



19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

11 Número de publicación: **2 328 426**

51 Int. Cl.:
G06F 9/46 (2006.01)
H04L 29/06 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **99906680 .6**
96 Fecha de presentación : **09.02.1999**
97 Número de publicación de la solicitud: **1053524**
97 Fecha de publicación de la solicitud: **22.11.2000**

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

30 Prioridad: **10.02.1998 US 21506**

45 Fecha de publicación de la mención BOPI:
12.11.2009

45 Fecha de la publicación del folleto de la patente:
12.11.2009

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 2 328 426 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

DESCRIPCIÓN

Localización optimizada de recursos de red.

5 1. Área de la invención

La presente invención hace referencia a la duplicación de recursos en redes informáticas.

10 2. Antecedentes de la invención

15 La llegada de las redes informáticas globales, como por ejemplo Internet, ha generado maneras completamente nuevas y diferentes de obtener información. Un usuario de Internet ahora puede acceder a la información desde cualquier lugar del mundo, sin importar la ubicación real del usuario ni de la información. Un usuario puede obtener información sólo conociendo una dirección de red para esa información y proporcionando dicha dirección a un programa de aplicación apropiado, como un navegador de red.

20 El rápido crecimiento de la popularidad de Internet ha impuesto una gran carga de tráfico en toda la red. Las soluciones a problemas de demanda (por ejemplo, una mejor accesibilidad y enlaces de comunicación más rápidos) sólo logran incrementar los niveles de exigencia de carga sobre el suministro de servicio. Los sitios web de Internet (aquí denominados “editores”) deben gestionar necesidades de ancho de banda cada vez mayores, adaptarse a los cambios dinámicos de carga, y mejorar el rendimiento para aquellos clientes navegadores distantes, especialmente para los que se encuentran en otros continentes. La adopción de aplicaciones ricas en contenido, como por ejemplo audio y vídeo en vivo, ha agravado el problema aún más.

25 Para hacer frente a las necesidades básicas de crecimiento de ancho de banda, un editor web se suscribe, por lo general, a un ancho de banda adicional proporcionado por un proveedor de servicios de Internet (ISP, por sus siglas en inglés), ya sea en forma de “cableados o tubos” adicionales, o más largos, o canales desde el ISP hasta las instalaciones del editor, o en la forma de un mayor compromiso de ancho de banda en un conjunto de servidores de hospedaje (de hosting) remoto del ISP. Estos incrementos no siempre son tan minuciosos como necesita el editor, y bastante a menudo 30 los tiempos de espera pueden causar que la capacidad del sitio web del editor se quede atrás con respecto a la demanda.

35 Para hacer frente a problemas más serios de crecimiento de ancho de banda, los editores pueden desarrollar soluciones personalizadas más complejas y costosas. La solución más común al requerimiento, es decir incrementar la capacidad, se basa por lo general en la duplicación de los recursos de hardware y el contenido del sitio (a lo que se denomina creación de réplicas o “mirroring”), y en la duplicación de los recursos de ancho de banda. Estas soluciones, sin embargo, son difíciles y costosas de utilizar y ofrecer. Como resultado, sólo los editores más importantes pueden utilizarlas, dado que sólo dichos editores pueden amortizar los costes sobre un gran número de usuarios (y consultas al sitio web).

40 Se ha desarrollado una gran cantidad de soluciones para promover la duplicación y la técnica de “mirroring”. En general, estas tecnologías están diseñadas para ser utilizadas por un solo sitio web y no incluyen elementos que permitan que sus componentes sean compartidos por muchos sitios web de manera simultánea.

45 Ciertos mecanismos de solución ofrecen software de duplicación que permite mantener actualizados los servidores espejados. Por lo general, estos mecanismos operan haciendo una copia completa de un sistema de archivos. Un sistema de este tipo, opera manteniendo sincronizadas de manera transparente múltiples copias de un sistema de archivos. Otro sistema proporciona mecanismos para copiar de manera explícita y regular archivos que han cambiado. Los sistemas de bases de datos son particularmente difíciles de espejar dado que cambian de manera continua. Varios 50 mecanismos permiten la duplicación de bases de datos, aunque no se cuenta con métodos estándar para lograrlo. Muchas compañías que ofrecen cachés de proxy los describen como herramientas de duplicación. Sin embargo, los cachés proxy difieren de éstas en que son operados en nombre de clientes y no de los editores.

55 Una vez que un sitio web es servido por múltiples servidores, asegurar que la carga sea distribuida o equilibrada correctamente entre los servidores es todo un desafío. La resolución de direcciones round-robin basadas en el servidor de nombres de dominio provoca que diferentes clientes sean direccionados a diferentes réplicas o “mirrors”.

Otra solución, la distribución compensada de carga, tiene en cuenta la carga de cada servidor (medida de diversas maneras) para determinar cuál de todos los servidores debería gestionar una petición en particular.

60 Los compensadores de carga utilizan una variedad de técnicas para enrutar la petición al servidor apropiado. La mayoría de dichas técnicas de distribución compensada de carga requiere que cada servidor sea una réplica exacta del sitio web primario. Los compensadores de carga no tienen en cuenta la “distancia de red” que existe entre el cliente y los posibles servidores espejo.

65 Suponiendo que los protocolos de cliente no puedan cambiar con facilidad, existen dos problemas principales en la utilización de recursos duplicados. El primero es cómo seleccionar la copia del recurso que se deberá utilizar. Es decir, cuando se efectúa una petición de un recurso a un solo servidor, cómo debería realizarse la elección de una

réplica del servidor (o de datos). A este problema se le denomina “problema de rendezvous”. Existe una gran cantidad de formas de hacer que los clientes se encuentren en servidores espejo distantes. Estas tecnologías, al igual que los compensadores de carga, deben enrutar una petición a un servidor apropiado, pero a diferencia de los compensadores de carga, tienen en cuenta el rendimiento y la topología de la red al realizar la determinación.

5 Muchas compañías ofrecen productos que mejoran el rendimiento de la red priorizando y filtrando el tráfico de red.

10 Los cachés de proxy proporcionan una manera para que los agregadores de los contenidos de los clientes puedan reducir el consumo de los recursos de red mediante el almacenamiento de copias de recursos populares cerca de los usuarios finales. Un cliente agregador es un proveedor de servicio de Internet u otra organización que incorpora una gran cantidad de clientes que operan navegadores de Internet. Los agregadores de cliente pueden utilizar cachés de proxy para reducir el ancho de banda requerido para proporcionar contenido web a estos navegadores. Sin embargo, los cachés de proxy tradicionales son operados en nombre de clientes web y no de los editores web.

15 Los cachés de proxy almacenan los recursos más populares de todos los editores, lo que significa que deben ser considerablemente grandes para alcanzar una eficiencia de caché razonable. (La eficiencia de un caché se define como la cantidad de peticiones de recursos que ya están almacenados dividida por la cantidad total de peticiones).

20 Los cachés de proxy dependen de indicaciones (hints) del control de caché entregadas con los recursos para determinar cuándo se deberían reemplazar los recursos. Estas indicaciones son predictivas, y a menudo son necesariamente incorrectas, por lo que los cachés de proxy con frecuencia proporcionan datos obsoletos. En muchos casos, los operadores de caché de Proxy le ordenan a su proxy que ignore las indicaciones con el fin de lograr que el caché sea más eficiente, incluso aunque esto cause que sirva, con una mayor frecuencia, datos caducados.

25 Los cachés de proxy ocultan a los editores la actividad de los clientes. Una vez que un recurso es almacenado, el editor no tiene forma de saber cuán a menudo se accedió a él desde el caché.

Los sistemas de caché son conocidos en el arte anterior en distintas modificaciones a partir de:

30 **Povey D. Harrison J:** “A Distributed Internet Caché”, Twentieth Australasian Computer Science Conference. ASCS’97, 5-7 Feb. 1997, Sydney, NSW, Australia, vol. 19, no. 1, páginas 176-184, XP000918489 Australian Computer Science Communications James Cook Univ. Australia ISSN: 0157-3055.

35 **Chankhunthod A et al:** “A Hierarchical Internet Object Caché”, Proceedings of the USENIX Annual Technical Conference, 22 de enero de 1996 (1996-01-22), páginas 153-163, XP0000918501.

40 **Wessels D et al:** “Application of Internet Caché Protocol (ICP), version 2-RFC2187” IETF Standard, Internet Engineering Task Force, IETF, CH, septiembre de 1997 (1997-09), XP015007971.

Malpani R et al: “Making World Wide Web Caching Servers Cooperate”, World Wide Web Journal, Sebastopol, CA, US, diciembre de 1995 (1995-12), páginas 107-117, XP000994815 ISSN: 1085-2301.

45 **Luotonen A, Altis K:** “World-Wide Web proxies” Computer Networks and ISDN Systems, vol. 27, no. 2, noviembre de 1994 (1994-11), páginas 147-154, XP004037985 ISSN: 0167-7552.

50 El objetivo de la presente invención es proporcionar un método mejorado para el procesamiento de peticiones de recursos en una red informática y una red informática para implementar el método. Este objetivo es alcanzado por la invención tal y como se define en la reivindicación 1 y en la reivindicación 22. Las realizaciones de la invención se definen en las reivindicaciones dependientes.

55 La presente invención proporciona a los servidores en una red informática una manera de descargar su procesamiento de peticiones de recursos para recursos específicos determinando un servidor diferente (un “repetidor”) que procese dichas peticiones. La selección del repetidor puede realizarse de manera dinámica, basándose en la información acerca de los posibles repetidores.

Si un recurso solicitado contiene referencias a otros recursos, algunas o todas estas referencias pueden ser reemplazadas por referencias a repetidores.

60 En primer lugar, un cliente realiza una petición de un recurso determinado desde un servidor origen, dicha petición que incluye un identificador de recursos para el recurso particular, y dicho identificador de recursos que a veces incluye una indicación del servidor origen. Las peticiones que llegan al servidor origen, no siempre incluyen una indicación del servidor origen; dado que son enviadas al servidor origen, no necesitan nombrarlo. Un mecanismo, denominado reflector, situado junto al servidor origen, intercepta la petición del cliente al servidor origen y decide si reflejar la petición o gestionarla de manera local. Si el reflector decide gestionar la petición localmente, la reenvía al servidor origen, de lo contrario selecciona un “mejor” repetidor para que procese la petición. Si la petición es reflejada, al cliente se le proporciona un identificador de recursos modificado que designa al repetidor.

ES 2 328 426 T3

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

5 Cuando el repetidor recibe la petición del cliente, le responde devolviendo el recurso solicitado al mencionado cliente. Si el repetidor tiene una copia local del recurso, entonces devuelve esa copia, de lo contrario reenvía la petición al servidor origen para obtener el recurso, y guarda una copia local del recurso para servirlo en peticiones posteriores.

10 La selección por parte del reflector de un repetidor apropiado para que gestione la petición puede llevarse a cabo de diversas maneras. En la realización preferente, dicha selección se realiza pre-particionando, en primer lugar, 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 de entre una pluralidad de repetidores en la red, teniendo los miembros de dicho conjunto un bajo coste en relación con el grupo de coste en el que se encuentra el cliente. Con el fin de determinar el coste más bajo, es necesario mantener y actualizar de manera regular una tabla para definir el coste entre cada grupo y cada repetidor. Luego se selecciona un miembro del conjunto, preferentemente al azar, como el mejor repetidor.

15 Si el recurso particular solicitado contiene en sí mismo identificadores de otros recursos, entonces es posible reescribir el recurso (antes de ser suministrado al cliente). En particular, el recurso es re-escrito para reemplazar, al menos, algunos de los identificadores de recursos, contenidos en el mismo, con identificadores de recursos modificados que designen un repetidor en lugar del servidor origen. Como resultado de este proceso de re-escritura, cuando el cliente solicita otros recursos basados en identificadores en el recurso particular solicitado, el cliente efectuará dichas peticiones de forma directa al repetidor seleccionado, eludiendo por completo al reflector y al servidor origen.

20 La re-escritura de recursos debe ser llevada a cabo por los reflectores. También puede ser realizada por los repetidores, en el caso de que los repetidores "se vean" entre ellos y hagan copias de recursos que incluyan identificadores de recursos re-escritos que designen a un repetidor.

25 En una realización preferente, la red es Internet y el identificador de recursos es un localizador uniforme de recursos (URL) para designar recursos en Internet, y el identificador de recursos modificado es un URL que designa al repetidor e indica el servidor origen (como se describe en el paso B3 a continuación), y el identificador de recursos modificado es suministrado al cliente utilizando un mensaje de REDIRECCIÓN. Se debe tener en cuenta que sólo cuando el reflector está "reflejando" una petición, se proporciona el identificador de recursos modificado utilizando un mensaje de REDIRECCIÓN.

30 En otro aspecto, la presente invención es una red informática que consta de una pluralidad de servidores de origen, teniendo al menos algunos de los servidores de origen reflectores asociados a ella, y una pluralidad de repetidores.

