. El formato es

5 `http://<repetidor>/<servidor>@proxy=<esquema>:<tipo>@/<ruta>`

Este localizador URL se interpreta al ser recibido por el repetidor.

Entonces el reflector envía (pasos B3-3) al cliente solicitante una respuesta REDIRECT que contiene este nuevo localizador URL. El comando HTTP REDIRECT permite al reflector enviarle al navegador un único localizador URL para recuperar la petición.

10 B4. Para suministrar una petición de forma local, la petición se emite por medio del reflector ("se envía") al servidor 102 origen. De este modo, el reflector actúa como un servidor proxy inverso. El servidor 102 origen procesa la petición de la forma normal (pasos A5-A7). Entonces, el reflector obtiene la respuesta del servidor origen a la petición, que éste examina para determinar si el recurso solicitado es un documento HTML, es decir, si el recurso solicitado es uno que contiene en sí mismo localizadores de recursos.

15 B5. Si el recurso es un documento HTML, entonces el reflector rescribe el documento HTML modificando los localizadores de recursos (localizadores URL) dentro de éste, como se describirá posteriormente. El recurso, posiblemente según se modifica al rescribirlo, se le devuelve entonces en una respuesta al cliente 106 solicitante.

20 Si el cliente solicitante es un repetidor, el reflector puede inutilizar temporalmente cualquier modificador de control de la memoria caché que el servidor origen haya adjuntado a la respuesta. Estos modificadores de control de la memoria caché inutilizados luego se vuelven a habilitar cuando se haya suministrado el contenido desde el repetidor. Este mecanismo posibilita que el servidor origen impida que los recursos se almacenen en proxy-cachés normales, sin que se vea afectado el comportamiento de la memoria caché en el repetidor.

25 B6. El reflector escribe en un archivo de registro local si la petición es reflejada o procesada de forma local, detalles sobre la transacción, tales como el tiempo actual, la dirección del solicitante, el localizador URL solicitado, y el tipo de respuesta generada.

30 Utilizando una base de reglas (paso B2), es posible reflejar los recursos de forma selectiva. Existen ciertas razones por las que determinados recursos concretos no pueden repetirse eficazmente (y, por tanto, no deberían reflejarse), por ejemplo:

- 35 - el recurso sólo se compone para cada petición;
- 40 - el recurso se basa en un denominado cookie (los navegadores no enviarán cookies a repetidores con diferentes nombres de dominio);
- 45 - el recurso es en realidad un programa (tal como Java applet) que se ejecutará en el cliente y que desea conectarse a un servicio (Java requiere que el servicio se esté ejecutando en la misma máquina que proporcionó el applet).

50 Además, el reflector 108 puede configurarse de manera que las peticiones desde ciertas direcciones de red (por ejemplo, peticiones de clientes en la misma red de área local que el propio reflector) nunca se reflejen. Asimismo, el reflector puede elegir no reflejar peticiones dado que el reflector está sobrepasando su velocidad de transferencia de información agregada asignada, según se describirá posteriormente.

55 Una petición que se refleja se duplica de forma automática en el repetidor cuando el repetidor recibe y procesa la petición.

La combinación del proceso de reflexión descrita aquí y el proceso de almacenamiento en la memoria caché descrito posteriormente, crean eficazmente un sistema en el que los recursos repetibles se trasladan al reflector seleccionado y se duplican en éste, mientras que los recursos no repetibles no se duplican.

Enfoque alternativo

60 Generalmente, colocar el nombre del servidor origen en el localizador URL reflejado es una buena estrategia, aunque puede considerarse poco deseable por razones estéticas (por ejemplo, en el caso de los cookies) o por ciertas razones técnicas.

ES 2 206 132 T5

Es posible evitar la necesidad de tener que introducir tanto el nombre del repetidor como el nombre del servidor en el localizador URL. En lugar de ello, puede crearse una “familia” de nombres para un determinado servidor origen, identificando cada nombre a uno de los repetidores utilizados por ese servidor.

Por ejemplo, si `www.ejemplo.com` es el servidor origen, podrían crearse nombres para tres repetidores:

`wr1.ejemplo.com`

`wr2.ejemplo.com`

`wr3.ejemplo.com`

El nombre “`wr1.ejemplo.com`” sería un alias para el repetidor 1, que también podría ser conocido por otros nombres, tales como “`wr1.otroejemplo.com`” y “`wr1.ejemplo.edu`”.

Si el repetidor puede determinar con qué nombre se han dirigido a él, puede utilizar esta información (junto con una tabla que asocia los alias del repetidor con los nombres del servidor origen) para determinar a qué servidor origen está dirigiéndose. Por ejemplo, si se dirige al repetidor 1 como `wr1.ejemplo.com`, entonces el servidor origen es “`www.ejemplo.com`”; si se dirige a éste como “`wr1.otroejemplo.com`”, entonces el servidor origen es “`www.otroejemplo.com`”.

El repetidor puede utilizar dos mecanismos para determinar con qué alias se dirigen a él:

1. Cada alias puede estar asociado con una dirección IP diferente. Desafortunadamente, esta solución no es apropiada, puesto que las direcciones IP son comúnmente escasas, y el número de direcciones IP requerido crece según el producto de servidores origen y repetidores.
2. El repetidor puede intentar determinar el alias utilizado examinando la etiqueta del “anfitrión:” en la cabecera HTTP de la petición. Desafortunadamente, algunos navegadores antiguos aún en uso no adjuntan la etiqueta del “anfitrión:” a la petición. Los reflectores necesitarían identificar este tipo de navegadores (la identidad del navegador es parte de cada petición) y evitar esta forma de reflexión.

Cómo procesa una petición un repetidor

Cuando un navegador recibe una respuesta REDIRECT (según se produce en el paso B3), éste vuelve a formular una petición para el recurso utilizando el nuevo localizador de recursos (localizador URL) (pasos A1-.A5). Dado que el nuevo localizador hace referencia a un repetidor, en lugar de al servidor origen, el navegador envía ahora una petición para el recurso al repetidor, que procesa una petición de la siguiente manera, en relación con la figura 5:

C1. Primero el repetidor analiza la petición para determinar la dirección de red del cliente solicitante y la ruta del recurso solicitado. En la ruta está incluido un nombre de servidor origen (como se describió anteriormente en relación con el paso B3).

C2. El repetidor utiliza una tabla interna para verificar que el servidor origen pertenece a un “abonado” conocido. Un abonado es una entidad (por ejemplo, una empresa) que publica recursos (por ejemplo, archivos) a través de uno o varios servidores origen. Cuando la entidad se suscribe, está permitido utilizar la red de repetidores. Las tablas de abonados descritas posteriormente incluyen la información que se utiliza para vincular los reflectores a los abonados.

Si la petición no es para un recurso de un abonado conocido, se deniega la petición. Para denegar una petición, el repetidor muestra una respuesta que indica que el recurso solicitado no existe.

C3. Entonces, el repetidor determina si el recurso solicitado está almacenado localmente en la memoria caché. Si el recurso solicitado está en la memoria caché del repetidor, se recupera. De otro modo, si no hay una copia válida del recurso solicitado en la memoria caché del repetidor, el repetidor modifica el localizador URL entrante, creando una petición que formula directamente al reflector que la origina, el cual la procesa (como en los pasos B1-B6). Dado que esta petición al reflector que la origina procede de un repetidor, el reflector siempre muestra el recurso solicitado, en lugar de reflejar la petición. (Recuérdese que los reflectores siempre procesan las peticiones de los repetidores de forma local). Si el repetidor obtuvo el recurso del servidor origen, entonces el repetidor almacena el recurso localmente en la memoria caché.

Si un recurso no está almacenado localmente en la memoria caché, la memoria caché puede consultar sus “memorias cachés de igual nivel” para ver si alguna de ellas contiene el recurso, antes o al mismo tiempo que solicita el recurso a partir del reflector/servidor origen. Si una memoria caché del mismo nivel responde de forma positiva en un periodo de tiempo limitado (preferiblemente una pequeña fracción de un segundo), el recurso será recuperado de la memoria caché de igual nivel.

ES 2 206 132 T5

C4. Entonces, el repetidor construye una respuesta que incluye el recurso solicitado (que fue recuperado de la memoria caché o del servidor origen) y envía esa respuesta al cliente solicitante.

C5. En un archivo de registro local del repetidor se escriben los detalles sobre la transacción, tales como el reflector asociado, el tiempo actual, la dirección del solicitante, el localizador URL solicitado y el tipo de respuesta generada.

Obsérvese que la fila inferior de la figura 2 se refiere a un servidor origen, o a un reflector, o un repetidor, dependiendo de qué identifique el localizador URL en el paso A1.

Seleccionar el mejor repetidor

Si el reflector 108 determina que reflejará la petición, entonces debe seleccionar el mejor repetidor para procesar esa petición (como se hace referencia en los pasos B3-1). Esta selección la realiza el mecanismo Selector del Mejor Repetidor (BRS) descrito aquí.

El objetivo del mecanismo BRS es seleccionar, de forma rápida y heurística, un repetidor apropiado para un cliente determinado dada únicamente la dirección de red del cliente. Un repetidor apropiado es un repetidor que no está demasiado cargado y que no está demasiado lejos del cliente en términos de distancia de red. El mecanismo utilizado aquí se basa en datos específicos, compactos y calculados previamente para tomar una decisión rápida. También pueden emplearse otras soluciones dinámicas para seleccionar un repetidor adecuado.

El selector BRS se basa en tres tablas calculadas previamente, concretamente, la Tabla de Reducción de Grupos, la Tabla de Costes de los Vínculos y la Tabla de Carga. Estas tres tablas (descritas posteriormente) se calculan fuera de línea y se descargan en cada reflector mediante su contacto en la red de repetidores.

La Tabla de Reducción de Grupos coloca cada dirección de red en un grupo, con el objetivo de que las direcciones de un grupo compartan gastos relacionados, de manera que tendrían el mismo mejor repetidor en condiciones cambiantes (es decir, el selector BRS se mantiene sin cambios en los miembros del grupo).

La Tabla de Costes de los Vínculos es una matriz bidimensional que especifica el coste actual entre cada repetidor y cada grupo. Inicialmente, el coste del vínculo entre un repetidor y un grupo se define como el “coste normalizado del vínculo” entre el repetidor y el grupo, como se definirá posteriormente. A lo largo del tiempo, se actualizará la tabla con las mediciones que reflejan de forma más precisa el coste relativo de transmitir un archivo entre el repetidor y un miembro del grupo. El formato de la Tabla de Costes de los Vínculos es <Identificador del grupo><Identificador del grupo><coste del vínculo>, donde los identificadores de grupo se proporcionan como números de sistema autónomo (ASN).

La Tabla de Carga es una tabla unidimensional que identifica la carga actual en cada repetidor. Dado que los repetidores pueden tener diferentes capacidades, la carga es un valor que representa la capacidad de un determinado receptor de aceptar trabajo adicional. Cada repetidor envía su carga actual a un repetidor maestro central a intervalos regulares, preferiblemente, al menos aproximadamente una vez por minuto. El repetidor maestro difunde la Tabla de carga a cada reflector de la red, a través del repetidor de contacto.

A un reflector sólo se le proporcionan entradas en la Tabla de Carga para los repetidores a los que está asignado a utilizar. La asignación de los repetidores a los reflectores se realiza de forma centralizada en el repetidor maestro mediante un operador de la red de repetidores. Esta asignación hace posible modificar el nivel de servicio de un determinado reflector. Por ejemplo, un reflector muy activo puede utilizar muchos repetidores, mientras que un reflector relativamente inactivo puede utilizar pocos repetidores.

