aDICS: Modelo de Índice Distribuido sobre una Red P2P para Búsquedas por Contenido

Tolosa, Gabriel H.[1]; Bordignon, Fernando R. A. y Peri, Jorge A.
Universidad Nacional de Luján
Departamento de Ciencias Básicas
{tolosoft, bordi, jperi)@unlu.edu.ar

[1] Becario de Investigación. Secretaría de Investigación y Postgrado. Universidad Nacional de Luján

Resumen

Se presenta aDICS, un modelo de índice distribuido sobre una red compañero a compañero que soporta la indexación completa de documentos de texto y permite búsquedas por palabras clave. Se propone dividir el índice invertido de términos sobre un subconjunto de los nodos de la red, de acuerdo a las letras del alfabeto. Los nodos que poseen documentos para compartir publican sus términos en algunos otros nodos del índice distribuido y – con esta información – luego se resuelven las consultas.

Se propone un modelo de tráfico y se lo evalúa experimentalmente bajo diferentes configuraciones. Las pruebas muestran que esta propuesta brinda capacidades de búsqueda similares a sistemas como Gnutella, siendo aDICS más eficiente en cuanto al consumo de ancho de banda. Además, se evaluó una política de selección de nodos índice a los cuales consultar que permite reducir la cantidad de mensajes requeridos.

Palabras clave: P2P, Índice distribuido, búsquedas por contenido, ruteo de consultas

aDICS: Modelo de Índice Distribuido sobre una Red P2P para Búsquedas por Contenido

Tolosa, Gabriel H.[1]; Bordignon, Fernando R. A. y Peri, Jorge A.
Universidad Nacional de Luján
Departamento de Ciencias Básicas
{tolosoft, bordi, jperi)@unlu.edu.ar

[1] Becario de Investigación. Secretaría de Investigación y Postgrado. Universidad Nacional de Luján

1 – Introducción

Las redes compañero a compañero (P2P) [4] son una tema de investigación que se ha desarrollado de manera sostenida durante los últimos años, especialmente con aplicaciones de intercambio de archivos como Napster [12] y Gnutella [3] primero y Fastrack [7] y Kazaa [10] luego.

Estas redes están conformadas – en general – por nodos que poseen las mismas capacidades y pueden comunicarse en intercambiar información de manera directa. Como infraestructura de comunicaciones permiten construir sistemas de información distribuidos de forma descentralizada, a partir de los recursos disponibles de cada uno de los nodos que la forman. Además, los sistemas basados en este modelo presentan características interesantes como tolerancia a fallas, auto-organización y balanceo de cargas, las cuales permiten que sean una plataforma válida y robusta para la construcción de diferentes servicios distribuidos.

La utilización de servicios basados en redes P2P crece permanentemente, incluso al punto de sobrepasar – en algunos casos – a los demás servicios en cuanto al tráfico generado en Internet [9], en particular, sistemas de intercambio de archivos. Estos sistemas – en general – identifican a los recursos mediante su nombre y un conjunto de metadatos para soportar búsquedas. Sin embargo, esta situación es ineficiente cuando los recursos son documentos de texto y las funciones de búsqueda no operan sobre el contenido del documento. Un aspecto clave es la posibilidad de soportar búsquedas eficientes y la recuperación de la información allí contenida, lo que presenta un desafío interesante en el diseño de técnicas y estrategias alternativas.

Las técnicas más simples se basan en la distribución de la consulta a todos los nodos de la red ó a un subconjunto de éstos [26]. Algunas propuestas utilizan enfoques híbridos [23] con servidores centralizados, en algunos casos combinadas con técnicas de resumen para reducir la cantidad de información intercambiada entre los nodos [1]. Otros enfoques se basan en la utilización de tablas de dispersión distribuidas [6, 14]. No obstante, algunos investigadores del área han reconocido las limitaciones de estos sistemas para la recuperación de información a partir del contenido completo de los documentos de texto y proponen continuar la investigación en redes P2P de las características básicas de Gnutella [5].