Breve descripción de los dibujos

40 Los anteriormente mencionados y otros objetos y ventajas de la invención, resultarán evidentes al considerar la siguiente descripción detallada, analizada en conjunto con los dibujos anexos, en la cual los caracteres de referencia indican partes iguales en todos los casos y en la cual:

45 la Figura 1 representa una porción 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 actuales realizaciones preferentes a modo de ejemplo

50 *Resumen*

La Figura 1 muestra una porción de un entorno de red 100 según la presente invención, en donde un mecanismo (un reflector 108, descrito en detalle a continuación) en un servidor (aquí, servidor origen 102) mantiene y lleva un registro de diversos servidores o repetidores parcialmente espejados 104a, 104b y 104c. Cada repetidor 104a, 104b y 104c espeja parte o toda la información disponible en el servidor origen 102 como así también la información disponible en otros servidores de origen en la red 100. El reflector 108 está conectado a un repetidor particular conocido como su repetidor "de contacto" ("Repetidor B" 104b en el sistema representado en la Figura 1). De manera preferente, cada reflector mantiene una conexión con un solo repetidor conocido como su contacto, y cada repetidor mantiene una conexión con un repetidor especial conocido como su repetidor principal (por ejemplo, 104m para los repetidores 104a, 104b y 104c en la Figura 1).

65 Por lo tanto, un repetidor puede ser considerado como un servidor proxy dedicado que mantiene una réplica parcial o pobre del servidor origen 102, mediante la implementación de un caché distribuido coherente 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 espejan recursos seleccionados proporcionados por servidores de origen en respuesta a las peticiones HTTP (protocolo de transmisión de hipertexto) y FTP (protocolo de transferencia de archivos) de los clientes.

ES 2 328 426 T3

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

El servidor origen 102 es un servidor en el cual se originan los recursos. De manera más general, el servidor origen 102 es cualquier proceso o grupo de procesos que proporciona recursos en respuesta a las peticiones de un cliente 106. El servidor origen 102 puede ser cualquier servidor web estándar. En una realización preferente, el servidor origen 102 es típicamente un servidor web como el servidor Apache o el servidor Enterprise™ de Netscape Communications Corporation.

El cliente 106 es un procesador que solicita recursos desde el servidor origen 102 en nombre de un usuario final. El cliente 106 es típicamente un agente de usuario (por ejemplo, un navegador web como Navigator™ de Netscape Communications Corporation) o un proxy para un agente de usuario. El resto de los componentes, aparte del reflector 108 y los repetidores 104a, 104b, etc., pueden implementarse utilizando programas de software comúnmente disponibles. En particular, esta invención funciona con cualquier cliente HTTP (por ejemplo, un navegador web), caché de proxy y servidor web. Además, el reflector 108 podría estar completamente integrado en el servidor de datos 112 (por ejemplo, en un servidor web). Estos componentes podrían estar integrados en una estructura modular, basados en el uso de mecanismos de expansión (como los denominados módulos de complemento o añadidos), o podrían estar sólidamente integrados mediante la modificación del componente de servicio para que sirva de manera específica al soporte de los repetidores.

Los recursos que se originan en el servidor origen 102 pueden ser estáticos o dinámicos. Es decir, los recursos pueden estar fijados o pueden ser creados por el servidor origen 102 específicamente en respuesta a una petición. Se debe tener en cuenta que los términos “estático” y “dinámico” son relativos, dado que un recurso estático podría cambiar en algún intervalo regular, aunque prolongado.

Las peticiones de recursos del cliente 106 al servidor origen 102 son interceptadas por el reflector 108, el cual, ante una petición dada, o bien reenvía la petición al servidor origen 102 o la refleja de manera condicional a algún repetidor 104a, 104b, etc., en la red 100. Es decir, dependiendo de la naturaleza de la petición realizada por el cliente 106 al servidor origen 102, el reflector 108 o bien sirve la petición de manera local (en el servidor origen 102), o selecciona uno de los repetidores (preferentemente, el mejor repetidor para esa tarea) y refleja la petición al repetidor seleccionado. En otras palabras, el reflector 108 ocasiona que las peticiones de recursos del servidor origen 102, efectuadas por el cliente 106, o bien sean servidas localmente por el servidor origen 102 o sean reflejadas de manera transparente al mejor repetidor 104a, 104b, etc. La noción de un repetidor óptimo y la manera en que se selecciona el mejor repetidor se describen en detalle a continuación.

Los repetidores 104a, 104b, etc. son procesadores intermediarios utilizados para servir las peticiones de clientes mejorando así la ejecución y reduciendo los costes de la manera aquí descrita. Dentro de los repetidores 104a, 104b, etc. se encuentran todos los procesos o grupos de procesos que suministran recursos al cliente 106 de parte del servidor origen 102. Un repetidor puede incluir un caché repetidor 110, utilizado para evitar las transacciones innecesarias con el servidor origen 102.

El reflector 108 es un mecanismo, de manera preferente un programa de software, que intercepta las peticiones que normalmente serían enviadas directamente al servidor origen 102. Aunque en los dibujos se muestran como componentes separados, el reflector 108 y el servidor origen 102 suelen estar en la misma ubicación, por ejemplo, en un sistema especial tal como el servidor de datos 112. (Como se discute a continuación, el reflector 108 puede ser incluso un módulo “de complemento o añadido” (o plug-in) que se vuelve parte del servidor origen 102).

La Figura 1 muestra sólo una parte de una red 100 según la invención. Una red completa operativa consta de una cantidad indeterminada de clientes, repetidores, reflectores y servidores de origen. Los reflectores se comunican con la red de repetidores, y los repetidores en la red se comunican entre sí.

Localizadores uniformes de recursos

Cada ubicación en una red informática tiene una dirección que por lo general puede especificarse como una serie de nombres o números. Para acceder a la información, es necesario conocer una dirección para esa información. Por ejemplo, en la red de la “telaraña mundial” (“la web”), que es un subconjunto de Internet, la forma en que se proporcionan las ubicaciones de las direcciones de la información se ha estandarizado en localizadores uniformes de recursos (URL, por sus siglas en inglés). Los URL especifican la ubicación de los recursos (información, archivos de datos, etc.) en la red.

La noción de URL resulta aún más útil al utilizar documentos de hipertexto. Un documento de hipertexto es aquel que incluye, dentro del propio documento, enlaces (indicadores o referencias) al documento en sí mismo o a otros documentos. Por ejemplo, en un sistema legal de investigación en línea, cada caso puede ser presentado como un documento de hipertexto. Cuando se citan otros casos, es posible proporcionar los enlaces a esos casos. De esta manera, mientras una persona lee un caso, puede seguir enlaces de citas para leer las partes correspondientes de los casos citados.

ES 2 328 426 T3

En el caso de Internet en general y de la red web mundial en particular, es posible crear documentos utilizando una forma estandarizada conocida como lenguaje descriptor de hipertexto (HTML, por sus siglas en inglés). En HTML, un documento consta de datos (texto, imágenes, sonidos, etc.), incluyendo enlaces a otras secciones del mismo documento o a otros documentos. Los enlaces suelen ser proporcionados como URL, y pueden estar expresados en forma absoluta o relativa. Los URL relativos simplemente omiten las partes de la URL que son iguales a las del documento que incluye el enlace, como la dirección del documento (cuando enlazan al mismo documento), etc. En general, un programa navegador se encargará de completar las partes que faltan de un URL utilizando las partes correspondientes del documento actual, formando así un URL completo que incluya un nombre de dominio completo, etc.

Un documento de hipertexto puede contener cualquier cantidad de enlaces a otros documentos, y cada uno de esos otros documentos pueden encontrarse en un servidor diferente en otro lugar del mundo. Por ejemplo, un documento puede contener enlaces a documentos en Rusia, África, China y Australia. Un usuario que visualiza ese documento en un cliente particular puede seguir cualquiera de los enlaces de manera transparente (es decir, sin saber dónde reside en realidad el documento al cual se enlaza). En consecuencia, el coste (en términos de tiempo o dinero o asignación de recursos) de seguir un enlace contra otro puede resultar bastante significativo.

Los URL suelen tener la siguiente forma (definida en detalle en T. Berners-Lee *et al*, Uniform Resource Locators (URL), Network Working Group, Request for Comments: 1738, Categoría: Standards Track, diciembre de 1994, ubicada en “<http://ds.internic.net/rfc/rfc1738.txt>”, la cual se incorpora aquí por el presente documento por referencia):

esquema: // host [:puerto] /ruta-url

donde “esquema” puede ser un símbolo tal como “archivo” (para un archivo en el 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 a un servicio basado en Telnet). También pueden utilizarse otros esquemas, y se agregan esquemas nuevos constantemente. El número de puerto es opcional, sustituyéndolo el sistema un número de puerto predeterminado (dependiendo del esquema) si no se proporciona ninguno. El campo “host” se asocia a una dirección particular de red de un ordenador en particular. La “ruta-url” es relativa al ordenador especificado en el campo “host”. Una ruta URL es por lo general, pero no necesariamente, el nombre de ruta de un archivo en un directorio de servidor web.

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

Para acceder al archivo “F” (el recurso) especificado por la URL anterior, un programa (por ejemplo, un navegador) ejecutándose en el ordenador de un usuario (es decir, un ordenador cliente) debería localizar primero el ordenador (es decir, un ordenador servidor) especificado por el nombre del sistema central. En otras palabras, el programa debería localizar el servidor “www.uspto.gov”. Para hacerlo, accedería a un servidor de nombres de dominio (DNS), proporcionando el DNS con el nombre del sistema central (“www.uspto.gov”). El DNS actúa como una especie de directorio centralizado para resolver direcciones a partir de nombres. Si el DNS determina que existe un ordenador (servidor remoto) correspondiente 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 dirección de protocolo de Internet (o IP) y presenta la forma “123.345.456.678”. El programa en el ordenador (cliente) del usuario utilizaría entonces la dirección real para acceder al ordenador (servidor) remoto.

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 (utilizando el esquema HTTP). El mensaje es típicamente 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 devuelve el recurso a través de la misma conexión.

Por lo tanto, de manera convencional las peticiones de clientes HTTP de recursos web en un servidor origen 102 son procesadas de la siguiente manera (ver Figura 2). (Esta es una descripción del proceso en el que no está instalado ningún reflector 108):

- A1. Un navegador (por ejemplo, Netscape’s Navigator) en un cliente recibe un identificador de recursos (es decir, un URL) de parte de un usuario.
- A2. El navegador extrae el nombre del hospedador (servidor origen) del identificador 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, de haber alguno, o utiliza un número de puerto predeterminado (el nombre de puerto predeterminado para peticiones http es 80).
- A3. El navegador utiliza la dirección de red y el número de puerto del servidor para establecer una conexión entre el cliente 106 y el hospedador o servidor origen 102.
- A4. El cliente 106 envía después una petición (GET) a través de la conexión identificando el recurso solicitado.

ES 2 328 426 T3

A5. El servidor origen 102 recibe la petición y

A6. sitúa o compone el recurso correspondiente.

5 A7. El servidor origen 102 le devuelve al cliente 106 una respuesta que contiene el recurso solicitado (o alguna forma de indicador de error si el recurso no está disponible). La respuesta es enviada al cliente a través de la misma conexión mediante la cual la petición fue recibida de parte del cliente.

A8. El cliente 106 recibe la respuesta desde el servidor origen 102.

10

Existen diversas variaciones de este modelo básico. Por ejemplo, en una variación, en lugar de proporcionar el recurso al cliente, el servidor origen puede decirle al cliente que vuelva a solicitar el recurso por medio de otro nombre. Para hacerlo, en A7 el servidor 102 devuelve al cliente 106 una respuesta denominada “REDIRECCIÓN” que contiene un nuevo URL que indica otro nombre. Después, el cliente 106 repite toda la secuencia, por lo general sin ninguna intervención del usuario, solicitando esta vez el recurso identificado por el nuevo URL.

15

Funcionamiento del sistema

En la presente invención, el reflector 108 ocupa de manera efectiva el lugar de un servidor web habitual o del servidor origen 102. El reflector 108 logra hacerlo asumiendo la dirección IP y el número de puerto del servidor origen. De esta manera, cuando un cliente intenta conectarse al servidor origen 102, éste en realidad se conectará con el reflector 108. El servidor web original (o servidor origen 102) debe entonces aceptar las peticiones en una dirección de red (IP) diferente, o en la misma dirección IP pero en un número de puerto diferente. Por lo tanto, utilizando esta invención, el servidor al que se hace referencia en A3-A7 con anterioridad es en realidad un reflector 108.

20

Se debe tener en cuenta que también es posible dejar intacta la dirección de red del servidor origen 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 direccionadas específicamente al reflector. Por lo tanto, el sistema puede ser probado y configurado sin interrumpir su funcionamiento normal.

25

El reflector 108 soporta el procesamiento de la siguiente manera (ver Figura 3): al recibir una petición,

30

B1. El reflector 108 analiza la petición para determinar si debe 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 emitidas por los repetidores deben ser servidas de manera local por el servidor origen 102. Esta determinación puede realizarse buscando la dirección de red (IP) del emisor en una lista de direcciones de red (IP) de repetidores conocidas. De manera alternativa, podría efectuarse esta determinación adjuntando información a la petición para indicar que la petición proviene de un repetidor específico, o los repetidores pueden solicitar recursos desde un puerto especial que no sea el utilizado por los clientes comunes.

35

40

B2. Si la petición no proviene de un repetidor, el reflector busca el recurso solicitado en una tabla (denominada “base de reglas”) para determinar si el recurso solicitado es “repetible”. Basándose en esta determinación, el reflector o bien refleja la petición (B3, descrita a continuación), o sirve la petición de manera local (B4, descrita a continuación).