Las tablas también pueden configurarse para proporcionar un servicio de repetidor selectivo a los abonados de otras formas, por ejemplo, para sus clientes en regiones geográficas específicas, tales como Europa o Asia.

Medición de la carga

En las realizaciones preferidas actualmente, la carga de los repetidores se mide en dos dimensiones, concretamente,

1. peticiones recibidas por el repetidor por intervalo de tiempo (RRPT), y
2. bytes enviados por el repetidor por intervalo de tiempo (BSPT).

Para cada una de estas dimensiones se configura un ajuste de capacidad máxima. La capacidad máxima indica el punto en el que se considera que el repetidor está completamente cargado. Una capacidad RRPT mayor generalmente indica un procesador más rápido, mientras que una capacidad BSPT mayor generalmente indica un cable de red más ancho. Esta forma de medición de la carga supone que un determinado servidor se dedica a la tarea de la repetición.

ES 2 206 132 T5

Cada repetidor calcula regularmente sus capacidades RRPT y BSPT actuales acumulando el número de peticiones recibidas y los bytes enviados durante un corto intervalo de tiempo. Estas mediciones se utilizan para determinar la carga del repetidor en cada una de estas dimensiones. Si la carga de un repetidor excede de su capacidad configurada, se envía un mensaje de alarma al administrador de la red de repetidores.

Los dos componentes actuales de la carga se combinan en un único valor que indica la carga actual general. De forma similar, los dos componentes de capacidad máxima se combinan en un único valor que indica la capacidad máxima general. Los componentes se combinan de la siguiente manera:

$$\text{carga-actual} = B \times \text{RRPT actual} + (1 - B) \times \text{BSPT actual}$$

$$\text{carga-máxima} = B \times \text{RRPT máxima} + (1 - B) \times \text{BSPT máxima}.$$

El factor B, un valor entre 0 y 1, permite que se ajusten los pesos relativos de las capacidades RRPT y BSPT, lo que favorece la consideración de la potencia de procesamiento o del ancho de banda.

Los valores de la carga actual general y de la capacidad máxima general se envían periódicamente desde cada repetidor al repetidor maestro, donde se agregan a la Tabla de Carga, resumiendo una tabla la carga general para todos los repetidores. Los cambios en la Tabla de Carga se distribuyen automáticamente a cada reflector.

Aunque la realización preferida utiliza una medida bidimensional de la carga del repetidor, puede utilizarse cualquier otra medida de carga.

Combinar los Costes de los Vínculos y la Carga

El selector BRS calcula el coste de proporcionar servicio a un cliente determinado desde cada uno de los repetidores que pueden elegirse. El coste se calcula combinando la capacidad disponible del repetidor candidato con el coste del vínculo entre el repetidor y el cliente. El coste del vínculo se calcula simplemente buscándolo en la tabla de costes de vínculos.

El coste se determina empleando la siguiente fórmula:

$$\text{umbral} = K * \text{carga-máxima}$$

$$\text{capacidad} = \text{máximo} (\text{carga-máxima} - \text{carga-actual}, e)$$

$$\text{capacidad} = \text{mínimo} (\text{capacidad}, \text{umbral})$$

$$\text{coste} = \text{coste-vínculo} * \text{umbral/capacidad}.$$

En esta fórmula, e es un número muy pequeño (épsilon) y K es un ajuste inicial del factor de sincronización a 0,5. Esta fórmula hace que se incremente el coste para un determinado repetidor, en una proporción determinada por K, si su capacidad desciende por debajo de un umbral regulable.

Dado el coste de cada repetidor candidato, el selector BRS selecciona todos los repetidores dentro de un factor delta de la mejor puntuación. A partir de este conjunto, se selecciona el resultado de forma aleatoria.

El factor delta impide que el selector BRS seleccione repetidamente un único repetidor cuando las puntuaciones son similares. Normalmente esto es necesario dado que la información disponible sobre la carga y los costes de los vínculos pierde precisión a lo largo del tiempo. Este factor no puede sintonizarse.

Selector del Mejor Repetidor (BRS)

El selector BRS funciona de la siguiente manera, en relación con la figura 6:

Dada una dirección de red de cliente y las tres tablas descritas anteriormente:

E1. Determina en qué grupo está el cliente utilizando la Tabla de Reducción de Grupos.

E2. Determina, para cada repetidor en la Tabla de Costes de los Vínculos y en la Tabla de Carga, el coste combinado de ese repetidor de la siguiente manera:

E2a. Determina la carga máxima y la carga actual en el repetidor (utilizando la Tabla de Carga).

E2b. Determina el coste del vínculo entre el repetidor y el grupo del cliente (utilizando la Tabla de Costes de los Vínculos).

ES 2 206 132 T5

E2c. Determina el coste combinado como se describió anteriormente.

E3. Selecciona un pequeño conjunto de repetidores con el coste más bajo.

5 E4. Selecciona un elemento aleatorio del conjunto.

10 Preferiblemente, los resultados del procesamiento del selector BRS se mantienen en una memoria caché local en el reflector 108. De esta manera, si se ha determinado recientemente el mejor repetidor para un cliente determinado (es decir, para una determinada dirección de red), ese mejor repetidor puede volverse a utilizar rápidamente sin haberlo determinado nuevamente. Puesto que el cálculo anteriormente descrito se basa en tablas calculadas previamente de forma estática, si las tablas no han cambiado, entonces no hay necesidad de volver a determinar el mejor repetidor.

15 *Determinar las Tablas de Reducción de Grupos y de Costes de los Vínculos*

La Tabla de Reducción de Grupos y la Tabla de Costes de los Vínculos utilizadas en el procesamiento del selector BRS se crean y actualizan regularmente mediante un procedimiento independiente al que se hace referencia aquí como NetMap (Gestión de Red). El procedimiento NetMap se inicia al ejecutar varias fases (descritas posteriormente) según sean necesarias.

20 El término Grupo se utiliza aquí para hacer referencia a un “grupo de direcciones” IP.

El término Grupo de Repetidores hace referencia a un Grupo que contiene la dirección IP de un repetidor.

25 El término Coste de los Vínculos hace referencia a un coste determinado de forma estática para transmitir datos entre dos grupos. En una realización preferida actualmente, esto es el mínimo de la suma de los costes de los vínculos a lo largo de cada ruta entre ellos. Los costes de los vínculos a los que básicamente se hace referencia aquí son los costes de los vínculos entre un Grupo y un Grupo de Repetidores.

30 El término Coste de Vínculos relativo hace referencia al coste de los vínculos relacionado con otros costes de vínculos para el mismo Grupo, que se calcula restando el coste mínimo de los vínculos de un Grupo a cualquier Grupo de Repetidores de cada uno de sus costes de vínculo a un Grupo de Repetidores.

35 El término Conjunto de Costes hace referencia a un conjunto de Grupos que son equivalentes en relación con la Selección del Mejor Repetidor. Es decir, dada la información disponible, se seleccionaría el mismo repetidor para cualquiera de ellos.

El procedimiento NetMap primero procesa archivos de entrada para crear una base de datos interna denominada Registro de Grupos. Estos archivos de entrada describen grupos, las direcciones IP dentro de los grupos, y vínculos entre los grupos, y muestra una variedad de fuentes, incluyendo bases de datos del Registro de Encaminamiento de Internet (IRR) públicamente disponibles, tablas de encaminadores BGP, y servicios de sondeo que se localizan en varios puntos alrededor de Internet y utilizan herramientas públicamente disponibles (tales como una “rutina de seguimiento”) para tomar muestras de las rutas de datos. Una vez que se haya completado este procesamiento, el Registro de Grupos contiene información esencial utilizada para el procesamiento subsiguiente, concretamente, (1) la identidad de cada grupo, (2) el conjunto de direcciones IP de un grupo determinado, (3) la presencia de vínculos entre grupos que indican rutas por las que puede desplazarse la información, y (4) el coste de enviar datos por un determinado vínculo.

50 Entonces se llevan a cabo los siguientes procesos en el archivo de Registro de Grupos.

Calcular los costes de los vínculos del Grupo de Repetidores

55 El procedimiento NetMap calcula un “coste de un vínculo” para la transmisión de datos entre cada Grupo de Repetidores y cada Grupo del Registro de Grupos. Este coste general del vínculo se define como el coste mínimo de cualquier ruta entre los dos grupos, donde el coste de una ruta es igual a la suma de los costes de los vínculos individuales de la ruta. El algoritmo del coste de vínculo presentado posteriormente es esencialmente el mismo algoritmo #562 de ACM journal Transactions on Mathematical Software: “Shortest Path From a Specific Node to All Other Nodes in a Network”, por U. Pape, ACM TOMS 6 (1980) pp. 450-455, <http://www.netlib.org/toms/562>.

60 En este procesamiento, el término Grupo de Repetidores hace referencia a un Grupo que contiene la dirección IP de un repetidor. Un grupo es adyacente a otro grupo si el Registro de Grupo indica que existe un vínculo entre los dos grupos.

65 Para cada Grupo T Repetidor de destino:

- Iniciar el coste de vínculo entre T y el mismo hacia cero.

ES 2 206 132 T5

- Iniciar el coste del vínculo entre T y cualquier otro Grupo hacia el infinito.
- Crear una lista L que contendrá Grupos que son equidistantes del Grupo T Repetidor de destino.
- 5 • Iniciar la lista L para contener el propio Grupo T Repetidor de destino.
- Mientras la lista L no está vacía:
 - 10 ○ Crear una lista L' vacía de elementos adyacentes a los miembros de la lista L.
 - Para cada Grupo G de la lista L:
 - 15 ■ Para cada Grupo N que está adyacente a G:
 - Permitir que el coste haga referencia a la suma del coste del vínculo entre T y G, y el coste de vínculo entre G y N. El coste entre T y G se determinó en el paso anterior del algoritmo; el coste del vínculo entre G y N procede del Registro de Grupo.
 - 20 • Si el coste es menor que el coste del vínculo entre T y N:
 - Ajustar el coste del vínculo entre T y N.
 - 25 ○ Añadir N a L' si aún no están sumadas.
 - Ajustar L a L'.

30 *Calcular Conjuntos de Costes*

Un Conjunto de Costes es un conjunto de Grupos que son equivalentes respecto a la Selección del Mejor Repetidor. Es decir, dada la información disponible, se seleccionaría el mismo repetidor para cualquiera de ellos.

35 El “perfil de coste” de un Grupo G se define aquí como el conjunto de costes entre G y cada Repetidor. Se dice que dos perfiles de costes son equivalentes si los valores en un perfil difieren en una cantidad constante de los valores correspondientes en el otro perfil.

40 Una vez que se conoce un Grupo de cliente, el algoritmo de Selección del Mejor Repetidor se basa en el perfil de costes para la información sobre el Grupo. Si dos perfiles son equivalentes, el algoritmo BRS seleccionaría el mismo repetidor dado cualquiera de los perfiles.

Un Conjunto de Costes es entonces un conjunto de grupos que tienen perfiles de costes equivalentes.

45 La eficacia de este método puede observarse, por ejemplo, en el caso en el que todas las rutas hacia un Repetidor desde algún Grupo A pasen a través de algún otro Grupo B. Los dos Grupos tienen perfiles de costes equivalentes (y, por tanto, están en el mismo Conjunto de Costes) puesto que cualquier Repetidor que sea el mejor para el Grupo A también va a ser el mejor para el Grupo B, sin tener en cuenta qué ruta se ha tomado entre los dos Grupos.

50 Al normalizar los perfiles de costes, los perfiles de costes equivalentes pueden hacerse idénticos. Un perfil de costes normalizado es un perfil de costes en el que el coste mínimo tiene el valor de cero. Un perfil de costes normalizado se calcula hallando el coste mínimo del perfil, y restando el valor de cada coste del perfil.

