GNUTWARE: MIDDLEWARE PARA SOPORTAR SERVICIOS DISTRIBUIDOS EN REDES COMPAÑERO A COMPAÑERO

Gabriel H. Tolosa y Fernando R. A. Bordignon
{tolosoft, bordi}@unlu.edu.ar

Universidad Nacional de Luján
Departamento de Ciencias Básicas
Cruce de rutas nacionales 5 y 7 - Luján (6700)
República Argentina

RESUMEN

En el presente trabajo, se describe un prototipo de middleware que brinda soporte de comunicaciones a aplicaciones de usuario final sobre una red de aplicación (overlay network) que opera bajo el modelo de comunicaciones compañero a compañero (P2P). La arquitectura propuesta está basada en una red Gnutella y un prototipo de middleware codificado en lenguaje Java que brinda acceso a dicha red a diferentes aplicaciones. A modo de ejemplo, se escribió un servicio de consulta de tiempo que opera distribuida y puede servir como soporte de un mecanismo de sincronización de relojes dentro del sistema. La arquitectura propuesta se presenta como una solución modular a los efectos de dividir la funcionalidad en lo que respecta por un lado a la red p2p y por otro a las aplicaciones, a través de una interfase normalizada.

Palabras clave: compañero a compañero, middleware, aplicaciones distribuidas, red de recubrimiento.

ABSTRACT

In this work, a prototype of middleware is described that offers support of communications to applications of end user on an application network that operates under the model of communications peer to peer. The architecture is based on Gnutella and we present a prototype of middleware (codified in Java) that offers access to this network to different applications. This system is modular solutions to the effects of divide the functionality in concerns with a side of the network p2p and by another one to the applications, through a standardized interface.

Keywords: peer to peer, middleware, distributed applications, overlay network.

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 4 Edición No 2 – Año 2005

INTRODUCCIÓN

Los sistemas compañero a compañero permiten que computadoras de usuario final se conecten directamente a los efectos de formar comunidades, cuya finalidad será el compartir información y servicios computacionales [3] [4] [14]. En este modelo, se toma ventaja de recursos existentes en los extremos de la red, tales como ciclos de CPU y espacio de almacenamiento.

La primera característica que define a un nodo en un sistema P2P, es que cumple tanto el rol de cliente como de servidor (en la terminología se lo denomina *servent*, palabra que deriva de la conjunción de los términos *serv*er-*client*). Este modelo, basado en interacciones entre pares, puede reemplazar al paradigma cliente/servidor, donde cada nodo es capaz de generar y contestar consultas de otros nodos. Desde el punto de vista de la comunicación, las interacciones son simétricas.

Si un sistema P2P está íntegramente conformado por nodos que tienen iguales roles y ninguno cumple tareas administrativas o especiales, se lo considera P2P puro o completamente descentralizado [5]. En cambio si algún nodo posee alguna característica diferente que le hace cumplir con tareas centralizadas, entonces al sistema se lo considera híbrido. Según Intel, uno de los más grandes beneficios de P2P es el concepto de comunidad, dado que hace posible que los usuarios se organicen ellos mismos en grupos ad hoc, y puedan eficientemente y de forma segura llenar requerimientos, compartir recursos, colaborar y comunicarse. [1]

Se plantea una división de la funcionalidad de la red de aplicación y de las aplicaciones propiamente dichas. En el primer caso, se pueden plantear diferentes estrategias para su construcción y mantenimiento, lo que permite tener prestaciones varias. En el segundo caso, la aplicación accede al

middleware a través de una interfase normalizada para utilizar los servicios de propagación de mensajes. La principal contribución de este trabajo es:

- Un modelo operativo de utilización de una red basada en el protocolo Gnutella [2] como infraestructura de intercambio de mensajes inespecíficos utilizando la técnica de difusión.

- Un modelo de comunicación entre aplicaciones de usuario final, a través de una red de mensajes por difusión.

- Una arquitectura sencilla y robusta para implementar servicios distribuidos sobre redes compañero a compañero.

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 4 Edición No 2 – Año 2005

• Un prototipo de middleware que presenta una interfaz normalizada a las aplicaciones para acceder a la red de transporte de mensajes.