45

La base de reglas es una lista de expresiones regulares y atributos asociados. (Las expresiones regulares son bien conocidas en el campo de la informática. Puede encontrarse una acotada bibliografía acerca de su uso en Aho, *et al.*, “Compilers, Principles, techniques and tools”, Addison-Wesley, 1986, páginas 157-158). El identificador de recursos (URL) para una petición dada es buscado en la base de reglas emparejándolo secuencialmente con cada expresión regular. La primera concordancia identifica los atributos para el recurso, a saber repetible o local. Si no existe ninguna concordancia en la base de reglas, se utiliza un atributo predeterminado. Cada reflector tiene su propia base de reglas, la cual es configurada de forma manual por el operador del reflector.

50

B3. Para reflejar una petición, (para servir una petición de manera local remitirse a B4), como se muestra en la Figura 4, el reflector determina (B3-1) el mejor repetidor al que reflejar la petición, como se describe en detalle a continuación. El reflector entonces crea (B3-2) un nuevo identificador de recursos (URL) (utilizando la URL solicitado y el mejor repetidor) que identifique el mismo recurso en el repetidor seleccionado.

55

60

Es necesario que el paso de reflexión cree un solo URL que contenga la URL del recurso original, como así también la identidad del repetidor seleccionado. Una forma especial de URL se crea para proporcionar esta información. Esto se lleva a cabo creando un nuevo URL de la siguiente manera:

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

65

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

ES 2 328 426 T3

Si la forma utilizada no es “http”, la realización preferente utiliza el siguiente formato:

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

El reflector también puede adjuntar un tipo MIME a la petición, para hacer que el repetidor le proporcione el resultado a ese tipo MIME. Esto resulta útil porque muchos protocolos (como FTP) no proporcionan una manera de adjuntar un tipo MIME a la petición. El formato es

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

Este URL es interpretado al ser recibido por el repetidor.

El reflector envía entonces (B3-3) al cliente solicitante una respuesta REDIRECCIÓN que contiene este nuevo URL. El comando HTTP REDIRECCIÓN permite que el reflector envíe al navegador un solo URL para reintentar la petición.

B4. Para servir una petición de manera local, la petición es enviada por el reflector (“reenviada a”) al servidor origen 102. En este modo, el reflector actúa como un servidor proxy inverso. El servidor origen 102 procesa la petición en la forma habitual (A5-A7). El reflector obtiene entonces la respuesta del servidor origen a la petición, la cual es inspeccionada para determinar si el recurso solicitado es un documento HTML, es decir, si el recurso solicitado es un recurso que contiene en sí mismo identificadores de recursos.

B5. Si el recurso es un documento HTML, entonces el reflector reescribe el documento HTML modificando los identificadores de recursos (URL) dentro de éste, como se describe a continuación. El recurso, posiblemente tal y como fue modificado por la re-escritura, es luego devuelto en una respuesta al cliente 106 que realiza la petición.

Si el cliente solicitante es un repetidor, el reflector podría deshabilitar de manera temporal cualquier modificador de control de caché al que el servidor origen adjuntó la respuesta. Estos modificadores de control de caché deshabilitados son rehabilitados posteriormente cuando el contenido es servido desde el repetidor. Este mecanismo permite que el servidor origen evite que los recursos sean almacenados en cachés de proxy normales, sin afectar el comportamiento del caché en el repetidor.

B6. Ya sea la petición reflejada o servida de manera local, los detalles acerca de la transacción, como la hora actual, la dirección del solicitante, la URL solicitado, y el tipo de respuesta generada, son escritas por el reflector en un archivo de registro local.

Utilizando una base de reglas (B2), es posible reflejar recursos de manera selectiva. Existen distintas razones por las que ciertos recursos particulares no pueden ser repetidos de manera efectiva (y por lo tanto no deberían ser reflejados), como por ejemplo:

- el recurso es compuesto únicamente para cada petición;

- el recurso depende de lo que se denomina *cookie* (los navegadores no enviarán cookies a repetidores con diferentes nombres de dominios);

- el recurso es en realidad un programa (como una aplicación Java) 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 suministró la aplicación).

Además, el reflector 108 puede ser configurado para que las peticiones de ciertas direcciones de red (por ejemplo, las peticiones de clientes en la misma red de área local que el propio reflector) nunca sean reflejadas. Asimismo, el reflector puede elegir no reflejar las peticiones porque el reflector está excediendo su tasa de información comprometida total, como se describe a continuación.

Una petición que es reflejada se espeja de manera automática en el repetidor cuando el repetidor recibe y procesa la petición.

La combinación del proceso de reflexión descrito aquí y el proceso de cachéo descrito a continuación, crea de forma efectiva un sistema en el cual los recursos repetibles son migrados a, y espejados en, el reflector seleccionado, mientras que los recursos no repetibles no son espejados.

Aproximación alternativa

Situar el nombre del servidor origen en la URL reflejado suele ser una buena estrategia, pero puede considerarse poco deseable por razones estéticas o (en el caso, por ejemplo, de las cookies) por ciertas razones técnicas.

ES 2 328 426 T3

Es posible evitar la necesidad de colocar tanto el nombre del repetidor como el nombre del servidor en la URL. En su lugar, podría crearse una “familia” de nombres para un servidor origen dado, con cada nombre identificando a uno de los repetidores utilizados por ese servidor.

5 Por ejemplo, si `mwv.ejemplo.com` es el servidor origen, pueden crearse nombres para tres repetidores:

`wr1.ejemplo.com`

`wr2.ejemplo.com`

10

`wr3.ejemplo.com`

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

15

Si el repetidor puede determinar por qué nombre fue direccionado, puede utilizar esta información (junto con una tabla que asocia los nombres de alias repetidores con los nombres de servidores de origen) para determinar a qué servidor origen está siendo direccionado. Por ejemplo, si el repetidor 1 es direccionado como `wr1.ejemplo.com`, entonces el servidor origen es “`www.ejemplo.com`”; si es direccionado como “`wr1.otroEjemplo.com`”, entonces el servidor origen es “`www.otroEjemplo.com`”.

20

El repetidor puede utilizar dos mecanismos para determinar por qué alias es direccionado:

25

1. Cada alias puede estar asociado con una dirección IP diferente. Desafortunadamente, esta solución no es la mejor, dado que las direcciones IP actualmente son escasas, y la cantidad requerida de direcciones IP crece a la vez que el producto de servidores de origen y repetidores.

30

2. El repetidor puede intentar determinar el nombre de alias utilizado inspeccionando la etiqueta “host:” en el encabezado HTTP de la petición. Lamentablemente, algunos buscadores antiguos aún en uso no adjuntan la etiqueta “host:” a la petición. Los reflectores necesitarían identificar dichos navegadores (la identidad del navegador es una parte de cada petición) y evitar esta forma de reflexión.

Cómo un repetidor gestiona una petición

35

Cuando un navegador recibe una respuesta REDIRECCIÓN (como se produce en B3), vuelve a emitir una petición del recurso utilizando el nuevo identificador de recursos (URL) (A1-A5). Dado que el nuevo identificador hace referencia a un repetidor en lugar de al servidor origen, el navegador envía ahora una petición del recurso al repetidor el cual procesa una petición de la siguiente manera, en referencia a la Figura 5:

40

C1. En primer lugar, el repetidor analiza la petición para determinar la dirección de red del cliente solicitante y la ruta del recurso solicitado. La ruta incluye el nombre de un servidor origen (como se describió con anterioridad en referencia a B3).

45

C2. El repetidor utiliza una tabla interna para verificar que el servidor origen pertenezca a un “suscriptor” conocido. Un suscriptor es una entidad (por ejemplo, una compañía) que publica recursos (por ejemplo archivos) a través de uno o más servidores de origen. Cuando la entidad suscribe, está permitido utilizar la red del repetidor. Las tablas del suscriptor descritas a continuación incluyen la información que es utilizada para enlazar los reflectores con los suscriptores.

50

Si la petición no es para un recurso perteneciente a un suscriptor conocido, la petición es rechazada. Para rechazar una petición, el repetidor devuelve una respuesta indicando que el recurso solicitado no existe.

55

C3. El repetidor determina entonces si el recurso solicitado está almacenado de forma local. Si el recurso solicitado se encuentra en el caché del repetidor, éste es recuperado. Por otra parte, si en el caché del repetidor no existe una copia válida del recurso solicitado, el repetidor modifica la URL entrante, creando una petición que emite directamente al reflector origen el cual se encarga de procesarla (como en B1-B6). Dado que esta petición al reflector origen le proviene de un repetidor, el reflector siempre devuelve el recurso solicitado en lugar de reflejar la petición. (Se debe recordar que los reflectores siempre gestionan las peticiones de los repetidores de manera local). Si el repetidor obtuvo el recurso del servidor origen, entonces el repetidor almacena el recurso de forma local.

60

Si un recurso no está almacenado en forma de caché local, el caché puede consultar a “partes de cachés” para ver si alguna de ellas contiene el recurso, antes o al mismo tiempo de solicitar el recurso desde el reflector/servidor origen. Si alguna parte de caché responde de forma positiva en un período limitado de tiempo (preferentemente, una pequeña fracción de segundo), el recurso será recuperado desde la parte de caché.

65

ES 2 328 426 T3

C4. El repetidor construye entonces una respuesta incluyendo el recurso solicitado (el cual fue recuperado desde el caché o desde el servidor origen) y envía esa respuesta al cliente solicitante.

C5. Los detalles acerca de la transacción, como el reflector asociado, la hora actual, la dirección del solicitante, la URL solicitado, y el tipo de respuesta generada, son escritos en un archivo de registro local en el repetidor.

Se debe tener en cuenta que la flecha inferior de la Figura 2 hace referencia a un servidor origen, o a un reflector, o a un repetidor, dependiendo de lo que la URL en el paso A1 identifique.

Selección del mejor repetidor

Si el reflector 108 determina que reflejará la petición, debe entonces seleccionar el mejor repetidor para gestionar esa petición (cómo se hizo referencia en el paso B3-1). Esta selección es llevada a cabo por el mecanismo selector de mejor repetidor (BRS, por sus siglas en inglés) descrito aquí.

El objetivo del BRS es seleccionar, de manera rápida y heurística, un repetidor apropiado para un determinado cliente teniendo únicamente la dirección de red del cliente. Un repetidor apropiado es un repetidor que no está demasiado cargado y que tampoco se encuentra demasiado lejos del cliente en términos de alguna medida de distancia de red. El mecanismo utilizado aquí depende de datos específicos, compactos y pre-calculados para poder tomar una rápida decisión. También es posible emplear otras soluciones dinámicas para seleccionar un repetidor apropiado.

El BRS depende de tres tablas pre-calculadas, a saber la Tabla de reducción de grupos, la Tabla de coste de enlaces, y la Tabla de carga. Estas tres tablas (descritas a continuación) son calculadas fuera de línea y descargadas a cada reflector por su contacto en la red de repetidores.

La Tabla de reducción de grupos sitúa cada dirección de red en un grupo, con el objetivo de que las direcciones de un grupo compartan costes relativos, de modo que tendrían el mismo mejor repetidor bajo condiciones diversas (es decir, el BRS es invariable con respecto a los miembros del grupo).

La Tabla de coste de enlaces es una matriz de dos dimensiones que especifica el coste actual entre cada repetidor y cada grupo. Inicialmente, el coste de enlace entre un repetidor y un grupo se define como el “coste de enlace normalizado” entre el repetidor y el grupo, como se define a continuación. Con el paso del tiempo, la tabla será actualizada con mediciones que reflejen de manera más exacta el coste relativo de transmitir un archivo entre el repetidor y un miembro del grupo. El formato de la Tabla de coste de enlaces es <identificación de grupo> <identificación de grupo> <coste de enlace>, donde las identificaciones de grupo son dadas como números AS.

La Tabla de carga es una Tabla de una sola dimensión que identifica la carga actual en cada repetidor. Dado que los repetidores pueden tener diferentes capacidades, la carga es un valor que representa la habilidad de un repetidor determinado de aceptar trabajo adicional. Cada repetidor envía su carga actual al repetidor principal central en intervalos regulares, preferentemente al menos una vez por minuto, de manera aproximada. El repetidor principal emite la Tabla de carga a cada reflector en la red, a través del repetidor de contacto.

Un reflector es provisto con entradas en la Tabla de carga sólo para los repetidores que está asignado a utilizar. La asignación de repetidores a reflectores es llevada a cabo de manera central por un operador de red de repetidores en el repetidor principal. Esta asignación hace que sea posible modificar el nivel de servicio de un reflector determinado. Por ejemplo, un reflector muy activo podría utilizar muchos repetidores, mientras que un reflector relativamente inactivo podría utilizar pocos repetidores.

Las tablas también pueden ser configuradas para proporcionar un servicio de repetidor selectivo a suscriptores de otras maneras, por ejemplo, para sus clientes en regiones geográficas específicas, como Europa o Asia.

Medición de la carga

En las actuales realizaciones preferentes, la carga del repetidor se mide en dos dimensiones, a saber

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

Para cada una de estas dimensiones se establece una configuración de capacidad máxima. La capacidad máxima indica el punto en el cual se considera que el repetidor está completamente cargado. Una mayor capacidad RRPT indica por lo general un procesador más rápido, mientras que una mayor capacidad BSPT indica por lo general un cable o “ancho de tubo” de red más ancho. Esta forma de medición de carga, supone que un servidor determinado está dedicado a la tarea de repetición.