Entonces, los Conjuntos de Costes se calculan utilizando el siguiente algoritmo:

- 55 • Para cada Grupo G:
 - calcular el perfil de costes normalizado para G
 - 60 ○ buscar un Conjunto de Costes con el mismo perfil de costes normalizado.
 - Si se encuentra un conjunto de este tipo, añadir G al Conjunto de Costes existente;
 - 65 ○ de lo contrario, crear un nuevo Conjunto de Costes con el perfil de costes normalizado calculado, que sólo contiene G.

El algoritmo para hallar los Conjuntos de Costes emplea una tabla de elección arbitraria para reducir el tiempo necesario para determinar si ya existe el Conjunto de Costes deseado. La tabla de elección arbitraria emplea un valor arbitrario calculado a partir del perfil de costes de G.

Cada Conjunto de Costes se numera entonces con un único número de Índice de Envío de Costes. Los Conjuntos de Costes se utilizan entonces de una forma sencilla para generar la Tabla de Costes de Vínculos, que proporciona el coste a cada Repetidor a partir de cada Conjunto de Costes.

Como se describirá posteriormente, la Tabla de Reducción de Grupos relaciona cada dirección IP con uno de estos Conjuntos de Costes.

Construir el Mapa IP

El Mapa IP es una lista clasificada de registros que relacionan rangos de direcciones IP con claves de la Tabla de Costes de Vínculos. El formato del mapa IP es:

<dirección IP base><máxima dirección IP><clave de Tabla de Costes de Vínculos>

donde las direcciones IP se representan actualmente por números enteros de 32 bits. Estas entradas se clasifican por la dirección base descendente, y por la máxima dirección ascendente entre direcciones base iguales, y por la clave de Tabla de Costes de Vínculos ascendente entre direcciones base iguales y direcciones máximas. Obsérvese que los rangos pueden solaparse.

El procedimiento NetMap genera un mapa IP intermedio que contiene un mapa entre los rangos de las direcciones IP y los números de los Conjuntos de Costes de la siguiente manera:

- Para cada Conjunto S de costes:
 - Para cada Grupo G en S:
 - Para cada rango de dirección IP en G:
 - Añadir un triplete (dirección baja, dirección alta, número de Conjunto de Costes de S) al mapa IP.

Entonces, el archivo de mapa IP se clasifica por la dirección base descendente, y por la máxima dirección ascendente entre direcciones base iguales, y por el número ascendente del Conjunto de Costes entre direcciones base iguales y direcciones máximas. El orden de clasificación para la dirección base y la dirección máxima minimiza el tiempo para construir la Tabla de Reducción de Grupos y produce los resultados apropiados para entradas que se solapan.

Finalmente, el procedimiento NetMap crea la Tabla de Reducción de Grupos procesando el mapa IP clasificado. La Tabla de Reducción de Grupos detalla direcciones IP (especificadas por rangos) en números de Conjuntos de Costes. Es necesario el procesamiento especial del archivo de mapa IP para detectar rangos de direcciones que se solapan, y para mezclar rangos de direcciones adyacentes para minimizar el tamaño de la Tabla de Reducción de Grupos.

Se mantiene una lista ordenada de segmentos de rangos de direcciones, componiéndose cada segmento de una dirección B base y de un número N del Conjunto de Costes, clasificado por la dirección B base. (La máxima dirección de un segmento es la dirección base del siguiente segmento menos uno).

Se utiliza el siguiente algoritmo:

- Iniciar la lista con los elementos [-infinito, NO-GRUPO], [+infinito, NOGRUPO].
 - Para cada entrada en el mapa IP, en orden clasificado, que se compone de (b, m, s),
 - Insertar (b, m, s) en la lista (recuérdese que las entradas del mapa IP están en la forma (dirección baja, número del Conjunto de Costes de dirección alta de S)).
- Para cada rango (b, m) de dirección LAN reservado:
 - Insertar (b, m, LOCAL) en la lista.
- Para cada Repetidor en la dirección a:
 - Insertar (a, a, REPETIDOR) en la lista.

ES 2 206 132 T5

- Para cada segmento S de la lista ordenada:

- Mezclar S con los siguientes segmentos con el mismo Conjunto de Costes

- Crear una entrada de la Tabla de Reducción de Grupos con dirección base procedente de la dirección base de S,

- Dirección máxima = siguiente base-1 de segmento.

- Indicador de grupo = Número del Conjunto de Costes de S.

Un rango de dirección LAN reservado es un rango de dirección reservado para ser utilizado por redes LAN que no deberían aparecer como una dirección de Internet global. LOCAL es un índice de Conjunto de Costes especial diferente de todos los otros, que indica que el rango se dirige a un cliente que nunca debería reflejarse. REPETIDOR es un índice de Conjunto de Costes especial diferente de todos los otros, que indica que el rango de dirección se dirige a un repetidor. NOGRUPO es un índice de Conjunto de Costes especial diferente de todos los otros, que indica que este rango de direcciones no tiene correlación conocida.

Dados (B, M, N), insertar una entrada en la lista de direcciones ordenada de la siguiente manera:

Encontrar el último segmento (AB, AN) para el cual AB es menor o igual que B.

Si AB es menor que B, insertar un nuevo segmento (B, N) después de (AB, AN).

Hallar el último segmento (YB, YN) para el cual YB es menor o igual que M.

Reemplazar por (XB, N) cualquier segmento (XB, NOGRUPO) para el cual XB sea mayor que B y menor que YB.

Si YN es distinto de N, y también YN es distinto de NOGRUPO o YB es menor o igual que B,

Permitir que (ZB, ZN) sea el segmento que sigue a (YB, YN).

Si M+1 es menor que ZB, insertar un nuevo segmento (M+1, YN) antes de (ZB, ZN).

Reemplazar (YB, YN) por (YB, N).

Rescribir recursos HTML

Como se explicó anteriormente en relación con la figura 3 (paso B5), cuando un reflector o repetidor suministra un recurso que incluye en sí mismo localizadores de recursos (por ejemplo, un recurso HTML), se modifica (rescribe) ese recurso para reflejar previamente localizadores de recursos (localizadores URL) de recursos repetibles que aparecen en el recurso. La reescritura garantiza que cuando un navegador solicita recursos repetibles identificados por el recurso solicitado, los obtiene desde un repetidor sin volver al servidor origen, pero cuando solicita recursos no repetibles identificados por el recurso solicitado, irá directamente al servidor origen. Sin esta optimización, el navegador realizaría todas las peticiones al servidor origen (aumentando el tráfico en el servidor origen y haciendo necesarias muchas más redirecciones a partir del servidor origen), o realizaría todas las peticiones al repetidor (haciendo que el repetidor solicitara y copiara de modo redundante los recursos, que no podrían almacenarse en la memoria caché, aumentando la sobrecarga de suministrar este tipo de recursos).

La reescritura requiere que se haya seleccionado un repetidor (como se describió anteriormente en relación con el Selector del Mejor Repetidor). La reescritura emplea una denominada directiva BASE. La directiva BASE permite que el HTML identifique un servidor base diferente. (La dirección base es normalmente la dirección del recurso HTML):

La reescritura se lleva a cabo de la siguiente manera:

F1. Se añade una directiva BASE al comienzo del recurso HTML, o se modifica cuando sea necesario. Normalmente, un navegador interpreta qué localizadores URL relativos son relativos respecto a la dirección base por defecto, concretamente, el localizador URL del recurso HTML (página) en la que se encuentran. La dirección BASE añadida especifica el recurso al reflector que suministró originalmente el recurso. Esto significa que los localizadores URL relativos no procesados (tal como los generados por programas JavascriptTM) se interpretarán como relativos al reflector. Sin esta dirección BASE, los navegadores combinarían las direcciones relativas con nombres de repetidores para crear localizadores URL que no estuvieran en la forma requerida por los repetidores (como se describió anteriormente en el paso D1).

F2. El dispositivo de reescritura identifica directivas, tales como imágenes y anclas incrustadas, que contienen localizadores URL. Si el dispositivo de reescritura se está ejecutando en un reflector, debe analizar el archivo HTML para identificar estas directivas. Si se está ejecutando en un repetidor, el dispositivo de reescritura puede tener acceso a información calculada previamente que identifica la localización de cada localizador URL (colocado en el archivo HTML en el paso F4).

F3. Para cada localizador URL encontrado en el recurso que se va a describir, el dispositivo de reescritura debe determinar si puede repetirse el localizador URL (como en los pasos B1-B2). Si no puede repetirse el localizador URL, no se modifica. Por otra parte, si puede repetirse el localizador URL, se modifica para hacer referencia al repetidor seleccionado.

F4. Después de que se hayan identificado y modificado todos los localizadores URL, si se está suministrando el recurso a un repetidor, se adjunta una tabla al comienzo del recurso que identifica la localización y el contenido de cada localizador URL encontrado en el recurso. (Este paso es una optimización que elimina la necesidad de analizar los recursos HTML en el repetidor).

F5. Una vez que se hayan identificado todos los cambios, se calcula una nueva longitud para el recurso (página). La longitud se introduce en la cabecera HTTP antes de suministrar el recurso.

Actualmente se está desarrollando una extensión de HTML, conocida como XML. El proceso de describir localizadores URL será similar para XML, con algunas diferencias en el mecanismo que analiza el recurso e identifica los localizadores URL incrustados.

Procesar Protocolos No HTTP

Esta invención hace posible reflejar referencias a recursos que son suministrados por protocolos diferentes de HTTP, por ejemplo, el Protocolo de Transferencia de Archivos (FTP) y protocolos de corrientes de audio/vídeo. Sin embargo, muchos protocolos no proporcionan la capacidad de redireccionar peticiones. Sin embargo, es posible redireccionar referencias antes de que se realicen realmente las peticiones rescribiendo localizadores URL incrustados en páginas HTML. Las siguientes modificaciones a los algoritmos anteriores se utilizan para soportar esta capacidad.

En el paso F4, el dispositivo de reescritura rescribe localizadores URL para servidores si esos servidores aparecen en una tabla configurable del servidor origen que coopera o los denominados co-servidores. El operador del reflector puede definir esta tabla para incluir los servidores FTP y otros servidores. Un localizador URL rescrito que hace referencia a un recurso no-HTTP toma la forma:

`http://<repetidor>/<servidor origen>@proxy=<esquema>[:<tipo>]@/recurso`

donde <esquema> es un nombre de protocolo soportado tal como "ftp". Este formato URL es una alternativa a la forma mostrada en el paso B3.

En el paso C3, el repetidor busca un protocolo incrustado en la respuesta que llega. Si está presente un protocolo y el recurso solicitado aún no está almacenado en la memoria caché, el repetidor utiliza el protocolo seleccionado en lugar del protocolo HTTP por defecto para solicitar el recurso cuando lo está suministrando y almacenando en la memoria caché.

Configuración y Administración del sistema

Además del procesamiento anteriormente descrito, la red de repetidores requiere varios mecanismos para la configuración del sistema y para la administración de la red. Aquí se describen algunos de estos mecanismos.

Los reflectores permiten a sus operadores que sincronicen las memorias cachés de los repetidores al realizar tareas de edición. Posteriormente se describirá el proceso de mantener sincronizadas las memorias cachés de los repetidores. La edición indica que un recurso o una colección de recursos ha cambiado.

Los repetidores y reflectores participan en diversos tipos de procesamiento de registro. Los resultados de los registros recogidos en los repetidores se reúnen y fusionan con registros recogidos en los reflectores, como se describirá posteriormente.