A continuación se describe conceptualmente la arquitectura del middleware, posteriormente se 3 detalla la implementación de GnutWare, acompañada de un ejemplo simple de aplicación. Finalmente se presentan trabajos relacionados, se sugieren trabajos futuros y se describen ciertas consideraciones sobre el modelo presentado.

Arquitectura del Middleware

En la arquitectura propuesta se definen dos niveles ó capas de software (Figura 1) para la implementación de servicios basados en redes compañero a compañero. Una capa de middleware, que toma servicio de la capa de transporte (dado que el modelo opera bajo el juego de protocolos TCP/IP), cuyo objetivo es propagar mensajes y permitir que los usuarios accedan a la red, y una capa de aplicación, que toma servicio del middleware, cuyo objetivo es implementar servicios de usuario final tales como sistemas de archivos, búsquedas ó procesamiento distribuido.

Este modelo permite separar la funcionalidad de cada capa de manera que se puedan desarrollar aplicaciones que corran sobre una red dada, o bien distintas redes de aplicación (con características propias) que se adecuen de mejor manera a la necesidad del servicio. Por ejemplo podría existir un servicio de mensajería, el cual opere sobre una red de aplicación que implementa mecanismos de protección de privacidad ó sobre otra red que implemente estrategias destinadas a garantizar anonimato. De igual manera una red de aplicación debería soportar distintos servicios, tales como mensajería o computación distribuida.

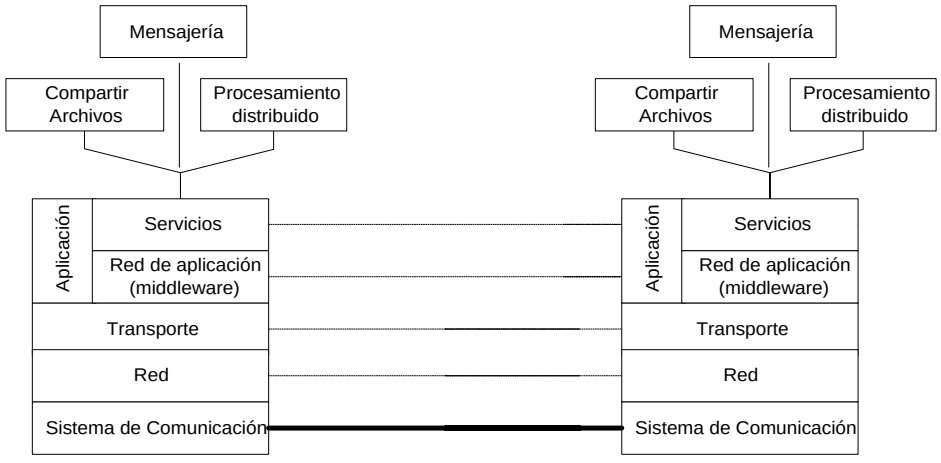

Figura 1 – División por capas de la funcionalidad de la red P2P

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 4 Edición No 2 – Año 2005

En cada nodo compañero, las dos capas de software deben proveer un conjunto de funciones, de manera tal de poder asegurar el cumplimiento de los objetivos pero manteniendo el principio de división de tareas. A nivel de middleware debe:

- Definir la red a nivel de aplicación
- Permitir que la capa de servicio acceda a la red.
- Propagar mensajes propios y de terceros (mediante estrategias de ruteo)
- Tender a mantener la existencia de la red (obteniendo y brindando información de puntos de acceso a la red alternativos)
- Mantener conexiones simultaneas con otros pares
- Proporcionar a la capa de servicio información de puntos de acceso alternativos

Para cumplir con estas funciones, el modelo de comunicación que se propone se basa en dos tipos de mensajes:

Mensajes de exploración

Tienen por finalidad el anuncio del ingreso de un nuevo nodo a la red, a los efectos de recopilar información, a través de mensajes respuesta, de otros nodos existentes.

Mensajes de servicio

Son generados y respondidos únicamente por los nodos que brindan el servicio y permiten implementar las solicitudes y respuestas de los servicios que se implementen.

Y a nivel de capa de servicio, las funciones requeridas son:

- Implementar servicios distribuidos de usuario final
- Comunicarse con los demás nodos pares

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 4 Edición No 2 – Año 2005

- Tender a mantener su existencia en la red (obteniendo información de puntos de acceso alternativos).