La principal contribución de este trabajo es un modelo de índice distribuido sobre una red P2P que soporta la indexación completa de documentos y la distribución de los términos. Se divide el índice invertido de términos y nodos de acuerdo a las letras del alfabeto, es decir, diferentes nodos serán responsables de la parte del índice que corresponde a los términos de una determinada letra. Se determina un mecanismo de selección de nodos índice y una estrategia de ruteo de consultas. La

idea base es reemplazar la estrategia de reenvío de consultas – tal como ocurre en Gnutella – por la de mantener un esquema de índice distribuido que opere como primer punto de resolución de tales consultas. Una vez obtenida una respuesta inicial proveniente de nodos que gerencien el índice distribuido se consulta a nodos que – potencialmente – pueden satisfacerla, generando menor tráfico en la red.

Además, se presentan experimentos de evaluación del modelo propuesto en base a la generación de una topología de red y a la indexación de un *corpus* real de archivos, sobre el cual se realizan diferentes tipos de consultas. Se definen parámetros de operación que permiten determinar la eficiencia del modelo en cuanto al tráfico generado por consulta y se compara sus prestaciones respecto de Gnutella.

En la siguiente sección se presentan trabajos relacionados y en la sección 3 se describe el modelo propuesto. La sección 4 contiene la metodología y la configuración de los experimentos. En las secciones 5 y 6 se presentan los resultados y un análisis comparativo. Finalmente, las conclusiones y trabajos futuros se enuncian en la sección 7.

2 – Trabajos Relacionados

Existen diversos enfoques para resolver el problema de las búsquedas en un red compañero a compañero. En general, hay dos enfoques sobre los que se proponen estrategias que pretenden resolver este problema. Por un lado, se encuentran los sistemas que se basan en el reenvío de consultas, de manera similar a Gnutella. Sin embargo, debido al volumen de tráfico que genera este protocolo, se implementan diversas variantes para aumentar la eficiencia.

Algunas propuestas como BFS Dirigido (*Directed BFS)*, Profundización Iterativa (*Iterative Deepening)* e Índices Locales (*Local Indices)* se describen y evalúan en [24]. En el primer caso, se propone el reenvío de la consulta a un subconjunto de los nodos con los cuales se mantienen conexiones. La decisión sobre cuáles nodos seleccionar se realiza en base a estadísticas sobre las respuestas a las consultas anteriores, como por ejemplo cantidad de respuestas entregadas, latencia, cantidad de saltos y demás. En Profundización Iterativa se propone realizar varias búsquedas de profundidad *D* simultáneas, ampliando en cada iteración el valor del parámetro *D*, hasta alcanzar un nivel de satisfacción. Finalmente, utilizando Índices Locales un nodo mantiene un índice sobre el contenido – en temas y no en términos – de los nodos a *r* saltos del éste. Con esta información se determina la política de reenvío.

En la misma línea, en [26] se propone una técnica denominada Inundación Híbrida Periódica (*Hybrid Periodical Flooding)*, como una variante del algoritmo *K-Walker* [11]. Ésta emplea un mecanismo de búsqueda adaptivo que permite determinar el número de nodos a los que se les reenvía la consulta.

Por otro lado, se encuentran los sistemas que utilizan Tablas de Dispersión Distribuidas (DHT – *Distributed Hash Tables*) [6, 14, 17] sobre las cuales se mapean identificadores de nodos y de recursos sobre un único espacio n-dimensional. Garcés y otros [8] proponen un sistema con un mecanismo que permite localizar datos en una red P2P con información incompleta, el cual funciona sobre una DHT. Los objetos a localizar son documentos caracterizados por una estructura de atributos (metadatos) fija – denominada descriptor de archivo – de los cuales se mapean partes separadas o combinaciones de éstos sobre la DHT. Si bien permite aumentar las capacidades de

estos modelos, su aplicación se encuentra limitada a búsquedas exactas sobre los términos de los descriptores de los documentos.