Añadir Abonados a la Red del Repetidor

Cuando se añade un nuevo abonado a la red, se introduce información sobre el abonado en una Tabla de Abonados del repetidor maestro y se difunde a todos los repetidores de la red. Esta información incluye la Velocidad de Transferencia de Información de Agregado Asignada (Committed Aggregate Information Rate, CAIR) para los servidores que pertenecen al abonado, y una lista de los repetidores que pueden utilizarse por los servidores que pertenecen al abonado.

Añadir Reflectores a la Red de Repetidores

Cuando se añade un nuevo reflector a la red, simplemente se conecta y se anuncia él mismo a un repetidor de contacto, preferiblemente mediante un certificado encriptado de forma segura que incluye el localizador del abonado del repetidor.

El repetidor de contacto determina si la dirección de la red del reflector está permitida para este abonado. Si lo está, el repetidor de contacto acepta la conexión y actualiza el reflector con todas las tablas necesarias (mediante números de versión para determinar qué tablas están desfasadas).

El reflector procesa las peticiones durante este tiempo, pero no está “capacitado” (autorizado para reflejar peticiones) hasta que se actualicen todas sus tablas.

Mantener sincronizadas las memorias cachés de los Repetidores

Las memorias caché son coherentes, en el sentido de que cuando un reflector identifica un cambio a un recurso, éste se le notifica a todas las memorias caché de los repetidores, y acepta el cambio en una única transacción.

Sólo se transmite el localizador del recurso modificado (y no todo el recurso) a los repetidores; el localizador se utiliza para invalidar de forma eficaz el recurso correspondiente almacenado en la memoria caché del repetidor. Este proceso es mucho más eficaz que transmitir el contenido del recurso modificado a cada repetidor.

Un repetidor cargará el recurso nuevamente modificado la próxima vez que sea solicitado.

En el reflector, un cambio de recurso se identifica de forma manual por el operador, o a través de un programa de instrucciones cuando los archivos se instalan en el servidor, o automáticamente a través de un mecanismo de detección de cambios (por ejemplo, un proceso independiente que comprueba regularmente si se producen cambios).

Un cambio de recurso hace que el reflector envíe un mensaje “invalidar” a su repetidor de contacto, que envía el mensaje al repetidor maestro. El mensaje de invalidar contiene una lista de localizadores de recursos (o expresiones regulares que identifican modelos de localizadores de recursos) que han cambiado. (Las expresiones regulares se utilizan para invalidar un directorio o un servidor completo). La red de repetidores utiliza un proceso de asignación en dos fases para garantizar que todos los repetidores invalidan correctamente un recurso determinado.

El proceso de invalidación funciona de la siguiente manera:

El maestro difunde una petición de invalidación de “fase 1” a todos los repetidores que indica los recursos y expresiones regulares que describen conjuntos de recursos que se van a invalidar.

Cuando cada uno de los repetidores recibe el mensaje de fase 1, primero coloca los localizadores de recursos o expresiones regulares en una lista de localizadores de recursos en trámites de invalidación.

Ninguno de los recursos solicitados (en el paso C3) que esté en la lista de invalidación pendiente podrá suministrarse desde la memoria caché. Esto impide que la memoria caché solicite el recurso desde una memoria caché del mismo nivel, que puede no haber recibido una notificación de invalidación. Si fuera a solicitar un recurso de esta manera, tendría que reemplazar el recurso nuevamente invalidado por los mismos datos, ahora desfasados.

Entonces, el repetidor compara el localizador de recursos de cada recurso en su memoria caché frente a los localizadores de recursos y las expresiones regulares de la lista.

Cada coincidencia se invalida marcándola como anticuada y eliminándola opcionalmente de la memoria caché. Esto significa que una futura petición del recurso hará que recupere una nueva copia del recurso desde el reflector.

Cuando el repetidor ha completado la invalidación, muestra un acuse de recibo al maestro. El maestro espera hasta que todos los repetidores hayan acusado recibo de la petición de invalidación.

Si un repetidor no emite un acuse de recibo dentro de un determinado periodo de tiempo, se desconecta del repetidor maestro. Cuando se vuelve a conectar, se le solicitará que purgue toda su memoria caché, lo que eliminará cualquier problema de consistencia. (Para evitar purgar toda la memoria caché, el maestro podría mantener un registro de todas las invalidaciones llevadas a cabo, clasificadas por fecha, y purgar sólo los archivos invalidados desde la última vez que el repetidor que se vuelve a conectar completó con éxito una invalidación. En las realizaciones preferidas actualmente, esto no se hace puesto que se considera que los repetidores se desconectarán en raras ocasiones).

Cuando todos los repetidores hayan acusado recibo de la invalidación (o estén en tiempo de espera), el repetidor transmite una petición de invalidación de “fase 2” a todos los repetidores. Esto hace que los repetidores eliminen los localizadores de recursos correspondientes y las expresiones regulares de la lista de localizadores de recursos pendientes de invalidación.

En otra realización, la petición de invalidación se extenderá para permitir un “empuje del servidor”. En este tipo de peticiones, después de que se haya completado la fase 2 del proceso de invalidación, el repetidor que recibe la petición de invalidación solicitará inmediatamente una nueva copia del recurso invalidado para almacenarla en su memoria caché.

Registros y Procesamiento de registros

Los registros de la actividad del servidor Web son fundamentalmente para controlar la actividad en un sitio Web. Esta invención crea “registros fusionados” que combinan la actividad en los reflectores con la actividad en los repetidores, de manera que sólo aparece un registro de actividad en el servidor origen que muestra todas las peticiones de recursos Web realizadas para ese sitio en cualquier repetidor.

Este registro fusionado puede procesarse mediante herramientas de procesamiento estándar, como si se hubiera generado de forma local.

Sobre una base periódica, el repetidor maestro (o su delegado) recoge los registros de cada repetidor. Los registros recogidos se fusionan, se clasifican por el localizador del reflector y el sello temporal, y se almacenan en un archivo fechado basándose en el reflector. El registro fusionado para un determinado reflector representa la actividad de todos los repetidores por parte de ese reflector. Sobre una base periódica, según se configura por el operador del reflector, un reflector contacta con el repetidor maestro para solicitar sus registros fusionados. Los descarga y los fusiona con sus registros mantenidos de forma local, clasificándolos según su sello temporal. El resultado es un registro fusionado que representa toda actividad por parte de los repetidores y del reflector determinado.

Los registros de actividad se extienden opcionalmente con información importante para la red de repetidores, si el reflector está configurado para hacerlo así por el operador del reflector. En particular, un “código de estado extendido” indica información sobre cada petición, tal como:

1. la petición se suministró localmente por un reflector;
2. la petición se reflejó a un repetidor; *
3. la petición se suministró a un repetidor mediante un reflector;*
4. la petición del recurso no repetible se suministró por medio del repetidor;*
5. la petición se suministró desde la memoria caché por medio de un repetidor;
6. la petición se suministró por medio de un repetidor tras completar la memoria caché;
7. la petición pendiente de invalidación se suministró por medio de un repetidor.

(Las actividades marcadas con “*” representan estados intermedios de una petición y normalmente no aparecen en un registro final de actividad).

Además, los registros de actividad contienen una duración, y sellos temporales de precisión ampliados. La duración hace posible analizar el tiempo necesario para suministrar un recurso, el ancho de banda empleado, el número de peticiones procesadas en paralelo a un tiempo determinado, y otra información bastante útil. El sello temporal de precisión extendido hace posible fusionar de forma precisa los registros de actividad.

Los repetidores utilizan el Protocolo de Temporización de Red (NTP) para mantener los relojes sincronizados. Los reflectores pueden utilizar el protocolo NTP o calcular un sesgo temporal para proporcionar sellos temporales aproximadamente precisos respecto a su repetidor de contacto.

Imponer la Velocidad de Transferencia de Información de Agregado Asignada

La red de repetidores controla y limita la velocidad de transferencia del agregado a la cual los datos para un determinado abonado son suministrados por todos los repetidores. Este mecanismo proporciona los siguientes beneficios:

1. proporciona un medio de fijar los precios de un servicio de repetidor;
2. proporciona un medio para calcular y reservar la capacidad, en los repetidores;
3. proporciona un medio para impedir que los clientes de un sitio ocupado limiten el acceso a otros sitios.

Para cada abonado, en el repetidor maestro se configura y mantiene una “Velocidad de Transferencia de Información Umbral de Agregado” (TAIR). Este umbral no es necesariamente la velocidad de transferencia asignada, puede ser un múltiplo de la velocidad de transferencia asignada, basándose en una política de fijación de precios.

5 Cada repetidor mide el componente de la velocidad de transferencia de información de cada reflector para el cual suministra recursos, periódicamente (normalmente, una vez cada minuto), grabando el número de bytes transmitidos dirigidos a ese reflector cada vez que se suministra una petición. La tabla creada de esta manera se envía al repetidor maestro una vez por periodo. El repetidor maestro combina las tablas de cada repetidor, sumando la información medida de cada reflector de todos los repetidores que suministran recursos para ese reflector, para determinar la “Velocidad de Transferencia de Información de Agregado Medida” (MAIR) para cada reflector.

Si la velocidad MAIR para un determinado reflector es mayor que la velocidad TAIR para ese reflector, la velocidad MAIR se transmite por medio del maestro a todos los repetidores y al reflector correspondiente.

15 Cuando un reflector recibe una petición, éste determina si su velocidad MAIR calculada más recientemente es mayor que su velocidad TAIR. Si es éste el caso, el reflector decide, según las probabilidades, si suprimir la reflexión, suministrando la petición de modo local (en el paso B2). La probabilidad de suprimir la reflexión aumenta como una función exponencial de la diferencia entre la velocidad MAIR y CAIR.

20 Suministrar una petición de forma local durante un periodo cumbre puede ejercer presión sobre el servidor origen local; pero impide que este abonado tome más del ancho de banda asignado de la red de repetidores compartida.

25 Cuando un repetidor recibe una petición para un determinado abonado (en el paso C2), éste determina si el abonado se está ejecutando próximo a su velocidad de transferencia de información umbral de agregado. Si es este el caso, decide según la probabilidad si reduce su carga volviendo a dirigir la petición nuevamente al reflector. La probabilidad aumenta de forma exponencial mientras la velocidad de transferencia de información de agregado del reflector se aproxima a su límite.

30 Si una petición se refleja nuevamente hacia un reflector, se adjunta una cadena especial de caracteres al localizador de recursos de manera que el reflector que la recibe no intentará reflejarla nuevamente. En el sistema actual esta cadena tiene la siguiente forma:

“src=sobrecarga”.

35 En el paso B2, el reflector comprueba esta cadena.

40 El mecanismo para limitar la Velocidad de Transferencia de Información de Agregado descrito anteriormente es bastante tosco. Éste limita en el nivel de las sesiones con clientes (puesto que una vez que un cliente ha sido reflejado a un determinado repetidor, el proceso de reescritura tiende a mantener que el cliente regrese al repetidor) y, en el mejor de los casos, peticiones individuales para recursos. Un mecanismo más pulido para imponer los límites TAIR dentro de los repetidores funciona reduciendo el consumo de ancho de banda de un abonado ocupado cuando otros abonados están compitiendo por el ancho de banda.

45 El mecanismo más finamente pulido es una forma de “rate shaping” (adaptación de la velocidad de transferencia) de los datos. Éste amplía el mecanismo que copia los datos de los recursos a una conexión cuando se ha enviado una respuesta a un cliente. Cuando se establece un canal de salida en el momento en que se recibe una petición, el repetidor identifica para qué abonado está operando el canal, en el paso C2, y registra el abonado en un campo de datos asociado con el canal. Cada vez que está próxima a realizarse una operación de “escritura” en el canal, la Corriente de Salida Medida examina los valores actuales de las velocidades MAIR y TAIR, calculadas anteriormente, para el abonado determinado. Si la velocidad MAIR es mayor que la velocidad TAIR, entonces el mecanismo se detiene brevemente antes de realizar la operación de escritura. La longitud de la pausa es proporcional a la cantidad en que la velocidad MAIR sobrepasa a la velocidad TAIR. La pausa garantiza que las tareas que envían otros recursos a otros clientes, quizás dirigidos a otros abonados, tendrán una oportunidad de enviar sus datos.