ES 2 328 426 T3

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

5

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

$$10 \quad \text{carga-actual} = B \times \text{RRPT actual} + (1 - B) \times \text{BSPT actual}$$

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

15 El factor B, un valor entre 0 y 1, permite ajustar las cargas relativas de RRPT y BSPT, lo que favorece la consideración de la potencia de procesamiento o del ancho de banda.

Los valores de carga actual general y capacidad máxima general, son enviados de manera periódica desde cada repetidor al repetidor principal, donde son acumulados en la Tabla de carga, una tabla que resume la carga general de todos los repetidores. Los cambios en la Tabla de carga son distribuidos de manera automática a cada reflector.

20

Aunque la realización preferente emplea una medida de carga de repetidor de dos dimensiones, es posible utilizar cualquier otra medida de carga.

25 *Combinación de los costes de enlace y carga*

El BRS calcula el coste de dar servicio a un cliente determinado desde cada repetidor elegible. El coste es calculado combinando la capacidad disponible de cada repetidor posible con el coste del enlace entre ese repetidor y el cliente. El coste del enlace es calculado simplemente buscándolo en la Tabla de coste de enlaces.

30

El coste se determina utilizando la siguiente fórmula:

$$35 \quad \text{umbral} = K * \text{carga-máx}$$

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

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

40

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

En esta fórmula, e es un número muy pequeño (épsilon) y K es un factor de ajuste configurado inicialmente en 0.5. Esta fórmula hace que se incremente el coste a un repetidor dado, a una tasa definida por K , si su capacidad cae por debajo de un umbral configurable.

45

Dado el coste de cada repetidor posible, el BRS selecciona todos los repetidores dentro de un factor delta de la mejor calificación. De este conjunto se selecciona el resultado de forma aleatoria.

50 El factor delta evita que el BRS seleccione reiteradamente un solo repetidor cuando las calificaciones son similares. Este factor suele ser necesario porque la información disponible, acerca de los costes de carga y enlace, pierde exactitud con el paso del tiempo. Este factor es ajustable.

55 *Selector de mejor repetidor (BRS)*

El BRS funciona de la siguiente manera, en referencia a la Figura 6:

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

- 60 E1. Determinar en qué grupo se encuentra el cliente utilizando la Tabla de reducción de grupos.
- E2. Para cada repetidor en la Tabla de costes de enlaces y en la Tabla de carga, determinar el coste combinado de ese repetidor de la siguiente manera:
 - 65 E2a. Determinar la carga máxima y actual en el repetidor (utilizando la Tabla de carga).

ES 2 328 426 T3

E2b. Determinar el coste de enlace entre el repetidor y el grupo del cliente (utilizando la Tabla de coste de enlaces).

E2c. Determinar el coste combinado, como se describió con anterioridad.

E3. Seleccionar un pequeño conjunto repetidores con el coste más bajo.

E4. Seleccionar un miembro aleatorio del conjunto.

Preferentemente, los resultados del procesamiento BRS son conservados en un caché local en el reflector 108. Por lo tanto, si recientemente se ha determinado el mejor repetidor para un cliente determinado (es decir, para una dirección de red dada), ese mejor repetidor puede ser utilizado otra vez rápidamente sin ser re-determinado. Dado que el cálculo descrito arriba se basa en tablas estáticas y precomputadas, si las tablas no han cambiado entonces no existe la necesidad de volver a determinar el mejor repetidor.

Determinación de las tablas de reducción de grupos y coste de enlaces

La Tabla de reducción de grupo y la Tabla de coste de enlaces, utilizadas en el procesamiento BRS, son creadas y actualizadas de manera regular por medio de un procedimiento independiente, denominado de aquí en adelante NetMap. El procedimiento NetMap es llevado a cabo ejecutando diversas fases (descritas a continuación) según la necesidad.

El término Grupo se utiliza aquí en referencia a un “grupo de dirección” IP.

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

El término coste de enlace hace referencia a un coste determinado de manera estática para transmitir datos entre dos Grupos. En una actual implementación preferente, éste es el mínimo de las sumas de los costes de los enlaces a lo largo de cada ruta entre ellos. Los costes de enlace de principal preocupación aquí son los costes de enlace entre un Grupo y un Grupo repetidor.

El término coste de enlace relativo, hace referencia al coste de enlace en relación con otros costes de enlace para el mismo Grupo, el cual es calculado por la operación de diferencia entre el coste de enlace mínimo de un Grupo con cualquier Grupo repetidor de cada uno de sus costes de enlace, y un Grupo repetidor. El término Conjunto de costes hace referencia a un conjunto de Grupos que son equivalentes con respecto a la Selección de mejor repetidor. Es decir, dada la información disponible, el mismo repetidor sería seleccionado para cualquiera de ellos.

El procedimiento NetMap procesa primero archivos de entrada para crear una base de datos interna llamada Registro de grupo. Estos archivos de entrada describen los grupos, las direcciones IP dentro de los grupos, y los enlaces entre los grupos, y provienen de una variedad de fuentes, incluyendo bases de datos del registro de enrutamiento de Internet (IRR, por sus siglas en inglés) de uso público, tablas de rutas BGP, y servicios de análisis que están ubicados en distintos puntos de Internet y que utilizan herramientas de uso público (como las “traceroute” -trazador de rutas mediante paquetes-) para muestrear rutas de datos. Una vez que este procesamiento se completa, el Registro de grupo contiene información esencial utilizada para procesar posteriormente, a saber (1) la identidad de cada grupo, (2) el conjunto de direcciones IP en un grupo dado, (3) la presencia de enlaces entre grupos que indican rutas por las cuales la información puede viajar, y (4) el coste de enviar datos a través de un enlace dado.

Los siguientes procesos son entonces llevados a cabo en el archivo de Registro de grupo.

Cálculo de los costes de enlace del Grupo repetidor

El procedimiento NetMap calcula un “coste de enlace” por transmisión de datos entre cada Grupo repetidor y cada Grupo, en el Registro de grupo. Este coste de enlace general 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 enlaces individuales en la ruta. El algoritmo de coste de enlace presentado a continuación es básicamente el mismo algoritmo que el #562 del Transactions on Mathematical Software de la publicación ACM: “Shortest Path From a Specific Node to All Other Nodes in a Network” de U. Pape, ACM TOMS 6 (1980) páginas 450-455, <http://www.netlib.org/toms/562>.

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

Por cada Grupo repetidor T objetivo:

- Inicializar el coste de enlace entre T y sí mismo a cero.
- Inicializar el coste de enlace entre T y cada uno de los otros Grupos al infinito.

ES 2 328 426 T3

- Crear una lista L que contendrá Grupos que son equidistantes con respecto al Grupo repetidor T objetivo.
- Inicializar la lista L para que contenga sólo el Grupo repetidor T objetivo en sí mismo.
- 5 • Mientras que la lista L no está vacía:
 - Crear una lista vacía L' de vecinos de miembros de la lista L.
 - 10 • Para cada Grupo G en la lista L:
 - Para cada grupo N que es un vecino de G:
 - Permitir que el coste se refiera a la suma del coste de enlace entre T y G, y del coste de enlace entre G y N. El coste entre T y G fue determinado en el paso anterior del algoritmo; el coste de enlace entre G y N es del Registro de grupo.
 - 15 • Si el coste es inferior al coste de enlace entre T y N:
 - Establecer el coste de enlace entre T y N al coste.
 - 20 • Agregar N a L' si es que aún no está allí.
 - Ajustar L a L'.

Cálculo de los conjuntos de coste

Un Conjunto de coste es un conjunto de Grupos que son equivalentes con respecto a la Selección de mejor repetidor. Es decir, dada la información disponible, el mismo repetidor sería seleccionado para cualquiera de ellos.

El “perfil de costes” de un Grupo G se define aquí como el Conjunto de costes entre G y cada Repetidor. Dos perfiles de costes son equivalentes si los valores en un perfil difieren con respecto a los valores correspondientes en el otro perfil por una cantidad constante.

Una vez que se conoce un Grupo de cliente, el algoritmo de Selección de mejor repetidor depende del perfil de coste para obtener información acerca del Grupo. Si dos perfiles de costes son equivalentes, el algoritmo BRS seleccionaría el mismo repetidor, dados cualquiera de los dos perfiles.

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

La efectividad de este método puede verse, por ejemplo, en el caso en el que todas las rutas a un Repetidor desde algún Grupo A, pasan a través de algún otro Grupo B. Los dos Grupos tienen perfiles de costes equivalentes (y están por lo tanto en el mismo Conjunto de costes), ya que cualquiera que sea el mejor repetidor para el Grupo A también lo será para el grupo B, independientemente de la ruta que se tome entre los dos Grupos.

Mediante la normalización de los perfiles de costes, los perfiles de costes equivalentes pueden hacerse idénticos. Un perfil de coste normalizado es un perfil de coste en el cual el coste mínimo tiene un valor de cero. Un perfil de coste normalizado es computado encontrando el coste mínimo en el perfil, y restando ese valor de cada coste en el perfil. Los Conjuntos de coste son entonces computados utilizando el siguiente algoritmo:

- Para cada grupo G:
 - Calcular el perfil de coste normalizado para G
 - 55 • Buscar un Conjunto de coste con el mismo perfil de costes normalizado.
 - Si dicho conjunto es encontrado, agregar G al actual Conjunto de costes;
 - De lo contrario, crear un nuevo Conjunto de costes con el perfil de coste normalizado calculado, que contenga sólo G.

El algoritmo para encontrar los Conjuntos de costes emplea una tabla hash (tabla de direcciones calculadas) para reducir el tiempo necesario para determinar si el Conjunto de costes deseado ya existe. La tabla hash utiliza un valor computado del perfil de coste de G.

Cada Conjunto de costes es entonces numerado con un único número índice de conjunto de costes. Los Conjuntos de costes son entonces utilizados, de manera directa, para generar la Tabla de costes de enlaces, la cual proporciona el coste de cada Conjunto de costes a cada Repetidor.

ES 2 328 426 T3

Cómo se describe a continuación, la Tabla de reducción de grupos establece equivalencias entre cada dirección IP y uno de estos Conjunto de costes.

5 *Construcción del mapa IP*

El mapa IP es una lista ordenada de registros que establece equivalencias entre los rangos de direcciones IP y las claves de la Tabla de costes de enlace. El formato del mapa IP es:

10 <dirección IP base> <dirección IP máx> <clave de Tabla de costes de enlace>

15 donde las direcciones IP son actualmente representadas por números enteros de 32 bits. Las entradas son ordenadas por dirección base descendente, y por dirección máxima ascendente entre direcciones de igual base, y por clave de Tabla de coste de enlaces ascendente entre direcciones de igual base y direcciones máximas. Se debe tener en cuenta que los rangos podrían superponerse.

El procedimiento NetMap genera un mapa IP intermedio que contiene un mapa entre rangos de direcciones IP y números de Conjunto de costes, como se ve a continuación:

- 20 • Para cada Conjunto de costes S:
 - Para cada Grupo G en S:
 - 25 • Para cada rango de direcciones IP en G:
 - Agregar un triple (dirección inferior, dirección superior, número de Conjunto de costes de S) al mapa IP.

30 El archivo de mapa IP es entonces ordenado por dirección base descendente, y por dirección máxima ascendente entre direcciones de igual base, y por número de Conjunto de coste ascendente entre direcciones de igual base y direcciones máximas. El orden de clasificación de la dirección base y de la dirección máxima minimiza el tiempo necesario para construir la Tabla de reducción de grupos y produce los resultados correctos para las entradas que se superponen.

35 Por último, el procedimiento NetMap crea la Tabla de reducción de grupos procesando el mapa de IP ordenado. La Tabla de reducción de grupos traduce las direcciones IP (especificadas por rangos) en números de Conjunto de costes. El especial procesamiento del archivo de mapa de IP es necesario para detectar los rangos de direcciones superpuestas, y para fusionar los rangos de direcciones adyacentes para minimizar el tamaño de la Tabla de reducción de grupos.

40 Se conserva una lista ordenada de segmentos de rangos de direcciones, donde cada segmento consta de una dirección base B y de un número de Conjunto de costes N, ordenados por dirección base B. (La dirección máxima de un segmento es la dirección base del próximo segmento menos uno).

Se emplea el siguiente algoritmo:

- 45 • Inicializar la lista con los elementos [- infinito, NINGÚNGRUPO], [+ infinito, NINGÚNGRUPO].
 - Para cada entrada en el mapa IP, en orden clasificado, constando de (b, m, s),
 - 50 • Insertar (b, m, s) en la lista (se debe recordar que las entradas del mapa IP corresponden a la forma (dirección inferior, dirección superior, número de Conjunto de costes de S))
 - Para cada rango de dirección LAN reservado (b, m):
Insertar (b, m, LOCAL) en la lista.
 - 55 • Para cada Repetidor en la dirección:
Insertar (a, a, REPETIDOR) en la lista.
 - 60 • Para cada segmento S en la lista ordenada:
 - Fusionar S con los siguientes segmentos con el mismo Conjunto de costes
 - Crear una entrada de Tabla de reducción de grupos con dirección base de la dirección base de S,
 - 65 • dirección máxima = base del próximo segmento - 1,
 - identificación de grupo = número de Conjunto de costes de S.