En [15], los autores presentan un sistema de localización de contenido basado en un índice distribuido sobre una red P2P que se puede utilizar como motor de consultas. Básicamente, se opera sobre una DHT sobre la cual se mapean los recursos junto con un conjunto de palabras clave que lo caracterizan. El sistema implementa – además – un conjunto de técnicas que permiten reducir el tráfico generado y aumentar su escalabilidad, especialmente en búsquedas por múltiples términos.

Baquero y López [2] proponen un modelo de indexación de contenido con balanceo de cargas entre los nodos basándose en tablas de dispersión distribuidas. El requerimiento es construir un índice distribuido cuya carga se distribuya entre los nodos hasta un valor máximo, incluso cuando se indexan términos muy populares. Sobre este concepto, definen un mecanismo de división denominado Agrupación en Burbujas (*Bubble Grouping*) en el cual un conjunto de nodos se agrupan para gerenciar un conjunto de claves manteniendo réplicas de su estado y facilitar la tolerancia a fallas.

3 – Modelo Propuesto

La idea principal en este modelo es dividir un índice invertido de términos y nodos sobre una red P2P. La división se realiza de acuerdo a las letras del alfabeto, es decir, un nodo gerencia un espacio del índice distribuido que corresponde a sólo una letra, y existe la posibilidad que más de un nodo gerencien la misma letra, pero con diferente contenido.

Todos los nodos de la red deben mantener una tabla de contactos con – al menos – una dirección de un nodo índice por letra, es decir, la tabla de contactos contendrá 26 entradas (Figura 1). De cada documento para compartir se extraen sus términos representativos y se publican en algunos nodos de la red. Inicialmente, se tienen en cuenta solo aquellos términos de tres caracteres o más, aunque puede mejorarse esta selección con técnicas del área de recuperación de información, cuestión que está fuera del alcance de este trabajo.

La selección de los nodos índice dependerá de la letra inicial del término y del estado de su tabla de contactos (TC). Por ejemplo, para publicar el término “colina” se enviará un mensaje de publicación a un nodo que tenga bajo su dominio la parte del índice correspondiente a la letra “C”. Luego, cada nodo que gerencie una letra deberá tener una porción del índice invertido (TI) donde se asocie el término con la lista de nodos que contienen documentos con el mismo (Figura 2).

L	Dirección del nodo
A	646; 125
B	154
…	
Y	147; 287; 156
Z	200; 874

Figura 1 – Tabla de contactos de un nodo

Términos	Nodos
Casa	154; 165; 947; 877
Colina	472; 554; 489
Corona	587; 154; 554; 145
…	
…	

Figura 2 – Porción del índice invertido de un nodo

El proceso completo requiere: a) seleccionar los documentos a compartir, b) seleccionar de los documentos los términos representativos (indexación local), c) publicar los términos en los nodos correspondientes y d) actualizar o confirmar la publicación. Luego de este proceso el sistema cuenta con información distribuida para soportar el servicio de consulta.

3.1 – Ingreso de nodos a la red

Para acceder a la red, un nodo X debe conocer la dirección de otro nodo participante P y establecer un contacto con éste. Luego, es necesario determinar si X va a operar como parte del índice distribuido. Esto se realiza mediante una función que pondera los recursos con que cuenta el nuevo nodo X para tal tarea, por ejemplo: potencia de CPU y ancho de banda de su red. Si X operacomo parte de índice, es necesario determinar qué letra del alfabeto le corresponde ya que existirán en la red más de un nodo que maneje la misma letra. Esto se realiza mediante el algoritmo que se presenta en la figura 3.

```
Si (ManejarIndice(P(ndx)) {
   Si (TablaIncompleta) {
          LT = SelecLetraEntradaVacía
               AsignarLetra(LT)
   }
   sino {
          LT = SelecLetraAlAzar(P(l))
   }
}
sino {
     LT = ""
}
Retornar LT
```