- Mantener conexiones simultaneas con uno ó más nodos pares

Toda comunicación extra necesaria (más allá de los mensajes de consulta y respuesta definidos) se realizará de manera independiente de la red de aplicación, es decir, de manera directa entre dos nodos, ya sea mediante conexiones TCP ó mensajes UDP. Esta característica responde a un criterio de optimización de la red de aplicación. Como la red es un espacio público de comunicación, solo se permiten mensajes generales y no interacciones directas particulares.

Para ejemplificar el modo de operación de la red de aplicación y su mecanismo de propagación de mensajes, se puede hacer una analogía con el funcionamiento de una red tipo Ethernet. Aquí, cualquier nodo conectado a la red puede emitir un mensaje que alcanzará a todos los destinatarios, y solo aquellos que se hagan eco del mensaje lo procesarán y eventualmente lo contestarán. Esta red de aplicación es un “pasillo virtual”, cuyo ámbito de operación es una intranet ó Internet, y sus receptores todos los nodos participantes de la misma.

En el prototipo propuesto, la red de aplicación está formada por un conjunto de servents Gnutella (Gnodo), utilizando el software Gnut v.056 para Linux. Esta red p2p opera como un backbone para permitir prestar diferentes servicios distribuidos, donde una característica importante es la distribución total de las funciones, ya que no existen nodos con características especiales.

Descripción de GnutWare

El middleware GnutWare brinda servicio a las aplicaciones para acceder a la red de aplicación que sirve como transporte de mensajes genéricos. Los Gnodos se configuraron para que no posean archivos compartidos en sus sistemas locales, por lo cual se comportan simplemente como ruteadores de mensajes dentro de la red, sin prestar servicio alguno.

GnutWare se encuentra programado completamente en Java, por lo tanto es completamente portable a cualquier plataforma que soporte la ejecución de la Máquina Virtual Java. Básicamente se encuentra dividido en dos módulos principales (Figura 2):

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 4 Edición No 2 – Año 2005

- un módulo Gnodo, que permite conectarse a un Gnodo a partir de contar la dirección IP y puerto de alguno existente en la red de aplicación

- un módulo Network_Sap, que es una interfase para las aplicaciones para posibilitar la utilización del servicio.

El módulo Gnodo, implementa el protocolo Gnutella a los efectos de conectarse a la red e intercambiar mensajes de exploración (Ping/Pong) y de servicio (Query/Query_Hit), mientras que el módulo Network_Sap, acepta conexiones entrantes de alguna aplicación y luego acepta mensajes de servicio y devuelve las respuestas.

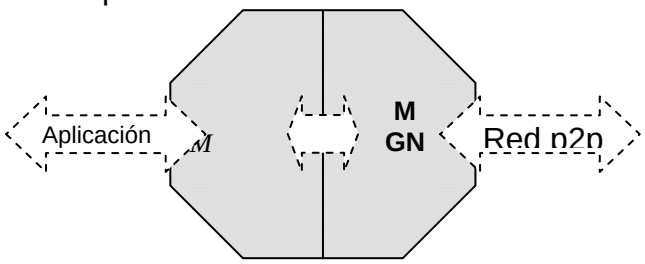

Figura 2 – Arquitectura de GnutWare. MNS, Módulo Network Sap. MGN, módulo Gnodo.

Cuando los módulos gnuWare se conectan a algún Gnodo, se forma una red de intercambio de mensajes inespecífica basada en el modelo compañero a compañero, disponible para implementar diferentes servicios distribuidos (Figura 3).