ES 2 328 426 T3

Un rango reservado de direcciones LAN es un rango de direcciones reservado para ser utilizado por LAN que no deberían aparecer como una dirección de Internet global. LOCAL es un índice especial de Conjunto de costes diferente de todos los demás, indicando que el rango se equipara con un cliente que nunca debería ser reflejado. REPETIDOR es un índice de Conjunto de costes diferente de todos los demás, indicando que el rango de direcciones se equipara con un repetidor. NINGÚNGRUPO es un índice de Conjunto de costes diferente de todos los demás, indicando que este rango de direcciones no tiene ninguna equivalencia conocida.

Dados (B, M, N), insertar una entrada en la lista de direcciones ordenada como se detalla a continuación: Encontrar el último segmento (AB, AN) para el cual AB es menor o igual a B. Si AB es menor que B, insertar un nuevo segmento (B, N) después de (AB, AN). Encontrar el último segmento (YB, YN) para el cual YB es menor o igual a M. Reemplazar por (XB, N) cualquier segmento (XB, NINGÚNGRUPO) para el cual XB es mayor que B y menor que YB.

Si YN no es N, y YN es NINGÚNGRUPO o YB es menor o igual a B,

Permitir que (ZB, ZN) sea el segmento siguiente a (YB, YN). Si M+1 es menor que ZB, insertar un nuevo segmento (M+1, YN) antes de (ZB, ZN). Reemplazar (YB, YN) con (YB, N).

Re-escritura de recursos HTML

Como se explicó anteriormente en referencia a la Figura 3 (B5), cuando un reflector o repetidor proporciona un recurso que incluye en sí mismo identificadores de recursos (por ejemplo, un recurso HTML), ese recurso es modificado (re-escrito) para pre-reflejar identificadores de recursos (URL) de recursos repetibles que aparecen en el recurso. La re-escritura asegura que, cuando un navegador solicite recursos repetibles identificados por el recurso solicitado, los obtenga desde un repetidor sin necesidad de regresar al servidor origen, pero que cuando solicite recursos no repetibles identificados por el recurso solicitado, vaya directamente al servidor origen. Sin esta optimización, el navegador efectuaría todas las peticiones al servidor origen (incrementando el tráfico en el servidor origen y necesitando muchas más redirecciones desde el servidor origen), o efectuaría todas las peticiones al repetidor (provocando que el repetidor solicite y copie repetidamente recursos que no pudieron ser almacenados, incrementando los costes operativos de proporcionar dichos recursos).

La re-escritura requiere que se haya seleccionado un repetidor (como se describió anteriormente en referencia al Selector de mejor repetidor). La re-escritura utiliza una directiva denominada BASE. La directiva BASE permite que el HTML identifique un servidor base diferente. (Por lo general, la dirección base es la dirección del recurso HTML).

La re-escritura se lleva a cabo de la siguiente manera:

- F1. Una directiva BASE es agregada al inicio del recurso HTML, o es modificada donde sea necesario. Normalmente, un navegador interpreta que los URL relativos, son relativos a la dirección base predeterminada, a saber, la URL del recurso HTML (página) en la cual se encuentran. La dirección BASE especifica el recurso en el reflector que originalmente proporcionó el recurso. Esto significa que los URL relativos no procesados (como aquellos generados por programas JavascriptTM), serán interpretados como relativos al reflector. Sin esta dirección BASE, los navegadores combinarían direcciones relativas con nombres de repetidores para crear URL que no tuvieran la forma requerida por los repetidores (como se describió con anterioridad en el paso D1).
- F2. El re-grabador identifica las directivas, como las imágenes embebidas y adjuntos, que contienen URL. Si el re-grabador se está ejecutando en un reflector, debe analizar el archivo HTML para identificar estas directivas. Si se está ejecutando en un repetidor, el re-grabador debe tener acceso a la información precomputada que identifica la ubicación de cada URL (situada en el archivo HTML en el paso F4).
- F3. Para que cada URL encontrado en el recurso sea re-escrito, el re-grabador debe determinar si la URL es repetible (como en los pasos B1-B2). Si la URL no es repetible, no es modificado. Por otra parte, si la URL es repetible, es modificada para referirse al repetidor seleccionado.
- F4. Después de que todos los URL hayan sido identificados y modificados, si el recurso se está sirviendo a un repetidor, se adjunta una tabla al inicio del recurso que identifica la ubicación y contenido de cada 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 han identificado todos los cambios, se calcula una nueva longitud para el recurso (página). La longitud es insertada en el encabezado HTTP antes de servir el recurso.

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

ES 2 328 426 T3

Gestión de protocolos no HTTP

Esta invención hace posible reflejar las referencias a los recursos que son servidos por protocolos distintos de HTTP, por ejemplo, el protocolo de transferencia de archivos (FTP) y los protocolos de flujo de audio/vídeo. Sin embargo, muchos protocolos no proporcionan la capacidad de redireccionar peticiones. Es, sin embargo, posible redireccionar referencias antes de que se efectúen las peticiones reescribiendo los URL embebidos en páginas HTML. Las siguientes modificaciones a los algoritmos mencionados con anterioridad se utilizan para respaldar esta capacidad.

En F4, el re-grabador reescribe URL para servidores si aquellos servidores aparecen en una tabla configurable de servidores de origen cooperativos, también llamados coservidores. El operador de reflector puede definir esta tabla para que incluya servidores FTP y otros servidores. Un URL re-escrito que hace referencia a un recurso no HTTP presenta la forma:

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

donde <esquema> es un nombre de protocolo admitido como “ftp”. Este formato de URL es una forma alternativa a la mostrada en B3.

En C3, el repetidor busca un protocolo embebido en la petición entrante. Si el protocolo está presente y el recurso solicitado aún no se ha almacenado, el repetidor utiliza el protocolo seleccionado en lugar del protocolo HTTP predeterminado para solicitar el recurso al servirlo y almacenarlo en el caché.

Configuración y administración del sistema

Además del procesamiento descrito anteriormente, la red de repetidores requiere de diversos mecanismos para la configuración del sistema y la administración de la red. Algunos de estos mecanismos se describen aquí.

Los reflectores permiten que sus operadores sincronicen los cachés repetidores llevando a cabo operaciones de publicación. El proceso de mantener sincronizados los cachés repetidores se describe a continuación. La publicación indica que un recurso o colección de recursos ha cambiado.

Los repetidores y reflectores participan en distintos tipos de procesamiento de registro. Los resultados de los algoritmos recopilados en los repetidores son recopilados y fusionados con algoritmos recopilados en los reflectores, como se describe a continuación.

Agregar suscriptores a la red de repetidores

Cuando se agrega un nuevo suscriptor a la red, la información acerca del suscriptor es ingresada en la Tabla de suscriptor en el repetidor principal y se propaga a todos los repetidores en la red. Esta información incluye la Tasa de información total comprometida o CAIR (por sus siglas en Inglés) para los servidores que pertenecen al suscriptor, y una lista de repetidores que podrían ser utilizados por los servidores que pertenecen al suscriptor.

Agregar reflectores a la red de repetidores

Cuando se agrega un nuevo reflector a la red, éste simplemente se conecta a y se anuncia ante un repetidor de contacto, preferentemente utilizando un certificado encriptado de manera segura, que incluye el identificador del suscriptor del repetidor.

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

El reflector procesa peticiones durante este tiempo, pero no está “habilitado” (no está autorizado a reflejar peticiones) hasta que todas sus tablas no estén al día.

Mantener sincronizados los cachés de repetidores

Los cachés de repetidores son coherentes, en el sentido de que cuando un reflector identifica un cambio a un recurso, todos los cachés de repetidores son notificados, y aceptan el cambio en una sola transacción.

Sólo el identificador del recurso cambiado (y no todo el recurso) es transmitido a los repetidores; el identificador es utilizado para invalidar de manera efectiva el recurso almacenado correspondiente en el repetidor. Este proceso es mucho más eficiente que emitir el contenido del recurso cambiado a cada repetidor.

ES 2 328 426 T3

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

Un cambio de recurso es identificado en el reflector, ya sea de forma manual por el operador, o a través de un script (archivo de órdenes o archivo de procesamiento por lotes) cuando los archivos están instalados en el servidor, o automáticamente mediante un mecanismo de detección de cambios (por ejemplo, un proceso individual que verifica los cambios de forma regular).

Un cambio de recursos provoca que el reflector envíe un mensaje de “invalidación” a su repetidor de contacto, el cual reenvía el mensaje al repetidor principal. El mensaje de invalidación contiene una lista de identificadores de recursos (o expresiones regulares que identifican patrones de identificadores de recursos) que han cambiado. (Las expresiones regulares son utilizadas para invalidar un directorio o todo un servidor). La red de repetidores utiliza un proceso de concomitancia de dos fases, para garantizar que todos los repetidores invaliden de manera correcta un recurso dado.

El proceso de invalidación funciona de la siguiente manera:

El repetidor principal emite una petición de invalidación de “fase 1” a todos los repetidores, indicando los recursos y expresiones regulares que describen conjuntos de recursos a ser invalidados.

Cuando cada repetidor recibe el mensaje de fase 1, colocará primero los identificadores de recursos o expresiones regulares en una lista de identificadores de recursos pendientes de invalidación.

Todo recurso solicitado (en C3) que se encuentre en la lista pendiente de invalidación no podrá ser servido desde el caché. Esto evita que el caché solicite el recurso desde una parte de caché que podría no haber recibido un aviso de invalidación. En el caso de solicitar un recurso de esta manera, podría reemplazar el recurso recientemente invalidado por los mismos datos, ahora caducados.

El repetidor entonces compara el identificador de recurso de cada recurso en su caché con los identificadores de recursos y expresiones regulares en la lista.

Cada coincidencia es invalidada marcándola como caduca y opcionalmente eliminándola del caché. Esto significa que una petición futura del recurso hará que recupere una nueva copia de este recurso del reflector.

Cuando el repetidor ha completado la invalidación, devuelve un reconocimiento al repetidor principal. El repetidor principal espera hasta que todos los repetidores hayan dado acuse de recibo a la petición de invalidación.

Si un repetidor no falla en hacer el acuse de recibo dentro de un período dado, es desconectado del repetidor principal. Cuando se vuelva a conectar, se le solicitará que vacíe todo su caché, lo cual eliminará cualquier problema de consistencia. (Para evitar vaciar todo el caché, el repetidor principal podría conservar un registro de todas las invalidaciones realizadas, ordenadas por fecha, y vaciar sólo los archivos invalidados desde la última vez que el repetidor de reconexión completó de forma exitosa una invalidación. En las actuales realizaciones preferentes, esto no se lleva a cabo dado que se cree que los repetidores raramente se desconectarán).

Cuando todos los repetidores han reconocido la invalidación (o han expirado), el repetidor emite una petición de invalidación de “fase 2” a todos los repetidores. Esto hace que los repetidores eliminen los identificadores de recursos y las expresiones regulares correspondientes de la lista de identificadores de recursos pendientes de invalidación.

En otra realización, la petición de invalidación se extenderá para permitir un “servidor push” (de transmisión de datos sin solicitud). En peticiones de ese tipo, después de que se haya completado la fase 2 del proceso de invalidación, el repetidor que recibe la petición de invalidación solicitará de manera inmediata una nueva copia del recurso invalidado para colocarlo en su caché.

Registros y procesamiento de registros

Los registros de actividad del servidor web son fundamentales para monitorear la actividad en un sitio web. La presente invención crea “registros fusionados” que combinan la actividad en los reflectores con la actividad en los repetidores, para que aparezca sólo un registro de actividades en el servidor origen que muestre todas las peticiones de recursos web realizadas en nombre de ese sitio en cualquier repetidor.

Este registro fusionado puede procesarse con herramientas estándar de procesamiento, como si hubiera sido generado de manera local.

De manera periódica, el repetidor principal (o su delegado) recopila registros de cada repetidor. Los registros recopilados se fusionan, clasificados por identificador de reflector y fecha de registro, y son almacenados en un archivo con marcación de fecha por cada reflector. El registro fusionado de un reflector determinado representa la actividad de todos los repetidores en nombre de ese reflector. De manera periódica, tal y como lo configuró el operador del reflector, un reflector contacta con el repetidor principal para solicitar sus registros fusionados. Éste los descarga y los

ES 2 328 426 T3

fusiona con sus registros mantenidos de forma local, clasificándolos por fecha de registro. El resultado es un registro fusionado que representa toda la actividad en nombre de los repetidores y del reflector dado.

5 Los registros de actividad son extendidos de manera opcional con información importante a la red de repetidores, si el reflector está configurado para hacerlo por el operador del reflector. En particular, un “código de estado extendido” indica información acerca de cada petición, como por ejemplo:

1. la petición fue servida por un reflector de manera local;
- 10 2. la petición fue reflejada al repetidor;*
3. la petición fue servida por un reflector a un repetidor;*
- 15 4. la petición de recurso no repetible fue servida por el repetidor;*
5. la petición fue servida por un repetidor desde el caché;
6. la petición fue servida por un repetidor después de completar el caché;
- 20 7. la petición pendiente de invalidación fue servida por un repetidor.

(Las actividades marcadas con “*” representan estados intermedios de una petición y no suelen aparecer en el registro final de actividad).