Figura 3 – Algoritmo para la asignación de letra a un nodo índice

3.2 – Publicación de términos

Para permitir a los demás nodos realizar búsquedas basadas en el contenido de sus documentos primero se debe realizar el proceso de indexación local y extraer los términos más relevantes de los mismos. Aquí resulta importante – como se mencionó antes – aplicar técnicas del área de recuperación de información para determinar el conjunto de términos a publicar, por ejemplo, por frecuencia de aparición o posición dentro del documento. Esto tiene como objetivo reducir la cantidad de términos de manera de que el proceso de publicación resulte eficiente. Una vez seleccionados los términos se agrupan por letra inicial y se selecciona de la tabla de ruteo un nodo $S_{i,l}$ encargado de la parte del índice correspondiente a la letra correspondiente. Al nodo $S_{i,l}$ se le envía un mensaje de publicación anunciando el término o los términos por los cuales puede resolver consultas. En la figura 4 se muestra el proceso de publicación.

En este ejemplo existen dos nodos con términos a publicar. El nodo 100, con los términos “casa” y “árbol” y el nodo 140 con los términos “árbol” y “colina”. El nodo 100 requiere publicar el primer término para lo cual revisa su tabla de contactos y en la entrada para la letra “C” posee dos direcciones: 102 y 165. Toma la primera y envía al nodo un mensaje de publicación del término “casa” (paso 1). El nodo receptor verifica que la inicial del término corresponde con la letra de la cual es responsable e incluye una entrada en su tabla de índice (TI) (paso2). A continuación, el nodo 100 repite el proceso para el término “árbol”. En este caso envía el mensaje de publicación al nodo 244 (paso 3) el cual agrega la entrada en su TI correspondiente (paso 4).

Este proceso se repite para el nodo 140 que requiere publicar los términos “árbol”, “colina” y “bota”. En este caso se enviarán mensajes de publicación a los nodos 244 y 102 respectivamente (pasos 5 a 8) para los dos primeros, mientras que por el término “bota” se realizará una publicación en la TI local ya que el nodo 140 es responsable de parte del índice correspondiente a la letra “B”.