55 *Capacidad de Recuperación de la Red de Repetidores*

La red de repetidores es capaz de recuperarse cuando un repetidor o conexión de red falla.

60 Un repetidor no puede operar a menos que esté conectado con el repetidor maestro. El repetidor maestro intercambia información crítica con otros repetidores, incluyendo información sobre la carga del repetidor, la velocidad de transferencia de información de agregado, abonados y coste de los vínculos.

65 Si un maestro falla, un proceso de “sucesión” garantiza que otro repetidor asumirá el papel del maestro, y la red, como un todo, permanecerá operativa. Si un maestro falla, o falla una conexión a un maestro debido a un problema de red, cualquier repetidor que intente comunicar con el maestro detectará el fallo, bien a través de una indicación del protocolo TCP/IP, o por el tiempo de espera a partir de un mensaje de “latido” regular que envía al maestro.

Cuando cualquier repetidor está desconectado de su maestro, éste trata de volverse a conectar inmediatamente a una serie de maestros potenciales basándose en un archivo configurable denominado su “lista de sucesión”.

5 El repetidor prueba cada sistema de la lista de forma sucesiva hasta que se conecta con un maestro de forma satisfactoria. Si en este proceso llega a su propio nombre, asume el papel del maestro, y acepta las conexiones de otros repetidores. Si un repetidor que no se encuentra en la parte superior de la lista se convierte en el maestro, se le denomina “maestro temporal”.

10 Una partición de red puede hacer que cada uno de dos grupos de repetidores elija a un maestro. Al corregir la partición, es necesario que el maestro más antiguo tome la red. Por tanto, cuando un repetidor es un maestro temporal, éste trata regularmente de volverse a conectar a cualquier maestro por encima suyo en la lista de sucesión. Si esto sucede, éste se desconecta inmediatamente de todos los demás repetidores conectados a él. Cuando ellos vuelven a probar sus listas de sucesión, conectarán con el repetidor maestro más antiguo.

15 Para evitar las pérdidas de datos, un maestro temporal no acepta los cambios de configuración y no procesa archivos de registro. Para llevar a cabo estas tareas, debe estar informado de que es maestro primario mediante modificación manual de su lista de sucesores. Cada repetidor recarga regularmente su lista de sucesores para determinar si debería cambiar su idea sobre quién es el maestro.

20 Si un repetidor está desconectado del maestro, debe volver a sincronizar su memoria caché cuando se vuelve a conectar al maestro. El maestro puede mantener una lista de invalidaciones recientes de la memoria caché y enviarle al repetidor cualquier invalidación que no haya sido capaz de procesar mientras estuvo desconectado. Si por alguna razón esta lista no está disponible (por ejemplo, debido a que el reflector ha sido desconectado demasiado tiempo), el reflector debe invalidar toda su memoria caché.

25 No se le permite a un reflector que refleje peticiones a menos que esté conectado a un repetidor. El reflector se basa en su repetidor de contacto para la información crítica, tal como las Tablas de Carga y de Costes de Vínculos, y la velocidad actual de transferencia de información de agregado. Un reflector que no está conectado a un repetidor puede continuar recibiendo peticiones y procesándolas de forma local.

30 Si un reflector pierde su conexión con un repetidor, debido a un fallo del repetidor o corte en la red, continua funcionando mientras trata de conectarse a un repetidor.

35 Cada vez que un reflector intenta conectarse a un repetidor, utiliza el servidor DNS para identificar a un conjunto de repetidores candidatos dado un nombre de dominio que representa a la red de repetidores. El reflector lo intenta con cada repetidor de este conjunto hasta que establece un contacto satisfactorio. Hasta que se establece un contacto satisfactorio, el reflector suministra todas las peticiones de forma local. Cuando un reflector se conecta con un repetidor, el repetidor puede solicitarle que intente contactar con un repetidor diferente; esto le permite a la red de repetidores garantizar que ningún repetidor individual cuenta con demasiados contactos.

40 Cuando se establece el contacto, el reflector le proporciona el número de versión de cada una de sus tablas a su repetidor de contacto. Entonces, el repetidor decide qué tablas deberían actualizarse y le envía al reflector las actualizaciones adecuadas. Una vez que se hayan actualizado todas las tablas, el repetidor le notifica al reflector que ahora puede empezar a reflejar las peticiones.

45 *Utilizar un proxy-caché dentro de un repetidor*

50 Los repetidores están diseñados intencionadamente de manera que cualquier proxy-caché pueda utilizarse como un componente dentro de ellos. Esto es posible porque el repetidor recibe peticiones HTTP y las convierte a una forma reconocida por el proxy-caché.

55 Por otra parte, se han realizado diversas modificaciones a un proxy-caché estándar o pueden realizarse como optimizaciones. Esto incluye, en particular, la capacidad de invalidar convenientemente un recurso, la capacidad de soportar las cuotas de la memoria caché y la capacidad de evitar hacer una copia extra de cada recurso según pasa del proxy-caché, a través del repetidor, al solicitante.

60 En una realización preferida, se utiliza un proxy-caché para llevar a cabo el paso C3. El proxy-caché está dedicado para ser utilizado únicamente por uno o varios repetidores. Cada repetidor que solicita un recurso del proxy-caché construye una petición al proxy a partir de la petición de recurso que llega. Una petición HTTP GET normal a un servidor contiene sólo la parte del nombre de ruta del localizador URL, el esquema y el nombre del servidor están implícitos. (En una petición HTTP GET a un repetidor, la parte del nombre de ruta del localizador URL incluye el nombre del servidor origen para el cual se está realizando la petición, como se describió anteriormente). Sin embargo, una petición GET de agente proxy toma un localizador URL completo. Por tanto, el repetidor debe construir una petición al proxy que contiene todo el localizador URL a partir la parte de la ruta del localizador URL que recibe. De forma específica, si la petición entrante adquiere la siguiente forma:

GET/<servidor origen>/<ruta>

entonces, el repetidor construye una petición al proxy de la siguiente forma:

```
GET http://<servidor origen>/<ruta>
```

5 y si la petición entrante adquiere la siguiente forma:

```
GET <servidor origen>@proxy=<esquema>:<tipo>@/<ruta>
```

entonces, el repetidor construye una petición al proxy de la siguiente forma:

```
10 GET <esquema>://<servidor origen>/<ruta>
```

15 *Control de la Memoria Caché*

Las respuestas HTTP contienen directivas denominadas directivas de control de la memoria caché, que se utilizan para indicarle a una memoria caché si el recurso adjunto puede almacenarse en la memoria caché y, si es así, cuando debería caducar. Un administrador de sitio Web configura el sitio Web para adjuntar directivas adecuadas. Con frecuencia, el administrador no sabrá durante cuánto tiempo permanecerá actual una página, y debe definir un tiempo de caducidad corto para intentar impedir que las memorias caché suministren datos anticuados. En muchos casos, un operador de sitio Web indicará un corto tiempo de expiración sólo para recibir las peticiones (o respuestas pertinentes) que, de lo contrario, estarían ocultas por la presencia de una memoria caché. Esto se conoce en la industria como “cache-busting”. Aunque algunos operadores de memoria caché pueden considerar el proceso de “cache-busting” como algo poco correcto, los anunciantes que se basan en esta información pueden considerarla imperativa.

Cuando un recurso se almacena en un repetidor, sus directivas de memoria caché pueden ser ignoradas por el repetidor, puesto que el repetidor recibe eventos de invalidación explícitos para determinar cuando está anticuado un recurso. Si se utiliza un proxy-caché como la memoria caché del repetidor, las directivas asociadas de la memoria caché pueden deshabilitarse temporalmente. Sin embargo, deben rehabilitarse nuevamente cuando se suministre el recurso desde la memoria caché a un cliente, para permitir que surta efecto la política de control de la memoria caché (incluyendo cualquier proceso caché-busting).

La presente invención contiene mecanismos para impedir que el proxy-caché dentro de un repetidor cumpla con las directivas de control de la memoria caché, mientras que permiten que las directivas se suministren desde el repetidor.

Cuando un reflector suministra un recurso a un repetidor en el paso B4, éste reemplaza todas las directivas de la memoria caché por directivas modificadas que son ignoradas por el proxy-caché del repetidor. Esto lo lleva a cabo anteponiendo como prefijo una cadena distintiva, tal como “wr-”, al comienzo de la etiqueta HTTP. De esta manera, “caduca” se convierte en “wr-caduca”, y “control-memoria caché” se convierte en “wr-control-memoria caché”. Esto impide que el propio proxy-caché cumpla con las directivas. Cuando un repetidor suministra un recurso en el paso C4, y el cliente solicitante no es otro repetidor, éste busca etiquetas HTTP que comiencen con “wr-” y elimina el prefijo “wr-”. Esto permite a los clientes recuperar el recurso para cumplir con las directivas.

45 *Revalidación de Recursos*

Existen diversos casos en los que puede almacenarse un recurso en la memoria caché mientras se consulta el servidor origen cada vez que es suministrado. En un caso, la petición del recurso se adjunta a un denominado “cookie”. El servidor origen debe presentarse con el cookie para registrar la petición y determinar si debe suministrarse o no el recurso almacenado en la memoria caché. En otro caso, la petición del recurso se adjunta a una cabecera de autenticación (que identifica al solicitante con un identificador de usuario y una contraseña). Cada nueva petición de recurso debe comprobarse en el servidor origen para garantizar que el solicitante está autorizado a acceder al recurso.

La especificación HTTP 1.1 define una cabecera de respuesta denominada “Debe-Revalidar”, que le permite a un servidor origen ordenarle a un proxy caché que “revalide” un recurso cada vez que se recibe una petición. Normalmente, este mecanismo se utiliza para determinar si un recurso aún sigue siendo actual. En la presente invención, Debe-Revalidar hace posible solicitarle a un servidor origen que valide una petición que, de lo contrario, se suministra a partir de un repetidor.

La base de reglas del reflector contiene información que determina qué recursos pueden repetirse, pero debe volverse a validar cada vez que éstos se suministran. Para cada uno de estos recursos, en el paso B4, el reflector adjunta una cabecera Debe-Revalidar. Cada vez que llega una petición a un repetidor de un recurso almacenado en la memoria caché marcado con una cabecera Debe-Revalidar, la petición se envía al reflector para que sea validada antes de suministrar el recurso solicitado.

Cuotas de la Memoria Caché

El componente de la memoria caché de un repetidor está compartido entre aquellos abonados que reflejan clientes a ese repetidor. Para permitirles a los abonados acceso limpio a las prestaciones de almacenamiento, la memoria caché puede ampliarse para soportar cuotas.

Normalmente, un proxy-caché puede estar configurado con un umbral T de espacio de disco. Si se almacenan más de T bytes en la memoria caché, la memoria caché intenta encontrar recursos que eliminar.

Normalmente, una memoria caché utiliza el algoritmo LRU (Utilizado Menos Recientemente) para determinar qué recursos eliminar; las memorias caché más sofisticadas utilizan otros algoritmos. Una memoria caché también puede soportar varios valores umbral, por ejemplo, un umbral más bajo que, al alcanzarlo, hace que un proceso secundario de prioridad baja elimine elementos de la memoria caché, y un umbral más alto que, al ser alcanzado, impide que los recursos se almacenen en la memoria caché hasta que se recupera suficiente espacio libre en disco.