25 Además, los registros de actividad contienen una duración, y fechas de registro de precisión extendida. La duración hace posible el análisis del tiempo requerido para servir un recurso, el ancho de banda utilizado, la cantidad de peticiones gestionadas en paralelo en un momento dado, y otras informaciones útiles. La marca de tiempo de precisión extendida hacen posible fusionar de forma precisa los registros de actividad.

30 Los repetidores utilizan el protocolo del tiempo de red (NTP, por sus siglas en inglés) para mantener sincronizados los relojes. Los reflectores pueden emplear el NTP o calcular un pulso temporal para proporcionar marcas de tiempos más o menos exactas en relación con su repetidor de contacto.

35 *Ejecución de la tasa de información total comprometida*

La red de repetidores monitorea y limita la tasa total a la cual los datos son servidos en nombre de un determinado suscriptor por todos los repetidores. Este mecanismo proporciona los siguientes beneficios:

- 40 1. proporciona un medio de cálculo de precios del servicio del repetidor;
2. proporciona un medio para estimar y reservar capacidad en los repetidores;
- 45 3. proporciona un medio para evitar que los clientes de un sitio ocupado limiten el acceso a otros sitios.

Para cada suscriptor, se configura y mantiene un “ratio del umbral de información agregada” (TAIR) en el repetidor principal. Este umbral no es necesariamente la tasa comprometida, sino que puede ser un múltiplo de la tasa comprometida, basándose en la política de cálculo de precios.

50 Cada repetidor mide el componente de tasa de información de cada reflector para el cual sirve recursos, periódicamente (por lo general, una vez por minuto), registrando la cantidad de bytes transmitidos en nombre de ese reflector cada vez que se entrega una petición. La tabla así creada es enviada al repetidor principal una vez por período. El repetidor principal combina las tablas de cada repetidor, sumando la información medida de cada reflector sobre todos los repetidores que proporcionan recursos para ese reflector, para determinar la “tasa de información total medida” (MAIR) para cada reflector.

Si la MAIR de un determinado reflector es mayor que la TAIR de ese reflector, la MAIR es transmitida por el repetidor principal a todos los repetidores y al respectivo reflector.

60 Cuando un reflector recibe una petición, determina si su MAIR calculada más recientemente es mayor que su TAIR. Si es así, el reflector decide de manera probabilística si debe suprimir la reflexión, sirviendo la petición de manera local (en B2). La probabilidad de suprimir la reflexión se incrementa como una función exponencial de la diferencia entre la MAIR y la CAIR.

65 Servir una petición de manera local durante un periodo de tráfico máximo puede forzar al servidor origen local, pero evita que este suscriptor ocupe más ancho de banda que el asignado de la red compartida de repetidores.

ES 2 328 426 T3

Cuando un repetidor recibe una petición de un determinado suscriptor (en C2), determina si el suscriptor se está ejecutando cerca de su ratio mínimo de información agregada. Si es éste el caso, decide de manera probabilística si debe reducir su carga redireccionando la petición de vuelta al reflector. La probabilidad se incrementa exponencialmente a medida que la tasa de información total del reflector se aproxima a su límite.

5

Si una petición es reflejada de vuelta a un reflector, se anexa una cadena especial de caracteres al identificador de recursos para que el reflector receptor no intente reflejarla otra vez. En el sistema actual, esta cadena tiene la forma "src=overload".

10 El reflector prueba esta cadena en B2.

El mecanismo para limitar el ratio de agregado de información descrito anteriormente es bastante tosco. Limita al nivel de las sesiones con clientes (ya que una vez que un cliente ha sido reflejado a un determinado repetidor, el proceso de re-escritura tiende a hacer que el cliente regrese a ese repetidor) y, en el mejor de los casos, limita peticiones individuales de recursos. Un mecanismo más minucioso para forzar los límites TAIR dentro de los repetidores, opera reduciendo el consumo de ancho de banda de un suscriptor ocupado cuando otros suscriptores están compitiendo por el ancho de banda.

15

El mecanismo minucioso es una forma de "ratio de forma" de datos. Extiende el mecanismo que copia datos de recursos a una conexión cuando se está enviando una respuesta a un cliente. Cuando se establece un canal de salida en el momento en que una petición es recibida, el repetidor identifica para qué suscriptor está operando el canal, en C2, y registra al suscriptor en un campo de datos asociado con el canal. Cada vez que una operación de "escritura" está a punto de realizarse al canal, el flujo medido de salida inspecciona, en primer lugar, los valores actuales de la MAIR y de la TAIR, calculados anteriormente, para el suscriptor dado. Si la MAIR es mayor que la TAIR, entonces el mecanismo se pausa brevemente antes de efectuar la operación de escritura. La duración de la pausa es proporcional a la cantidad en la que la MAIR excede la TAIR. La pausa asegura que las tareas que envían otros recursos a otros clientes, tal vez en nombre de otros suscriptores, tendrán la oportunidad de enviar sus datos.

20

Resiliencia de la red de repetidores

30

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

Un repetidor no puede operar a menos que esté conectado al repetidor principal. El repetidor principal intercambia información crítica con otros repetidores, incluyendo información acerca de la carga del repetidor, la tasa de información total, los suscriptores, y el coste de enlace.

35

Si un repetidor falla, un proceso de "sucesión" asegura que otro repetidor cumplirá la función del repetidor principal, por lo que la red en su conjunto continuará estando operativa. Si un repetidor principal falla, o si una conexión a un repetidor principal falla por un problema de red, cualquier repetidor que intente comunicarse con el repetidor principal detectará el fallo, ya sea a través de una indicación de TCP/IP, o por un time-out (tiempo límite del sistema) que se genere mediante un mensaje regular tipo "pulso" que envía al repetidor principal.

40

Cuando cualquier repetidor es desconectado de su repetidor principal, intenta de forma inmediata reconectarse a una serie de repetidores principales potenciales basándose en un archivo configurable denominado como su "lista de sucesión".

45

El repetidor prueba cada sistema en la lista en sucesión hasta que logra conectarse correctamente a un repetidor principal. Si se encuentra en este proceso, acude a su propio nombre, asume el papel de repetidor principal, y acepta conexiones desde otros repetidores. Si un repetidor que no está al principio de la lista se convierte en el repetidor principal, es denominado "repetidor principal provisional".

50

Una partición de red puede causar que dos grupos de repetidores elijan cada uno a un repetidor principal. Cuando la partición es corregida, es necesario que el repetidor maestro de más antigüedad se haga cargo del control de la red. Por lo tanto, cuando un repetidor es un repetidor principal temporal, intenta de forma regular reconectarse a cualquier repetidor principal que se encuentre por encima de él en la lista de sucesión. Si lo logra, se desconecta inmediatamente de todos los otros repetidores a él conectados. Cuando vuelvan a probar sus listas de sucesión, se conectarán al repetidor principal de más antigüedad.

55

Para evitar la pérdida de datos, un repetidor principal provisional no acepta cambios de configuración y no procesa archivos de registro. Para poder asumir estas tareas, se le debe informar que es un repetidor principal mediante la modificación manual de su lista de sucesores. Cada repetidor vuelve a cargar su lista de sucesores de manera regular para determinar si debería cambiar su idea de quién es el repetidor principal.

60

Si un repetidor es desconectado del repetidor principal, debe volver a sincronizar su caché cuando se reconecte al repetidor principal. El repetidor principal puede mantener una lista de invalidaciones recientes de caché y enviar al repetidor cualquier invalidación que no haya sido capaz de procesar mientras estuvo desconectado. Si por alguna razón esta lista no se encuentra disponible (por ejemplo, porque el reflector ha estado desconectado demasiado tiempo), el reflector debe invalidar todo su caché.

65

ES 2 328 426 T3

A un reflector no se le permite reflejar peticiones a menos que esté conectado a un repetidor. El reflector depende de su repetidor de contacto para obtener información crítica, como tablas de coste de enlace y carga, y la tasa actual de información total. Un reflector no conectado a un repetidor puede continuar recibiendo peticiones y gestionándolas de forma local:

5

Si un reflector pierde su conexión con un repetidor, debido a un fallo del repetidor o a una interrupción de la red, continúa operando mientras intenta conectarse a un repetidor.

10

Cada vez que un reflector intenta conectarse a un repetidor, utiliza DNS para identificar un conjunto de repetidores posibles, dado un nombre de dominio que representa a la red de repetidores. El reflector prueba cada repetidor en este conjunto hasta que establece contacto con éxito. Hasta que logra efectuar un contacto exitoso, el reflector gestiona todas las peticiones de manera local. Cuando un reflector se conecta a un repetidor, el repetidor puede indicarle que intente conectarse a un repetidor diferente; esto le permite a la red de repetidores asegurar que ningún repetidor individual tenga demasiados contactos.

15

Cuando se establece un contacto, el reflector proporciona el número de versión de cada una de sus tablas a su repetidor de contacto. El repetidor decide entonces qué tablas deberían ser actualizadas y envía las actualizaciones apropiadas al reflector. Cuando todas las tablas han sido actualizadas, el repetidor informa al reflector que ya puede comenzar a reflejar peticiones.

20

Utilización de un caché de proxy dentro de un repetidor

25

Los repetidores están diseñados intencionadamente para que cualquier caché de proxy pueda ser utilizado como un componente dentro de ellos. Esto es posible porque el repetidor recibe peticiones HTTP y las convierte a una forma reconocida por parte del caché de proxy.

30

Por otra parte, se han realizado o es posible realizar diversas modificaciones a un caché de proxy estándar a modo de optimizaciones. Esto incluye, en particular, la capacidad de invalidar de forma conveniente un recurso, la capacidad de admitir cuotas de caché, y la capacidad de evitar producir una copia adicional de cada recurso mientras pasa desde el caché de proxy al repetidor, y de allí al solicitante.

35

En una realización preferente, se utiliza un caché de proxy para implementar C3. El caché de proxy está dedicado para su uso sólo por parte de uno o más repetidores. Cada repetidor que solicita un recurso del caché de proxy construye una petición de proxy a partir de la petición de recurso entrante. Una petición común HTTP GET a un servidor contiene sólo la parte del nombre de ruta dla URL - el esquema y el nombre de servidor están implícitos. (En una petición HTTP GET a un repetidor, la parte del nombre de ruta dla URL incluye el nombre del servidor origen en nombre del cual se efectúa la petición, como se describió anteriormente). Sin embargo, una petición GET de agente proxy necesita una URL completa. Por lo tanto, el repetidor debe construir una petición proxy que contenga todo la URL de la porción de ruta dla URL que reciba. Específicamente, si la petición entrante tiene la forma:

40

```
GET / <servidor origen>l <ruta>
```

entonces el repetidor construye una petición proxy de la forma:

45

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

y si la petición entrante tiene la forma:

50

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

entonces el repetidor construye una petición proxy de la forma:

55

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

Control de caché

60

Las respuestas HTTP contienen directivas llamadas directivas de control de caché, las cuales se utilizan para indicarle a un caché si el recurso adjunto puede ser almacenado, y de ser así, cuándo debería expirar. Un administrador de sitio web configura el sitio web para adjuntar directivas apropiadas. A menudo, el administrador no sabrá por cuánto tiempo una página permanecerá actualizada, y debe definir un tiempo corto de expiración para intentar evitar que los cachés proporcionen datos obsoletos. En muchos casos, un operador de sitio web indicará un tiempo corto de expiración sólo con el fin de recibir las peticiones (o consultas) que de otra manera deberían ser enmascaradas por la presencia de un caché. Este proceso se conoce en la industria como “caché-busting”. Aunque algunos operadores de caché podrían considerar que el caché busting es inadecuado, los anunciantes que dependen de esta información podrían considerarlo de gran importancia.

65

ES 2 328 426 T3

Cuando un recurso es almacenado en un repetidor, sus directivas de caché pueden ser ignoradas por el repetidor, porque el repetidor recibe eventos explícitos de invalidación para determinar cuándo un recurso es caduco. Cuando un caché proxy es utilizado como el caché en el repetidor, las directivas de caché asociadas podrían estar deshabilitadas temporalmente. Sin embargo, deben ser rehabilitadas cuando el recurso es servido desde el caché al cliente, con el fin de permitir que la política de control de caché (incluyendo cualquier caché busting) haga efecto.

La presente invención contiene mecanismos para evitar que el caché de proxy dentro de un repetidor cumpla las directivas de control de caché, mientras que permite que las directivas sean servidas desde el repetidor.

Cuando un reflector sirve un recurso a un repetidor en B4, reemplaza todas las directivas de caché por directivas modificadas que son ignoradas por el caché de proxy del repetidor. Lo hace prefijando una cadena característica, tal como “wr-”, al principio de la etiqueta HTTP. De este modo, “expira” se convierte en “wr-expira”, y “control-caché” se convierte en “wr-control-caché”. Esto evita que el propio caché de proxy cumpla las directivas. Cuando un repetidor sirve un recurso en C4, y el cliente solicitante no es otro repetidor, busca etiquetas HTTP que comiencen con “wr-” y elimina el “wr-”. Esto permite que los clientes recuperen el recurso para cumplir las directivas.

Revalidación de recursos

Existen muchos casos en los que un recurso puede ser almacenado siempre que se consulte al servidor origen cada vez que es proporcionado/servido. En un caso, la petición del recurso está anexada a lo que se denomina “cookie”. Se le debe entregar la cookie al servidor origen para registrar la petición y determinar si es posible proporcionar/servir o no el recurso almacenado. En otro caso, la petición para el recurso está anexada a un encabezado de autenticación (el cual identifica al solicitante con una identificación de usuario y una contraseña). Cada nueva petición del recurso debe ser probada en el servidor origen para asegurar que el solicitante esté autorizado a acceder al recurso.