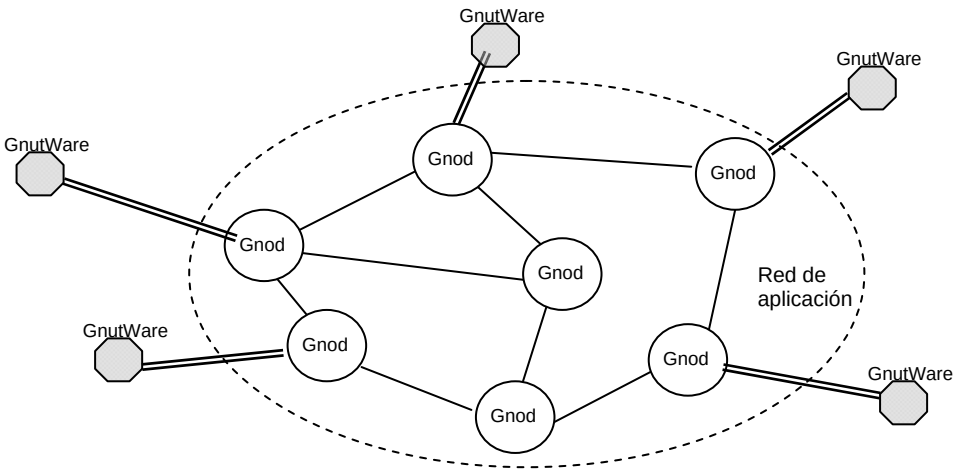

Figura 3 – Red inespecífica de intercambio de mensajes

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 4 Edición No 2 – Año 2005

Para acceder a dicha red, las aplicaciones debe establecer una conexión TCP contra algún módulo GnutWare y ejecutar un simple protocolo de acceso. La aplicación envía una cadena de conexión a nivel aplicación

GNUTWARE CONNECT\n\n

Luego se solicita a la aplicación que identifique el código de servicio y el identificador de servicio utilizando las primitivas

GNUTWARE SERVICE CODE?\n\n

Y

GNUTWARE SERVICE ID?\n\n

En este orden. La aplicación debe contestar con un entero de 2 bytes en el primer caso y con un entero de 4 bytes en el segundo. Estos datos se utilizan para permitir multiplexar a nivel de aplicación diferentes servicios y aplicaciones a los que un mismo módulo GnutWare puede brindar acceso a la red p2p. A partir de finalizar esta etapa de protocolo, la aplicación está en condiciones de enviar solicitudes de servicio y de recibir respuestas.Cuando un módulo GnutWare recibe una solicitud de servicio, debe armar un mensaje Gnutella tipo QUERY (manteniendo la estructura de datos Gnutella) y enviarlo al Gnodo con el que está vinculado, el cual se encargará de su propagación. Cuando se recibe un mensaje Gnutella tipo QUERY_HIT desde la red p2p, se pasa la respuesta a la aplicación. Las respuestas viajan dentro de la estructura de datos de una QUERY_HIT, como si fuese un mensaje Gnutella típico. Se debe respetar esta estructura ya que los Gnodos de la red de backbone solo propagan mensajes Gnutella bien formados.

A los efectos de validar el comportamiento del middleware y de la red de aplicación propuestos, se codificó una simple aplicación de solicitud de tiempo. Dicha aplicación puede servir como soporte de un mecanismo de sincronización de relojes dentro del sistema.

Un Ejemplo de Aplicación: Servicio Cooperativo de Tiempos

El servicio cooperativo de tiempos (p2pTime) pretende ser un ejemplo de una posible aplicación para correr en un ambiente distribuido soportado por GnutWare y una red p2p (Figura 4).

Una aplicación p2pTime accede a la red solicitando los tiempos (fecha y hora) de todos los demás nodos p2pTime dentro del sistema. Cuando llegan las solicitudes, todos los módulos p2pTime responden con su tiempo local al solicitante. Nótese que éstos módulos no deben preocuparse por determinar quién es el solicitante ni dónde se encuentra, ya que estas funciones son

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 4 Edición No 2 – Año 2005

provistas por la red de aplicación subyacente. Cuando el solicitante recibe todas las respuestas, puede utilizarlas a los efectos de establecer un tiempo global aproximado del sistema ó para sincronizar relojes.