Si dos abonados A y B comparten una memoria caché, y durante un periodo de tiempo se accede a más recursos del abonado A que a recursos del abonado B, entonces habrá menos recursos del abonado B en la memoria caché cuando lleguen nuevas peticiones. Es posible que, debido al comportamiento del abonado A, los recursos del abonado B nunca lleguen a almacenarse en la memoria caché cuando son solicitados. En la presente invención, este comportamiento es indeseable. Para tratar este tema, la invención amplía la memoria caché en un repetidor para que soporte cuotas de memoria caché.

La memoria caché registra la cantidad de espacio utilizado por cada abonado en D_s , y soporta un umbral T_s configurable para cada abonado.

Cada vez que se añade un recurso a la memoria caché (en el paso C3), el valor D_s se actualiza para el abonado que proporciona el recurso. Si D_s es mayor que T_s , la memoria caché intenta encontrar recursos que eliminar, de entre los recursos asociados con el abonado S. La memoria caché se divide de forma eficaz en áreas separadas para cada abonado.

El umbral T original aún se soporta. Si la suma de los segmentos reservados para cada abonado es menor que el espacio total reservado en la memoria caché, el área restante es "común" y está sujeta a competición entre los abonados.

Obsérvese que este mecanismo puede implementarse modificando el proxy-caché existente, explicado anteriormente, o también podría implementarse sin modificar el proxy-caché, si el proxy-caché posibilita al menos que un programa externo obtenga una lista de los recursos que hay en la memoria caché, y el eliminar un recurso determinado de la memoria caché.

Reescritura a partir de los Repetidores

Cuando un repetidor recibe una petición para un recurso, su proxy-caché puede estar configurada para determinar si una memoria caché de igual nivel contiene el recurso solicitado. Si es así, el proxy-caché obtiene el recurso de la memoria caché de igual nivel, lo que puede resultar más rápido que obtenerlo del servidor origen (el reflector). Sin embargo, una consecuencia de esto es que los recursos HTML rescritos recuperados de la memoria caché de igual nivel identificarían al repetidor erróneo. De esta manera, para que permitir los proxy-cachés cooperen, los recursos se rescriben preferiblemente en el repetidor.

Cuando un recurso se rescribe para un repetidor, se coloca una etiqueta especial al comienzo del recurso. Al construir una respuesta, el repetidor examina la etiqueta para determinar si el recurso indica que es necesaria una reescritura adicional. Si es así, el repetidor modifica el recurso reemplazando las referencias al antiguo repetidor con referencias al nuevo repetidor.

Sólo es necesario realizar esta reescritura cuando se suministra un recurso al proxy-caché de otro repetidor.

Include del Lado del Repetidor

Algunas veces, un servidor origen construye un recurso personalizado para cada petición (por ejemplo, al introducir un anuncio basándose en el historial del cliente solicitante). En un caso de este tipo, ese recurso debe suministrarse localmente, en lugar de repetirse. Generalmente, un recurso personalizado contiene, junto con la información personalizada, texto y referencias a otros recursos repetibles.

El proceso que ensambla una "página" a partir de un recurso de texto y, posiblemente, uno o varios recursos de imágenes, es llevado a cabo por el navegador Web, dirigido por HTML. Sin embargo, no es posible utilizar HTML para hacer que un navegador ensamble una página utilizando texto o directivas desde un recurso independiente. Por tanto, los recursos personalizados a menudo contienen necesariamente grandes cantidades de texto estático que, de otra forma, sería repetible.

Para solucionar esta ineficacia potencial, los repetidores reconocen una directiva especial denominada “repeater side include” (incluye del lado del repetidor). Esta directiva hace posible que el repetidor ensamble un recurso personalizado, utilizando una combinación de recursos repetibles y locales. De esta manera, el texto estático puede hacerse repetible, y sólo es necesario que el reflector suministre localmente la directiva especial.

Por ejemplo, un recurso X puede componerse de directivas personalizadas que seleccionan un letrero publicitario seguido de un artículo de texto de gran tamaño. Para hacer que pueda repetirse este recurso, el administrador del sitio Web debe hacer salir un segundo recurso, Y, para seleccionar el letrero. El recurso X se modifica para contener una directiva include del lado del repetidor que identifica el recurso Y, junto con el artículo. El recurso Y se crea y sólo contiene las directivas personalizadas que seleccionan un letrero publicitario. Ahora, el recurso X puede repetirse, y sólo el recurso Y, que es relativamente pequeño, no puede repetirse.

Cuando un repetidor construye una respuesta, determina si el recurso que se está suministrando es un recurso HTML y, si es así, lo examina en busca de directivas include del lado del repetidor. Cada una de estas directivas incluye un localizador URL, que el repetidor define y sustituye en lugar de la directiva. Todo el recurso debe ensamblarse antes de ser suministrado, para determinar su tamaño final, puesto que el tamaño se incluye en una cabecera de respuesta en la parte delantera del recurso.

De esta manera, se proporciona un método y un aparato para duplicar de forma dinámica recursos seleccionados en redes informáticas. Un experto en la técnica percibirá que la presente invención puede ponerse en práctica en una realización diferente de las descritas, las cuales se presentan con motivos ilustrativos y no restrictivos, y la presente invención sólo está limitada por las reivindicaciones que siguen.

Referencias citadas en la descripción

La lista de referencias citadas por el solicitante es solo para comodidad del lector. No forma parte del documento de Patente Europea. Aunque se ha tomado especial cuidado en recopilar las referencias, no puede descartarse errores u omisiones y la EPO rechaza toda responsabilidad a este respecto.

Documentos de patentes citados en la descripción

- WO 9 729 423 A [0016].

Bibliografía no de patentes citada en la descripción

- U. Pape. “Shortest Path From a Specific Node to All Other Nodes in a Network”. *ACM TOMS*, 1980, vol. 6, 450-455 [0093].

REIVINDICACIONES

5 1. Sistema que opera en una red (100) informática en la que los clientes (106) se conectan a un servidor (102) origen, comprendiendo el sistema:

un conjunto de servidores (104a...) repetidores, distintos del servidor (102) origen, para alojar al menos algunos de los objetos incrustados de las páginas Web que normalmente están alojados por el servidor (102) origen;

10 y un reflector (108) que comprende:

una rutina para modificar al menos un objeto URL incrustado de una página Web para indicárselo a un servidor del conjunto de servidores repetidores, en lugar de al servidor origen;

15 y el reflector comprende además

un mecanismo de selección de servidores construido y adaptado para identificar, para una determinada petición de cliente, un servidor repetidor adecuado a partir del conjunto de servidores repetidores,

20 en el que como respuesta a las peticiones de página Web generadas por los clientes, desde el reflector se suministra la página Web que incluye el localizador URL modificado del objeto incrustado, siendo el reflector parte de servidor origen, y el objeto incrustado identificado por la URL modificada incrustada se suministra desde un servidor determinado de los servidores repetidores según es identificado por el mecanismo selector de servidores;

25 **caracterizado** porque

el mecanismo selector del servidor identifica un servidor repetidor en función de la dirección de red del cliente solicitante y de datos relacionados con los costes de la conexión de red, los mencionados costes de conexión incluyendo datos que se han determinado utilizando servicios de prueba que muestrean rutas de datos en la red, con el objeto de obtener medidas que reflejen un coste relativo de transmitir datos desde servidores repetidores hasta otras localizaciones en la red.

35 2. Sistema según la reivindicación 1, en el que, si falla un determinado servidor de los servidores (104a...) repetidores, otro servidor repetidor asumirá el papel de un servidor fallido.

3. Sistema según la reivindicación 1, en el que el servidor que selecciona el mecanismo también identifica un servidor repetidor basándose en la carga de los servidores repetidores.

40 4. Sistema según la reivindicación 1, en el que el servidor que selecciona el mecanismo incluye una mapa de red para utilizarlo al dirigir una petición de objeto incrustado generada por un cliente.

5. Sistema según la reivindicación 1, que comprende además:

45 al menos un servidor DNS para proporcionar resolución de nombre a dirección, de nombres de servidor en objetos URL incrustados.

6. Método para suministrar una página soportada en un servidor origen, en el que dicho método hace funcionar un sistema operativo en una red informática, comprendiendo el sistema:

50 un conjunto de servidores (104a,...) repetidores, distintos de un servidor (102) origen, para alojar al menos algunos de los objetos incrustados de las páginas Web que normalmente son alojados por el servidor origen; y

un reflector (108) que comprende:

55 una rutina para hacer que al menos un localizador URL del objeto incrustado de una página Web se le indique a un servidor del conjunto de servidores repetidores, en lugar de al servidor origen; y comprendiendo además el reflector:

60 un mecanismo de selección de servidores construido y adaptado para identificar, para una determinada petición de cliente, un servidor repetidor adecuado del conjunto de servidores repetidores,

65 en el que el mecanismo de selección de servidores identifica un servidor repetidor en función de la dirección de red del cliente solicitante y de datos relativos a los costes de la conexión de red, incluyendo los mencionados costes de la conexión de red datos que se han determinado utilizando servicios de prueba que muestrean rutas de datos en la red, para obtener medidas que reflejen un coste relativo de transmitir datos desde servidores repetidores hasta otras localizaciones en la red,

comprendiendo la página un documento base que tiene objetos incrustados asociados, estando identificado cada objeto incrustado por un localizador de recursos uniformes (URL), comprendiendo el método:

modificar el localizador URL de un objeto incrustado para generar un localizador URL modificado, incluyendo el localizador URL modificado un nuevo nombre de anfitrión que hace referencia a un servidor repetidor del conjunto de servidores repetidores, según es identificado por el mecanismo selector de servidores;

como respuesta a una petición para suministrar una página, suministrar la página con el localizador URL modificado desde el reflector;

intentar suministrar el objeto incrustado desde un servidor repetidor según es identificado por el nuevo nombre de anfitrión;

si no está disponible una copia del objeto incrustado en el servidor repetidor, obtener por parte del servidor repetidor el objeto incrustado desde el servidor origen.

7. Método según la reivindicación 6 en el que el modificar el localizador URL comprende:

añadir un nuevo nombre de anfitrión al localizador URL original.

8. Método según la reivindicación 7, en el que el nombre de anfitrión original se mantiene como parte del localizador URL modificado.

9. Método según la reivindicación 6, en el que el localizador URL original incluye un nombre de servidor origen y una ruta, y en el que el localizador URL modificado incluye un nombre de servidor repetidor.

10. Método según la reivindicación 9, en el que el localizador URL original tiene la forma:

“http://<servidor>/<ruta>”

donde “<servidor>” es el nombre del servidor origen y donde “<ruta>” es una ruta al recurso designado por el localizador URL, y en el que el localizador URL modificado tiene la forma:

“http://<repetidor>/<servidor>/<ruta>”

donde “<repetidor>” es el nombre de anfitrión del servidor repetidor.

11. Método según la reivindicación 9, en el que el localizador URL original tiene la forma:

“http://<servidor>/<ruta>”

donde “<servidor>” es el nombre del servidor origen y donde “<ruta>” es una ruta al recurso designado por el localizador URL, y en el que el localizador URL modificado tiene la forma:

“http://<repetidor>/<servidor>/<ruta>”

donde “<repetidor>” es un nombre de anfitrión de un conjunto de servidores repetidores.