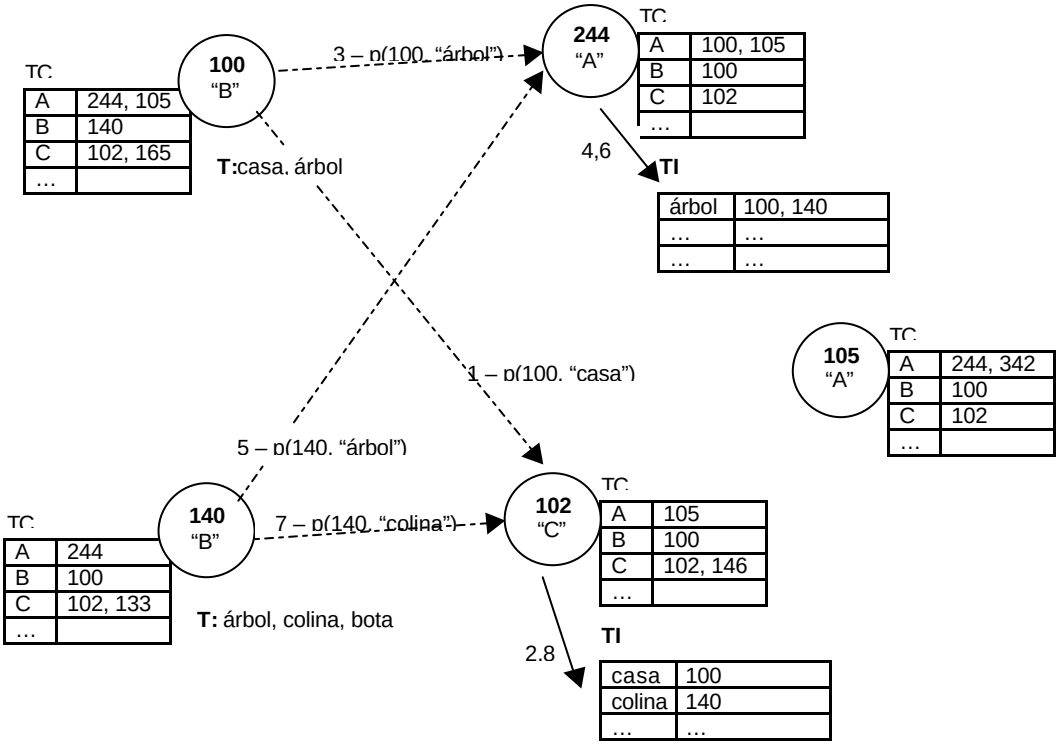

Figura 4 – Proceso de publicación de términos

3.3 – Resolución de consultas

Para resolver una consulta un nodo utiliza la información de su TC para identificar los nodos responsables del índice que le corresponde a cada uno los términos. Por ejemplo, supóngase que en el nodo 105 se requiere resolver la consulta por los términos "árbol" y "colina". En primer lugar, se consulta la tabla de contactos local para obtener las direcciones de los nodos responsables de los índices (NI) por las letras iniciales de cada término:

NI_A = consulta(TC, "árbol") => NI_A = {244, 342}
NI_c = consulta(TC, "colina") => NI_c = {102}

Luego, se consulta a los NI correspondientes. Primero, se realiza una consulta al nodo 244 por direcciones de nodos que publicaron contener el término "árbol" y se obtiene un conjunto respuesta surgido de la tabla de índice del nodo 244. Luego, se consulta a 102 por el término "colina".

$R_{(\text{"árbol"})}$ = consulta(244, "árbol") => $R_{(\text{"árbol"})}$ = {100, 140}
$R_{(\text{"colina"})}$ = consulta(102, "colina") => $R_{(\text{"colina"})}$ = {140}

Como la consulta corresponde a una conjunción de ambos términos, la respuesta final se obtiene calculando la intersección de los dos conjuntos de respuestas parciales:

$R = R_{(\text{"árbol"})}$ ***INTERSEC*** $R_{(\text{"colina"})}$ => R = {140}

Esto significa que el nodo 140 ha publicado de manera distribuida (en dos nodos en este caso) que contiene ambos términos en uno o varios documentos, pero no asegura que los contenga dentro de un mismo documento. El paso posterior es enviar un mensaje al nodo 140 por la expresión de consulta completa y luego se evaluará la respuesta.

4 – Metodología Experimental

A los efectos de validar la propuesta se construyó experimentalmente una red compañero a compañero de 10000 nodos y un alfabeto completo de 26 letras. Luego, se generaron aleatoriamente entradas en las tablas de contactos para cada uno. Para evaluar el tráfico generado se definieron 4 configuraciones (Tabla 1) que permitieron testear la distribución de los términos sobre el índice distribuido y luego el comportamiento del proceso de consulta a partir de un nodo cualquiera.

	Exp #1	**Exp #2**	**Exp #3**	**Exp #4**
Cantidad de nodos en la red	10000	10000	10000	10000
P(ndx)	0.25	0.25	0.25	0.25
P(doc)	0.50	0.50	0.75	0.75
Cantidad máxima de documentos por nodo qMaxDocsxNodo	200	400	200	400

Tabla 1 – Datos de configuración de los diferentes experimentos

La probabilidad de que un nodo maneje un índice o no, P(ndx), se determinó empíricamente en [22] mediante comparativas de tráfico generado y cantidad de índices para una letra particular. La probabilidad de que un nodos posea documentos, P(doc) y la cantidad máxima de documentos por nodo qMaxDocsxNodo se estimaron sobre la base que estudios previos [18] muestran que en Gnutella aproximadamente el 75% de los clientes comparten 100 archivos o menos y – según sus autores – hay una amplia variación entre la cantidad de documentos en cada nodo. Dado que Gnutella es un sistema para compartir archivos, sus características brindan una estimación básica que debe ajustarse para sistemas que permiten la indexación completa como el propuesto.