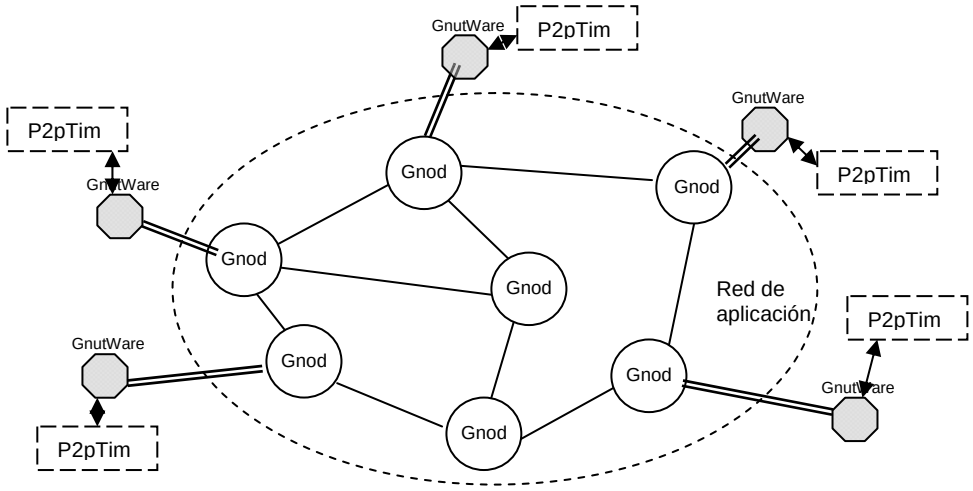

Figura 4 – Aplicaciones interactuando con el middleware

Las aplicaciones se conectan a un módulo GnutWare a través de la interface provista por éste, es decir, estableciendo una conexión TCP a un puerto conocido. Cuando una envía una solicitud de tiempo, ésta es propagada hacia todos, los cuales responderán a la misma. Esta estrategia de difusión es propia de la red Gnutella, por lo tanto las respuestas volverán por el mismo camino que las consultas.

El protocolo de la aplicación es sencillo. La primitiva de solicitud incluye dos caracteres que denotan que se trata de una solicitud (QY) y luego la misma:

QY:TIME, PLEASE!\n

Mientras que la respuesta tiene la misma estructura: dos caracteres que denotan que se trata de una respuesta y luego la hora anunciada:

QH:WED 29-11-2002 16:02:11\n

Se pueden recibir tantas respuestas como aplicaciones p2pTime hayan recibido la consulta. Téngase en cuenta que como se trata de una red de transporte de mensajes basada en Gnutella el alcance de las consultas está limitado por el horizonte definido por el valor del campo TTL del encabezado.

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 4 Edición No 2 – Año 2005

En la implementación de GnutWare, el TTL se inicializa en 7 pero es decrementado en uno al llegar al primer Gnodo.

Finalmente, una vez recibido todos los tiempos se puede aplicar un algoritmo de promedios a los efectos de establecer el tiempo global aproximado. Esta cuestión se puede resolver utilizando los algoritmos típicos de sincronización de relojes utilizados el área de sistemas distribuidos.

Trabajos Relacionados

Nuestra aplicación se enmarca dentro del concepto de red de recubrimiento (overlay network). Tales redes son sistemas que se implementan sobre Internet a los efectos de proveer una plataforma de comunicaciones que sea eficiente, puede reconfigurarse dinámicamente en pos de un buen desempeño y posea mecanismos que la hagan tolerante a fallas. Un ejemplo popular son las redes privadas virtuales (VPN) dado que es posible configurar una red IP restringida sobre una red pública de datos.

De forma más precisa, las redes de recubrimiento se clasifican en a) Redes de distribución de contenidos, tales como Akamai [6], FFnet [7], y Overcast [8], b) Sistemas de ruteo, tales como Xbone [9] and RON [10] y c) sistemas de localización de recursos, tales como Tapestry [11]; como Chord [12] y Content-Addressable Networks (CAN) [13].

Las redes de recubrimiento son una indicación positiva de que es posible construir sistemas distribuidos de gran escala que sean simples y robustos.

Trabajos Futuros

Actualmente, en el Laboratorio de Redes de la UNLu, se están diseñando dos servicios tendientes a operar sobre el middleware descrito. El primero, define un modelo de búsqueda distribuida sobre una comunidad de motores de consulta; donde cada motor al recibir una consulta de usuario, la procesa localmente y en paralelo la distribuye a sus pares, utilizando la red de recubrimiento. Finalmente, una vez que tales pares responden, fusiona las respuestas y las presenta al usuario. El segundo servicio en desarrollo está relacionado con el cómputo masivo colaborativo, donde computadoras de usuario final, cuando poseen recursos ociosos, procesan unidades de trabajo de terceros nodos. Tal servicio utiliza la red de recubrimiento a los efectos de obtener nodos que deseen cooperar aportando sus recursos.