La especificación HTTP 1.1 define un encabezado de respuesta titulado “Must-Revalidate” (“Debe-Revalidarse”) que permite que un servidor origen instruya a un caché de proxy “revalidar” un recurso cada vez que se recibe una petición. Por lo general, este mecanismo es utilizado para determinar si un recurso aún está actualizado. En la presente invención, el encabezado Must-Revalidate permite pedir a un servidor origen que valide una petición que de otra manera sería proporcionada desde un repetidor.

La base de reglas del reflector contiene información que determina qué recursos pueden ser repetidos pero deben revalidarse cada vez que son servidos. Para cada recurso de ese tipo, en B4, el reflector adjunta un encabezado Must-Revalidate. Cada vez que una petición llega a un repetidor para obtener un recurso almacenado marcado con un encabezado Must-Revalidate, la petición es reenviada al reflector para su validación antes de servir el recurso solicitado.

Cuotas de caché

El componente de caché de un repetidor se comparte entre aquellos suscriptores que reflejan clientes a ese repetidor. Para permitir que los suscriptores tengan un acceso justo a los recursos almacenados, el caché debe extenderse para soportar cuotas.

Por lo general, un caché de proxy puede configurarse con un umbral de espacio en disco T. Siempre que se almacenen más de T bytes en el caché, el caché intenta buscar recursos para eliminar.

Típicamente, un caché utiliza el algoritmo menos utilizado recientemente (LRU, por sus siglas en inglés) para determinar qué recursos eliminar; los cachés más sofisticados utilizan otros algoritmos. Un caché también puede admitir distintos valores de umbral, por ejemplo, un umbral menor que, cuando es alcanzado, haga que un proceso de baja prioridad en segundo plano elimine artículos del caché, y un umbral mayor que, cuando es alcanzado, evite que los recursos sean almacenados hasta que se reclame suficiente espacio libre en disco.

Si dos suscriptores A y B comparten un caché, y se accede a más recursos del suscriptor A durante un período de tiempo que a recursos del suscriptor B, entonces menos recursos de B estarán en el caché cuando lleguen nuevas peticiones. Es posible que, debido al comportamiento de A, los recursos de B nunca se almacenen al ser solicitados. En la presente invención, este comportamiento no es deseable. Para abordar este problema, la invención extiende el caché en un repetidor para soportar cuotas de caché.

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

Siempre que se agrega un recurso al caché (en C3), el valor D_s es actualizado para el suscriptor que proporciona el recurso. Si D_s es mayor que T_s , el caché intenta encontrar recursos para eliminar, de entre aquellos recursos asociados con el suscriptor S. El caché es dividido de manera efectiva en áreas separadas para cada suscriptor.

El umbral original T aún es soportado. Si la suma de los segmentos reservados para cada suscriptor es menor que el espacio total reservado en el caché, el área restante es “común” y está sujeta a la competición entre los suscriptores.

Se debe tener en cuenta que este mecanismo podría ser implementado modificando el caché de proxy existente analizado anteriormente, o también podría ser implementado sin modificar el caché de proxy, si el caché de proxy permite, al menos, que un programa externo obtenga una lista de recursos en el caché, y elimine un recurso determinado del caché.

5

Re-escritura desde repetidores

10 Cuando un repetidor recibe una petición de un recurso, su caché de proxy puede ser configurado para determinar si una parte de caché contiene el recurso solicitado. De ser así, el caché de proxy obtiene el recurso del caché par, lo que puede resultar más rápido que obtenerlo desde el servidor origen (el reflector). Sin embargo, una consecuencia de esto es que los recursos HILL re-escritos recuperados desde la parte de caché identificarían al repetidor equivocado. Por lo tanto, para permitir cachés de proxy cooperadores, los recursos son preferentemente re-escritos en el repetidor.

15 Cuando un recurso es re-escrito para un repetidor, se coloca una etiqueta especial al comienzo del recurso. Al construir una respuesta, el repetidor inspecciona la etiqueta para determinar si el recurso indica que es necesario efectuar una re-escritura adicional. De ser así, el repetidor modifica el recurso reemplazando las referencias al antiguo repetidor con referencias al nuevo repetidor.

20 Sólo es necesario llevar a cabo esta re-escritura cuando se proporciona un recurso al caché de proxy en otro repetidor.

Inclusión por parte del repetidor

25

En ocasiones, un servidor origen construye un recurso personalizado para cada petición (por ejemplo, al insertar un aviso basado en el historial del cliente solicitante). En dicho caso, ese recurso debe ser servido de manera local en lugar de ser repetido. Por lo general, un recurso personalizado contiene, junto con la información personalizada, texto y referencias a otros recursos repetibles.

30

El proceso que integra una “página” desde un recurso de texto y posiblemente uno o más 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 integre la página utilizando texto o directivas desde otro recurso. Por lo tanto, a menudo los recursos personalizados contienen necesariamente grandes cantidades de texto estático que de otro modo sería repetible.

35

Para resolver esta ineficiencia en potencia, los repetidores reconocen una directiva especial denominada “inclusión por parte del repetidor”. Esta directiva permite que el repetidor integre un recurso personalizado, utilizando una combinación de recursos repetibles y locales. De esta manera, el texto estático puede ser repetible, y sólo será necesario que la directiva especial sea servida de forma local por el reflector.

40

45 Por ejemplo, un recurso X podría constar de directivas personalizadas que seleccionen un cartel de publicidad, seguido por un extenso artículo de texto. Para hacer que este recurso sea repetible, el administrador del sitio web debe romper un segundo recurso, Y, para seleccionar el cartel. El recurso X es modificado para contener una directiva “inclusión por parte del repetidor” que identifique al recurso Y, junto con el artículo. Se crea el recurso Y, y contiene sólo las directivas personalizadas que seleccionan un cartel de publicidad. Ahora el recurso X es repetible, y sólo el recurso Y, que es relativamente pequeño, no es repetible.

50 Cuando un repetidor construye una respuesta, determina si el recurso que está siendo servido es un recurso HTML, y de ser así, lo examina en busca de directivas “repeater-side include”. Cada directiva de ese tipo incluye un URL, el cual el repetidor convierte y reemplaza en el lugar de la directiva. Todo el recurso debe ser integrado antes de poder ser servido, con el fin de determinar su tamaño final, ya que el tamaño está incluido en un encabezado de respuesta delante del recurso.

55 Por lo tanto, se proporciona un método y un aparato para espejar de manera dinámica los recursos seleccionados en redes informática. Una persona experta en el arte apreciará que la presente invención puede ser ejercida mediante otras realizaciones distintas de las aquí descritas, las cuales son presentadas con fines ilustrativos y no limitativos, y que la presente invención está limitada sólo por las reivindicaciones que figuran a continuación.

60

65

REIVINDICACIONES

1. Un método, en una red informática que consta de una red de servidores repetidores (104) para espejar al menos ciertos recursos almacenados en al menos un servidor origen (102), dichos servidores repetidores estando adaptados para servir contenido a al menos un cliente (106),

caracterizado porque el método consta de los siguientes pasos:

(A) recibir, en un módulo de software, una petición de un recurso de un servidor origen efectuada por un cliente, seleccionando el módulo de software un servidor repetidor de la red de repetidores (104) para servir el recurso al cliente, utilizando la selección un coste que incluya la carga o un coste de transmisión previsto o una velocidad de transmisión prevista, y causar que el cliente (106) solicite el recurso del servidor repetidor seleccionado (104), utilizando un identificador de recursos recibido desde el módulo de software, designando el identificador de recursos al servidor repetidor seleccionado.

(B) en respuesta al servidor repetidor seleccionado (104), al que se solicita que sirva el recurso al cliente (106), determinar, en el servidor repetidor seleccionado (104), si el recurso solicitado se encuentra disponible en el servidor repetidor seleccionado (104);

(C) cuando se determina que el recurso solicitado no está disponible en el servidor repetidor seleccionado (104), obtener el recurso solicitado desde el servidor origen (102) o desde otro servidor repetidor (104); y

(D) proporcionar el recurso al cliente (106) desde el servidor repetidor seleccionado (104).

2. El método de la reivindicación 1, donde dicho paso (A) de causar es llevado a cabo por medios conectados de manera operacional con el al menos un servidor origen (102).

3. El método de la reivindicación 1 ó 2, donde un cliente (106) puede solicitar un recurso utilizando al menos un primer identificador de recursos, y donde dicho paso (A) devuelve un segundo identificador de recursos distinto de dicho primer identificador de recursos, y estando asociado con el servidor repetidor seleccionado en la red de servidores repetidores.

4. El método de la reivindicación 3, donde el segundo identificador de recursos es una modificación del primer identificador de recursos.

5. El método de la reivindicación 3 ó 4, donde el primer identificador de recursos comprende un nombre de equipo.

6. El método de cualquiera de las reivindicaciones 3 a 5, donde el segundo identificador de recursos comprende una dirección de Protocolo de Internet (IP) o un nombre de equipo asociado con un servidor repetidor en la red de servidores repetidores.

7. El método de cualquiera de las reivindicaciones 3 a 6, donde el primer identificador de recursos está asociado con una ubicación del recurso.

8. El método de cualquiera de las reivindicaciones 1 a 7, donde dicho paso (A) de causar proporciona al cliente un identificador de recursos modificado que designa un repetidor seleccionado en la red de servidores repetidores.

9. El método de cualquiera de las reivindicaciones 1 a 8, donde el paso de selección realiza una selección basada, al menos en parte, en una dirección de red asociada con el cliente.

10. El método de cualquiera de las reivindicaciones 1 a 9, donde el paso de selección realiza una selección basada, al menos en parte, en una carga de al menos algunos de los servidores repetidores.

11. El método de cualquiera de las reivindicaciones 1 a 10, donde el paso de selección realiza una selección basada, al menos en parte, en un coste o velocidad de transmisión entre al menos algunos de los servidores repetidores y el cliente.

12. El método de cualquiera de las reivindicaciones 3 a 11, donde la red es parte de Internet y donde el primer identificador de recursos es un localizador uniforme de recursos (URL, por sus siglas en inglés) para designar recursos en Internet.

13. El método de cualquiera de las reivindicaciones 3 a 12, donde los identificadores son localizadores uniformes de recursos (URLs).

14. El método de cualquiera de las reivindicaciones 3 a 13, donde los identificadores de recursos comprenden nombres que deben ser convertidos por un Servicio de Nombres de Dominio (DNS, por sus siglas en inglés) o direcciones IP.

ES 2 328 426 T3

15. El método de la reivindicación 14, donde los nombres son nombres de equipos o nombres de dominio.

16. El método de cualquiera de las reivindicaciones 3 a 15, donde el primer identificador de recursos está incluido en otro recurso anteriormente solicitado por y servido al cliente.

5

17. El método de cualquiera de las reivindicaciones 3 a 16, donde la ubicación especificada por el primer identificador de recursos es un servidor origen.

10

18. El método de cualquiera de las reivindicaciones 3 a 17, donde la ubicación especificada por el primer identificador de recursos es distinta de la red de servidores repetidores.

19. El método de cualquiera de las reivindicaciones 1 a 8, donde el recurso solicitado es una imagen embebida.

15

20. El método de cualquiera de las reivindicaciones 1 a 19, donde el recurso solicitado en sí mismo contiene identificadores de recursos, y donde el método consta adicionalmente de:

reescribir el recurso solicitado para reemplazar, al menos, algunos de los identificadores de recursos contenidos en el mismo por identificadores de recursos modificados que designan el servidor repetidor seleccionado en la red de servidores repetidores.

20

21. El método de cualquiera de las reivindicaciones 1 a 20, donde los recursos son seleccionados del grupo que consta de: contenido de vídeo estático y dinámico, contenido de audio, texto, contenido de imagen, páginas web, archivos HTML, archivos XML, archivos en lenguaje descriptor, documentos, documentos de hipertexto, archivos de datos y recursos embebidos.

25

22. Red informática que consta de una red de servidores repetidores (104) para espejar al menos ciertos recursos almacenados en al menos un servidor origen (102), dichos servidores encontrándose adaptados para servir contenido a al menos un cliente (106), la red informática constando de medios para ejecutar el método de cualquiera de las reivindicaciones 1-21.