12. Método de suministrar una página según la reivindicación 7, en el que

modificar el localizador URL comprende: modificar el localizador URL para el objeto incrustado de la página para que incluya un nombre de anfitrión adjuntado a una ruta y nombre de dominio suministrados por un servidor origen;

suministrar la página comprende: suministrar la página con el localizador URL modificado desde el servidor origen a un navegador de cliente;

comprendiendo el método además:

enviar de nuevo al navegador del cliente una dirección IP del servidor repetidor identificado para permitirle al navegador intentar recuperar el objeto desde ese servidor de repetidor.

13. Método en la reivindicación 12, en el que las copias del objeto incrustado de la página se almacenan en un subconjunto del conjunto de servidores repetidores.

14. Método según la reivindicación 6, en el que la modificación del localizador URL del objeto incrustado hace que la dirección del objeto incrustado en la página indique una dirección diferente de una dirección del servidor origen al añadir determinados datos a una dirección suministrada por el servidor origen para generar una dirección alternativa.

15. Método según la reivindicación 14, en el que la red informática es Internet y la dirección del objeto es un Localizador de Recursos Uniformes (URL).

16. Método según la reivindicación 15, en el que la dirección alternativa está formada adjuntando los datos dados a una parte de la dirección suministrada por el servidor origen.

17. Método según una cualquiera de las reivindicaciones 14-16, en el que indicar la dirección alternativa comprende:

utilizar una dirección de red del cliente solicitante y datos relacionados con los costes de la conexión de Internet, para identificar el servidor repetidor.

18. Método según la reivindicación 6, comprendiendo el método adicionalmente:

duplicar un conjunto de objetos de página a través de una red de área amplia de servidores repetidores.

19. Método según se describe en la reivindicación 6, que comprende además:

para cada localizador URL modificado del objeto incrustado, identificar al menos un servidor repetidor desde el que pueda recuperarse el objeto incrustado.

20. Método según la reivindicación 19, en el que la identificación comprende:

indicar una petición para un objeto incrustado como una función de una dirección de red del cliente solicitante.

21. Método según la reivindicación 19, en el que la identificación comprende:

decidir una petición para un objeto incrustado, como una función de una dirección de red del cliente solicitante y de los costes actuales de la conexión a Internet.

22. Método según la reivindicación 6 que comprende además:

como respuesta a una petición para un objeto incrustado, indicar el nombre del anfitrión para una dirección IP de un determinado servidor repetidor.

23. Método según la reivindicación 22, en el que indicar el nombre del anfitrión incluye:

identificar un subconjunto de servidores (104a,...) repetidores que puede ser capaz de suministrar el objeto incrustado; e identificar el servidor repetidor particular desde el subconjunto identificado de servidores repetidores.

24. Método según la reivindicación 23, en el que la identificación del servidor repetidor concreto está basada en una dirección de red de la máquina cliente y en los costes de la conexión a Internet.

25. Método según la reivindicación 6, que comprende además:

distribuir un conjunto de objetos de páginas a través de una red de servidores (104a, ...) repetidores, en el que la red de servidores repetidores está organizada en un conjunto de grupos de repetidores; como respuesta a una petición de un cliente de un objeto incrustado de una página:

decidir la petición del cliente como una función de una dirección de red del cliente y de los costes de la conexión a Internet, para identificar un grupo de repetidores determinado;

reenviar al cliente una dirección IP de un servidor particular de los servidores repetidores dentro del grupo de repetidores determinado.

26. Sistema según la reivindicación 1, en el que la rutina para la modificación modifica un localizador URL para incluir un nombre de anfitrión asignado a un nombre de dominio y a una ruta.

27. Sistema según la reivindicación 1, en el que la rutina para la modificación añade un nuevo nombre de anfitrión al localizador URL original.

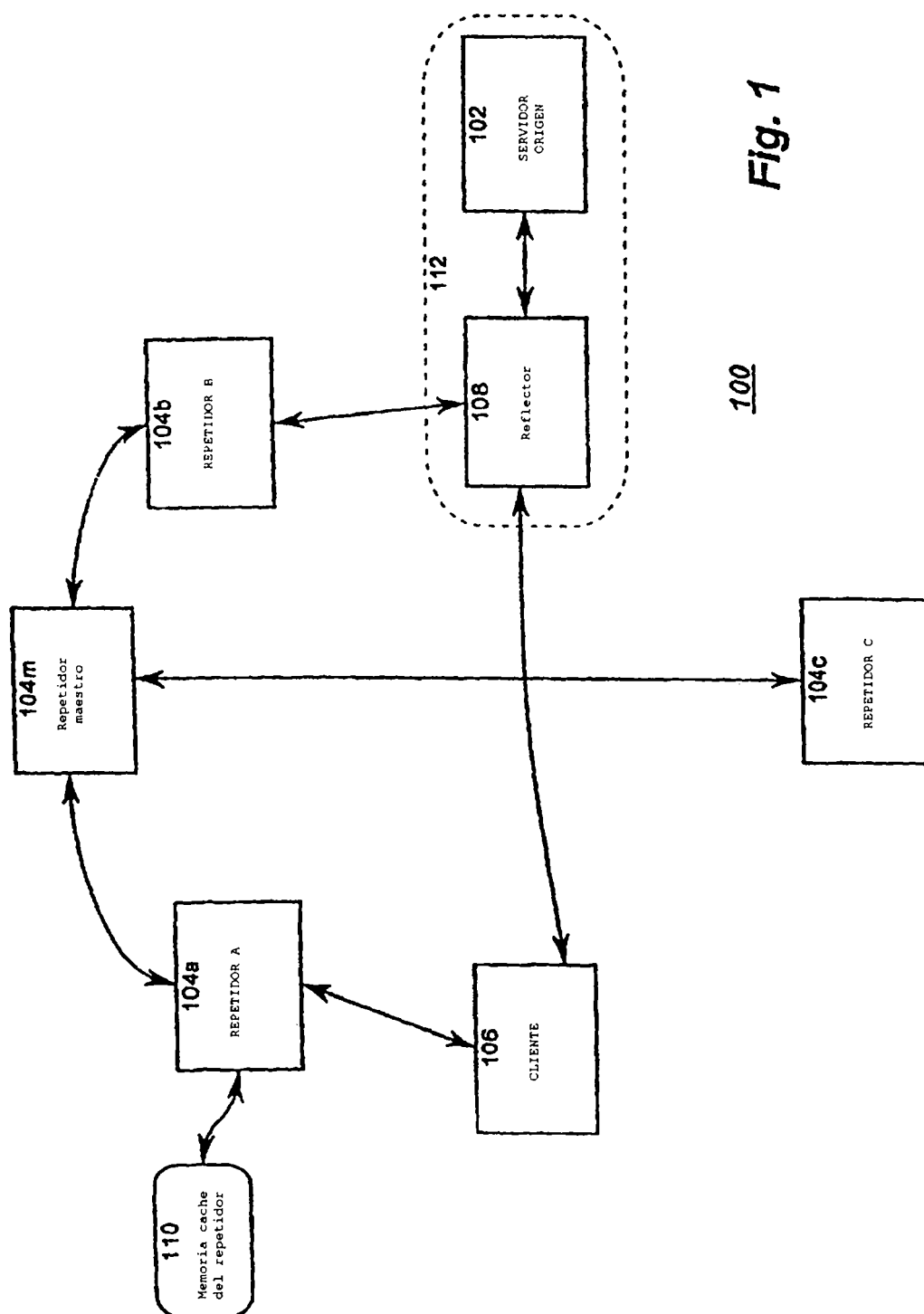


Fig. 1

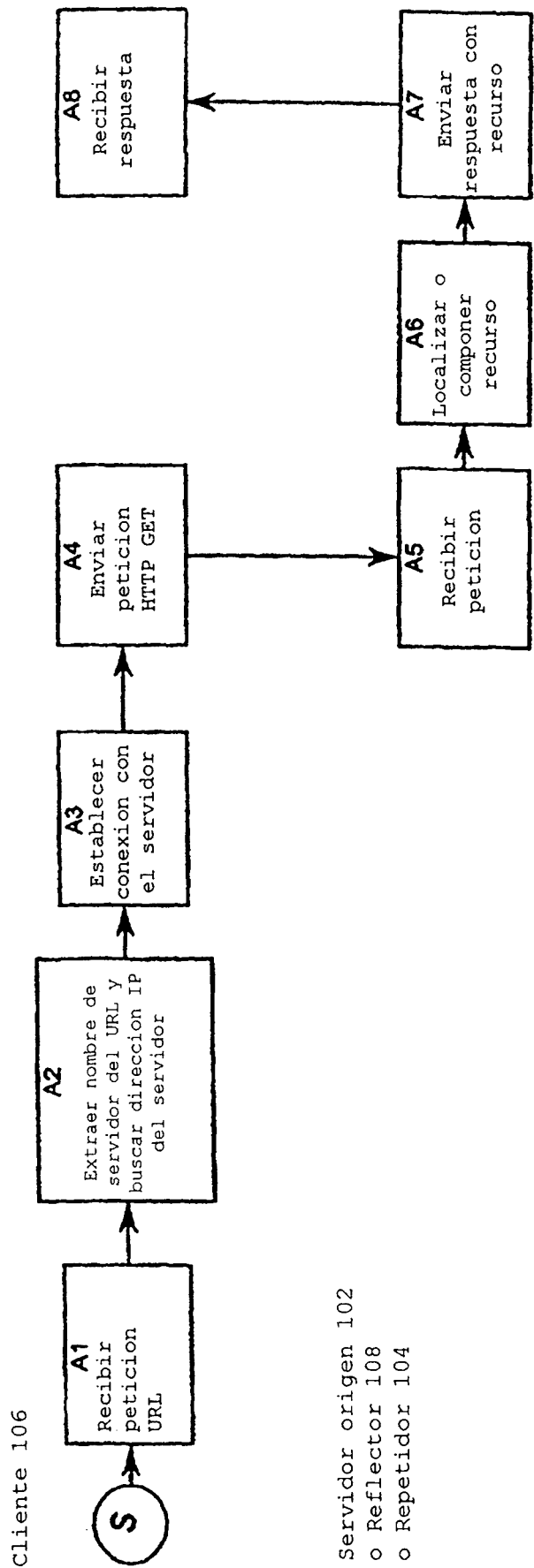
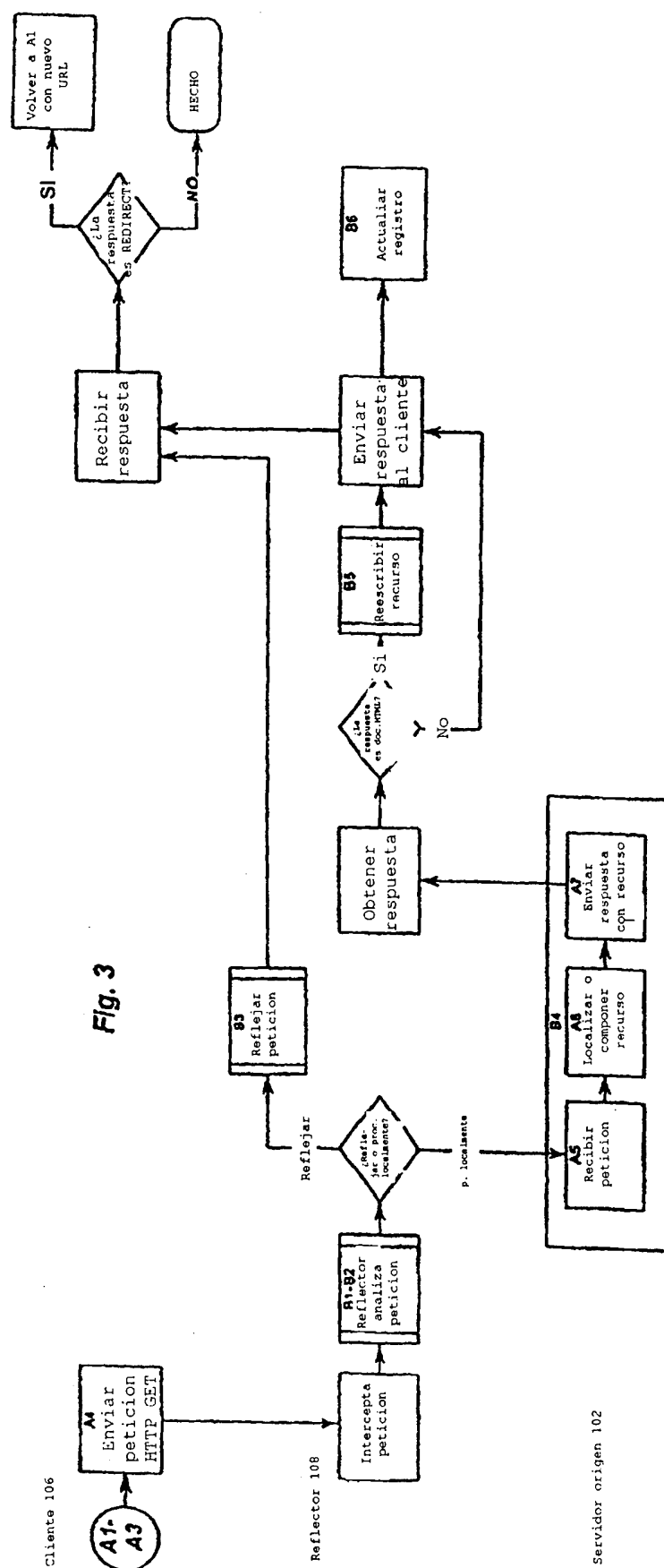