Por otro lado, una de las líneas de investigación que se están llevando cabo, en el mencionado laboratorio, tiene que ver con mejoras al modelo de mantenimiento de la red Gnutella, a los efectos de lograr mayor escalabilidad

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 4 Edición No 2 – Año 2005

y un mejor desempeño. Tales investigaciones se orientan a definir una nueva política de comunicación de los gnodos con sus vecinos y a establecer mejoras en las reglas de reenvío de mensajes.

Consideraciones Finales

El prototipo de middleware resulta una solución válida a los efectos de brindar un punto de acceso a una red p2p a diferentes aplicaciones que requieren operar en un ambiente distribuido. En este caso, el servicio que se brinda de propagación de mensajes se hace en modo broadcast, ya que la red subyacente se basa en Gnutella.

En esta arquitectura propuesta, se estima que la red de aplicación opera de forma más eficiente, ya que los nodos de los extremos no generan mensajes de exploración. Esta tarea solo la realizan los Gnodos, por lo cual la sobrecarga de la red es menor. Esta situación favorece los límites de escalabilidad encontrados a Gnutella.

REFERENCIAS BIBLIOGRÁFICAS

[1] Barkay. D. (2000). “An introduction to Peer-to-Peer Computing”. Intel Developer Update Magazine. Febrero.

[2] Bordignon, F. y Tolosa. G. (2001) "Gnutella: Distributed System for Information Storage and Searching. Model Description” . JIT- Journal of Internet Technology. Taipei (Taiwan)l. 2(5).

[3] FIPA. (2000). “FIPA Peer-to-Peer Positioning Paper”. Technical Report F-OUT-00076, Foundation for Intelligent Pyhsical Agents, Diciembre.

[4] Minar, N. y M. Hedlund, M. (2001). “A Network of Peers: Peer-to-Peer Models Through the History of the Internet”,. A. Oram, editor, “, “Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology”. Capítulo 1. O’Reilly & Associates, Marzo.

[5] Yang, B. y H. Garcia-Molina, H. (2001). “Comparing Hybrid Peer-to-Peer Systems”. Proceedings of the 27th VLDB Conference, Roma, Italia.

[6] Akamai Technologies, Inc. [En línea] <http://www.akamai.com/> [2004, enero 10].

[7]. “FastForward Networks: Broadcast overlay architecture”, [En línea]. <www.ffnet.com/pdfs/BOA-whitepaperv6.PDF> [2003, noviembre 22]

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 4 Edición No 2 – Año 2005

[8]. Jannotti, J., Gifford, D. K., Johnson, K. L., Kaashoek, M. F., y Jr., J. W. O. (2000). “Overcast: Reliable multicasting with an overlay network”. Proc. 4th USENIX OSDI. 197-212.

[9]. Touch, J., y Hotz, S. (1998). “The X-Bone”. Third Global Internet Mini-Conference in conjunction with Glo becom '98, Sydney, Australia.

[10] Andersen, D., Balakrishnan, H., Kaashoek, M. F., y Morris, (2001). R. “Resilient Overlay Networks”. Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP '01) Chateau Lake Louise, Canada.

[11]. Zhao, B.; Ling Huang; Stribling, J.; Rhea, S.; Joseph, A. y Kubiatowicz, J. (2003). “Tapestry: A resilient global-scale overlay for service deployment”. IEEE Journal on Selected Areas in Communications.

[12] Stoica, I.; Morris, R.; Liben-Nowell, D.; Karger, D.; Frans Kaashoek, M.; Dabek, F. y Balakrishnan, H. (2003). “Chord: a scalable peer-to-peer lookup protocol for internet applications”. IEEE/ACM Trans. Netw. 11(1),17-32

[13]. Ratnasamy, D.; Francis, P.; Handley, M.; Karp, R. y Schenker, S. (2001). "A Scalable Content-Addressable Network". Proc. SIGCOMM, San Diego, CA, Agosto 2001.

[14] Bordignon, F. y Tolosa G. (2003) “Redes Compañero a Compañero: Conceptos y Tendencias de Aplicación”. Novática. 166,(nov-dic).

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 4 Edición No 2 – Año 2005