30

35

40

45

50

55

60

65

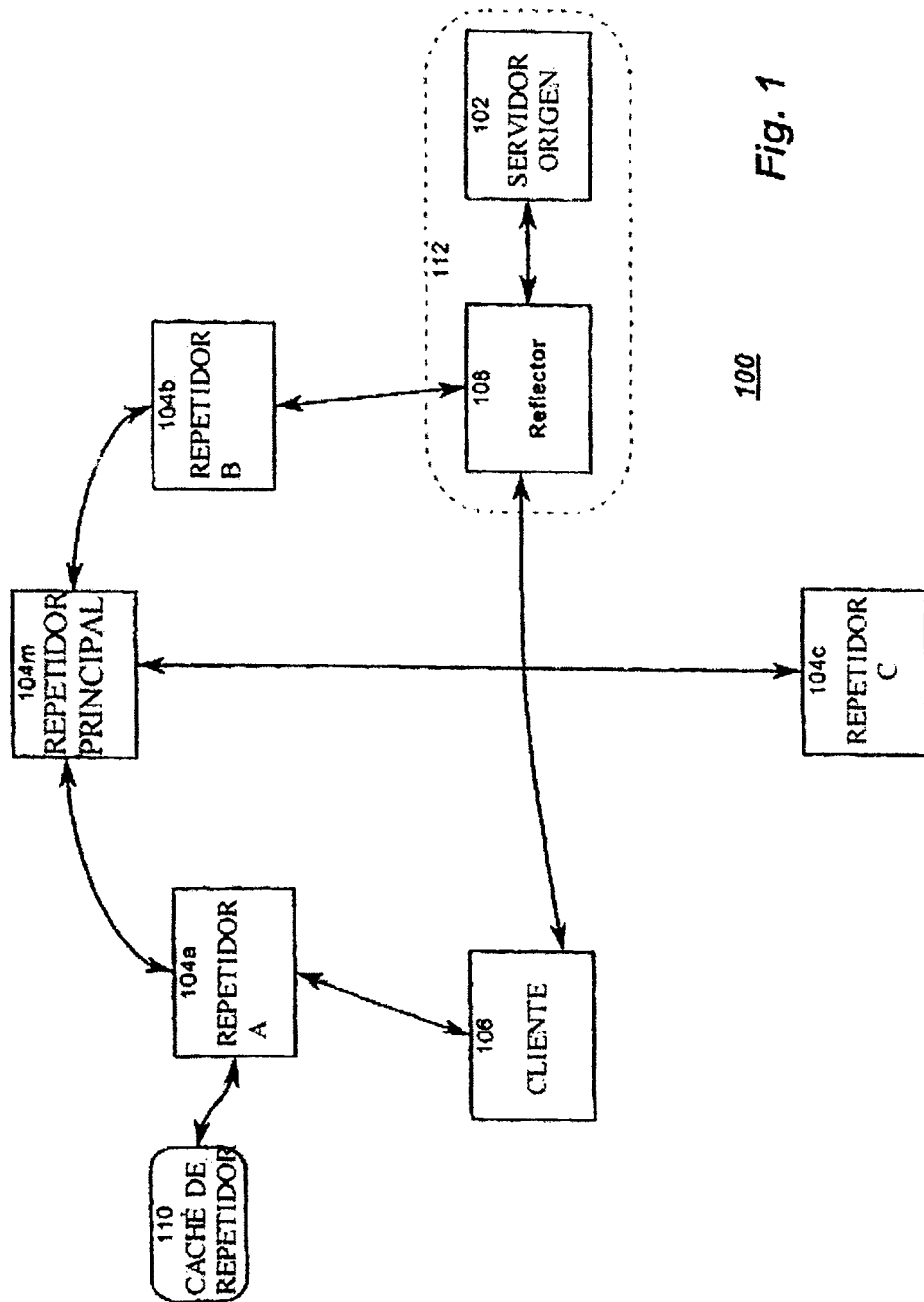


Fig. 1

100

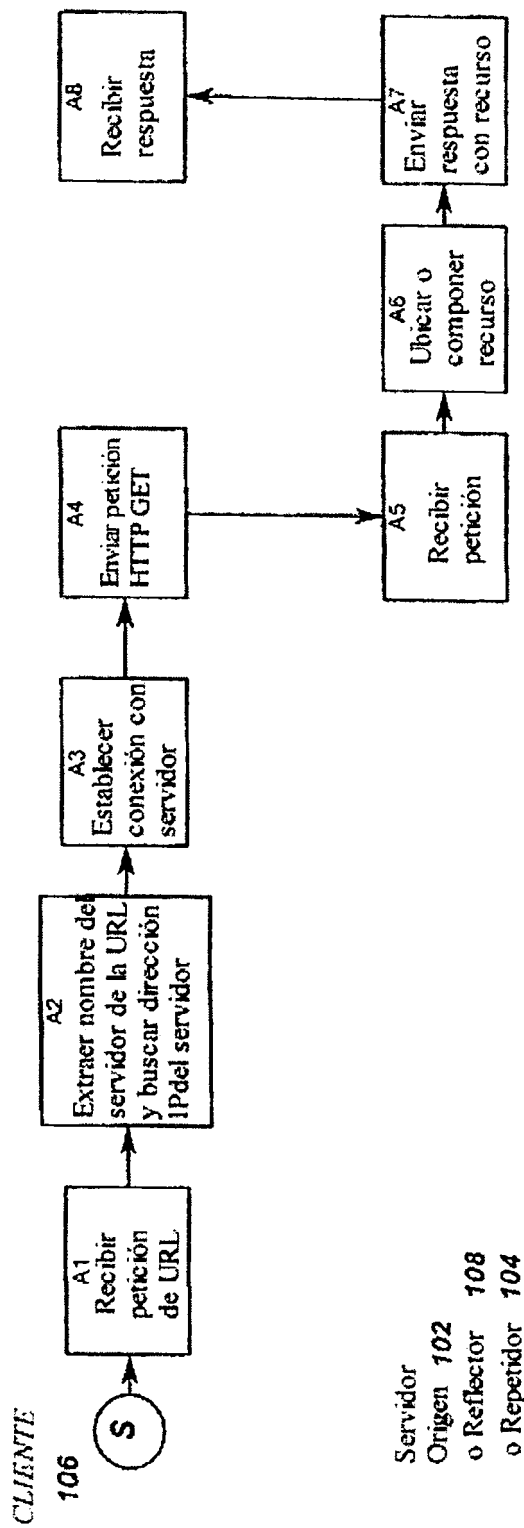


Fig. 2

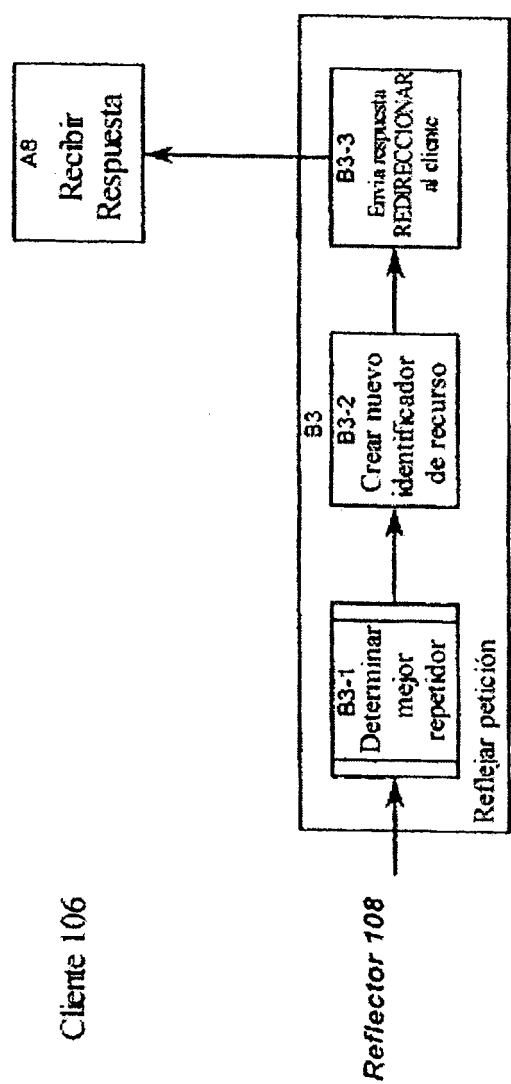


Fig. 4

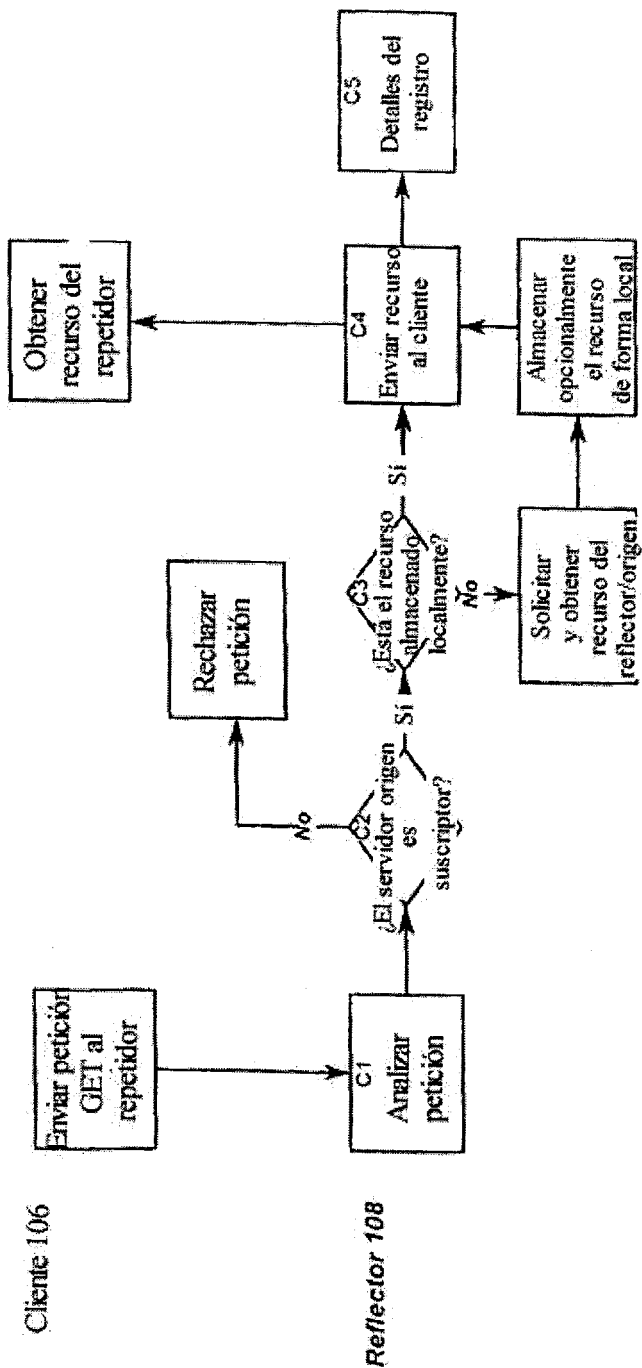


Fig. 5

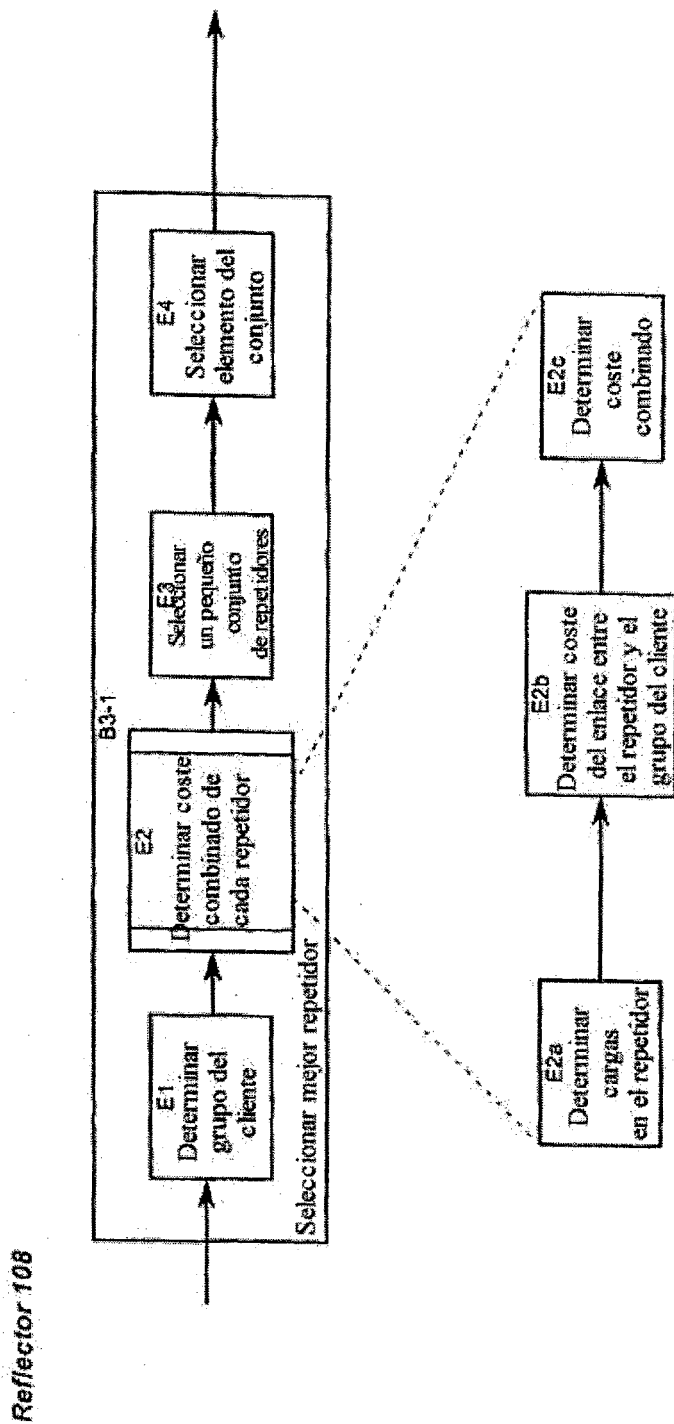


Fig. 6