Fig. 2



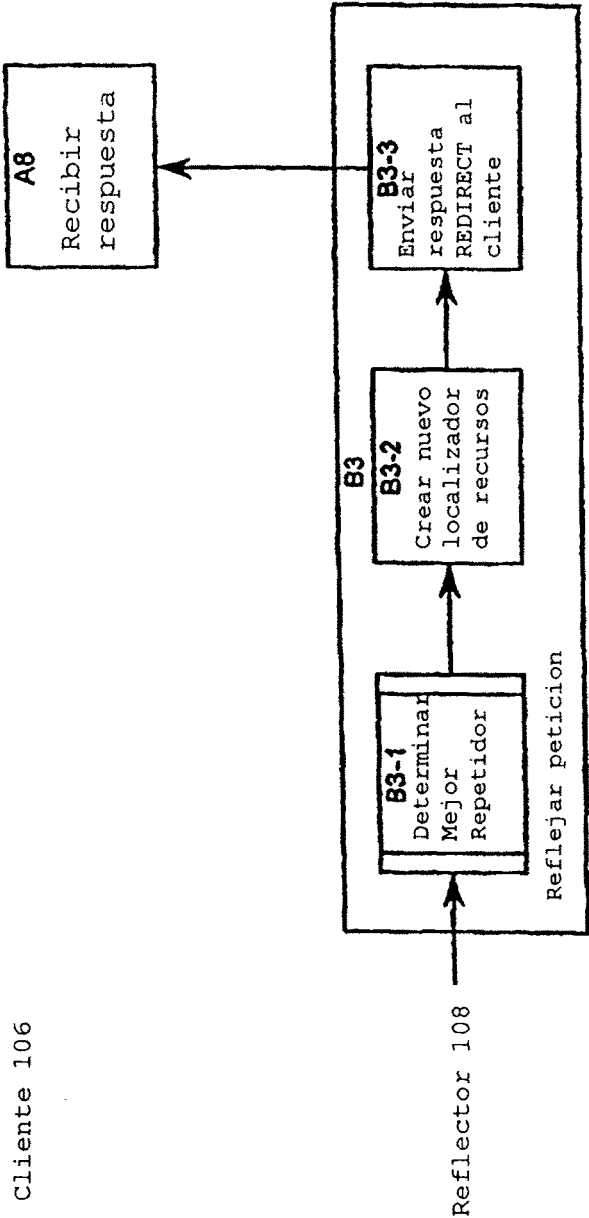


Fig. 4

Cliente 106

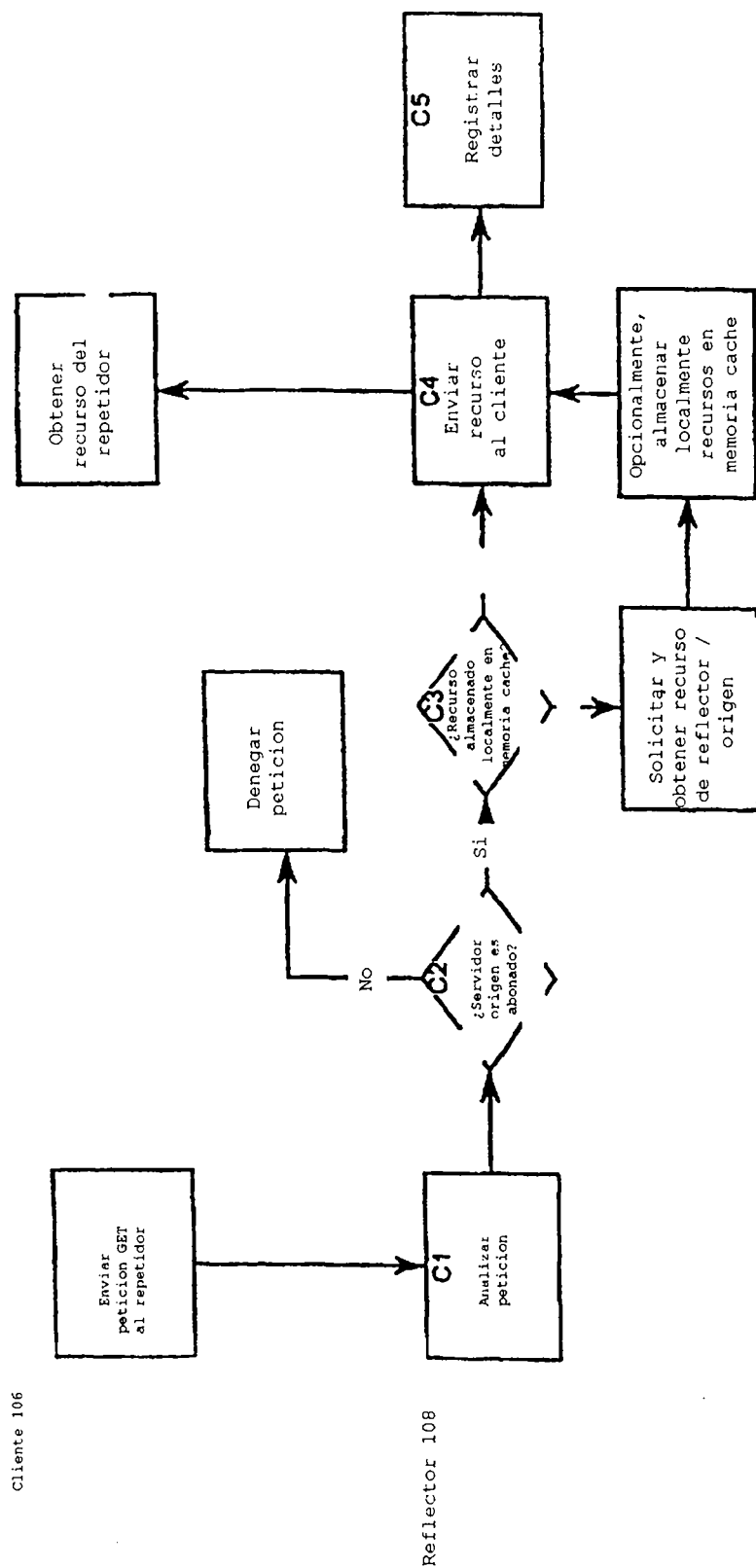


Fig. 5

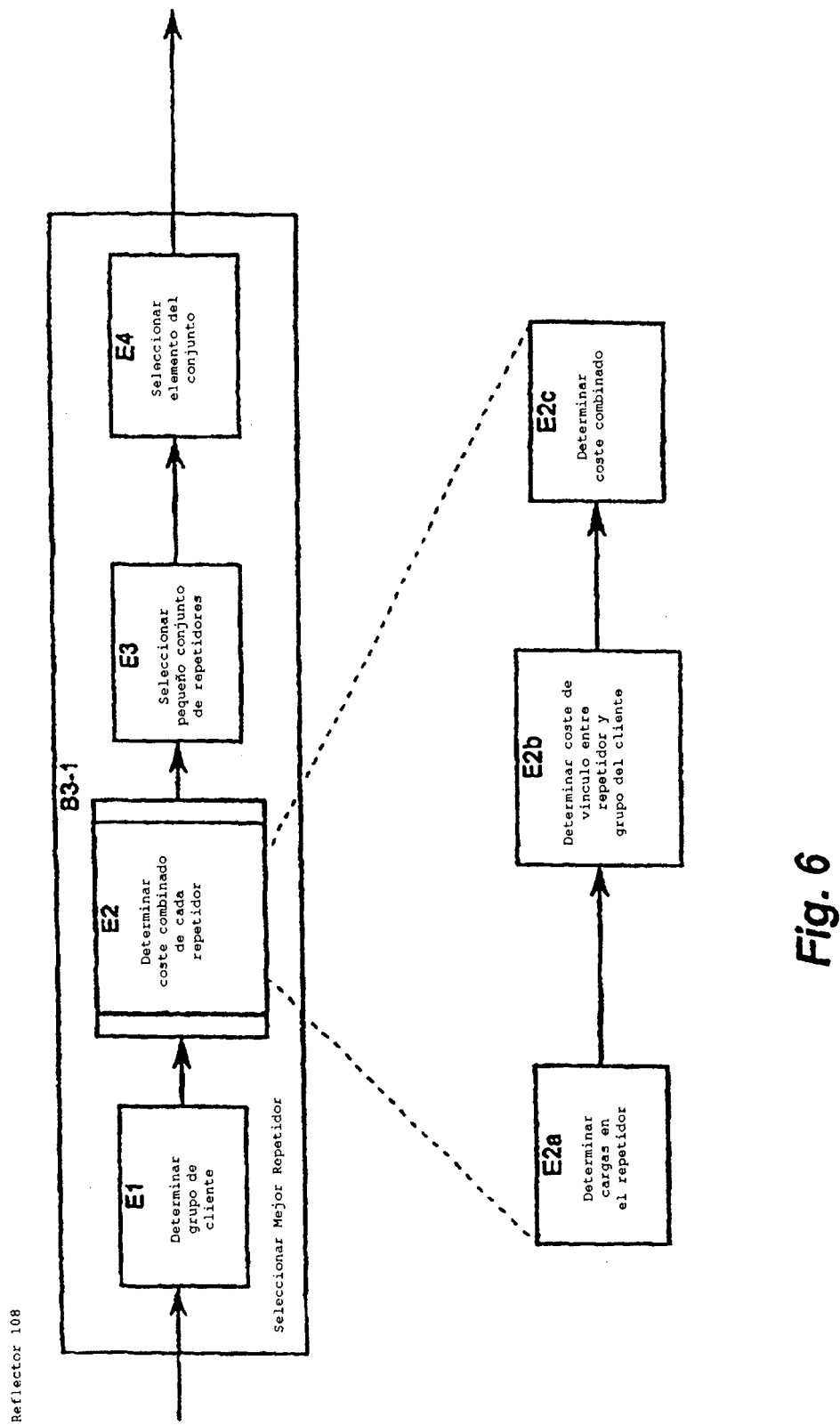


Fig. 6