En todos los casos, se utilizó como juego de prueba un *corpus* de documentos correspondiente a la parte número 1 del repositorio de resúmenes de investigación de la National Science Foundation de los años 1990 a 2003 [13]. En total se procesaron 49078 documentos los cuales contaban con 30799 términos diferentes que fueron utilizados para el proceso de indexación.

El software utilizado se codificó en Perl v5.6.1 sobre Linux. El proceso completo se dividió en: a) Generación aleatoria de la red, b) Indexación local y publicación y c) consultas. En este último ítem, se generaron aleatoriamente expresiones de consulta de 1, 2 y 3 términos extraídos del lexicon general. Se realizaron 500 consultas para cada caso y para cada experimento.

5 – Resultados

Primero, se determinó para cada letra la cantidad de nodos que – en promedio – seleccionaron un mismo nodo índice (Qnil), que surge del cociente entre la cantidad de nodos por letra (Qn) sobre la cantidad total de nodos elegidos por letra en tablas de ruteo (Qnrt) de los 10000 nodos de la red. Este parámetro permite estimar la cobertura de un nodo índice. Se observó un valor de Qnil = 213, lo que significa que un NI almacena – en promedio – información de otros 213 nodos de la red.

5.1 – Cálculo de tráfico

Se determinó el costo total de una consulta (Ctc) de acuerdo a:

$$\mathbf{Ctc = Cci + Qri * (Leti + Le + Cq)} \qquad (1)$$

Luego,

$$\mathbf{Cci = 2 (Leti + Le) + Cq + Cr} \qquad (2)$$

y

Cr = Qri * (Li + Ls) (3)

Donde,

Cci	Costo consulta a un nodo índice (y sus respuestas)
Qri	Cantidad promedio de respuestas
Leti	Longitud del encabezado de red y transporte
Le	Longitud del encabezado de aplicación del protocolo
Cq	Costo (en bytes) de la expresión de consulta
Cr	Costo de una respuesta por parte de un nodo índice
Li	Longitud del identificador de nodo
Ls	Longitud del separador

Para los cálculos se tomaron los valores observados en los experimentos (Tabla 2).

Qri	Exp #1	20 nodos	Cantidad de respuestas (de nodos índice) por consultas de un solo término
	Exp #2	22 nodos	
	Exp #3	29 nodos	
	Exp #4	38 nodos	
Leti	40 bytes		Encabezados TCP e IP
Le	20 bytes		Encabezado protocolo
Cq	8 bytes		Longitud promedio de los términos
Li	6 bytes		Dirección del nodo (red)
Ls	1 bytes		Caracter separador

Tabla 2 – Valores de los parámetros utilizados en los cálculos

Para el experimento #2, reemplazando en (3) se obtiene:

Cr = 22 * (6 + 1)

Cr = 154 bytes

Luego, de (2):

Cci = 2 (40 + 20) + 8 + 154

Cci = 282 bytes

Finalmente, de (1) resulta:

Ctc = 282 + 22 * (40 + 20 + 8)

Ctc = 1778 bytes

Por lo tanto, una consulta a un nodo índice y luego el reenvío de la consulta a aquellos nodos que pueden satisfacerla requiere de 1778 bytes. De acuerdo al valor Qnil observado, esto significa que con 1778 bytes se estaría alcanzando al contenido de la letra de los publicado por 213 nodos. En el caso de la configuración del experimento #4, resulta:

Ctc = 394 + 38 * (40 + 20 + 8) = 2978 bytes

En el gráfico 1 se presentan los resultados comparativos de acuerdo a las diferentes configuraciones para consultas por un término. Para el cálculo del tráfico utilizado por las consultas utilizando 2 y 3 términos en la expresión se utiliza como base la fórmula descripta en (1), ajustándola al hecho de que por cada término se debe generar una consulta diferente y que de la cantidad de respuestas recibidas, solo se deben reenviar consultas a los nodos que se encuentren en ambas (AND). De aquí que resulta:

Ctc = (tq * Cci) + Qria * (Leti + Le + Cq) (4)

Donde,

tq	Cantidad de términos de la expresión de consulta
Qria	Es la conjunción (AND) de la cantidad promedio de respuestas (Qri)

En el gráfico 2 se muestra comparativamente el tráfico generado por consultas para tq = 1, 2 y 3. Se puede observar que las consultas por 2 términos resultan más eficientes que las de 1 y 3 términos, contrariamente a lo que se podría esperar. Sin embargo, esto se justifica por el hecho que el costo de agregar un término (respecto del primero) no incide tanto como la cantidad de nuevas consultas que se deben realizar de acuerdo al Qri. En el caso de la comparación con las consultas por 3 términos, resulta que el valor de Qria en ambos casos es cercano y – en este caso – la incidencia del valor de Cci es la que determina las diferencias.

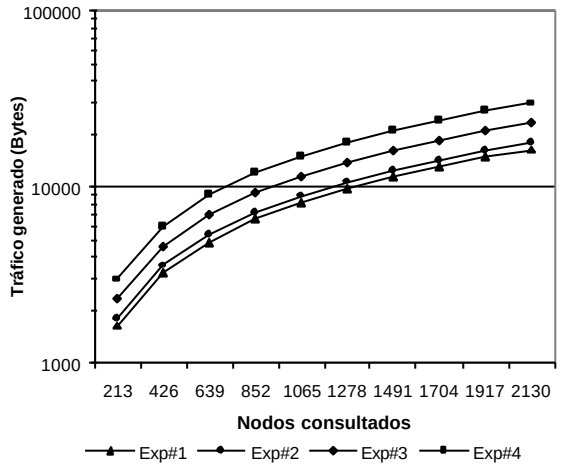

Gráfico 1 – Tráfico generado según la cantidad de nodos que se consultan

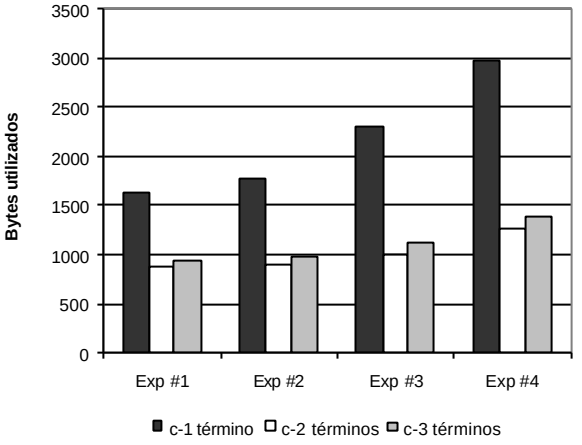

Gráfico 2 – Tráfico generado según la cantidad de términos de la expresión

6 – Análisis comparativo

La funcionalidad del modelo propuesto puede ser directamente comparada con la técnica BFS de la cual se conoce su eficiencia [24]. En general, el protocolo paradigmático que implementa esta técnica es Gnutella, del cual se han planteado limitantes de escalabilidad [16], de acuerdo a una fórmula de tráfico propuesta por Ritter [16]. Sin embargo, otros autores han determinado que esta fórmula no modela el comportamiento de la red Gnutella, ya que se aplica a topologías de árbol con cantidad balanceada de conexiones por nodo. Además, se determinó que la probabilidad de establecer una topología de árbol disminuye drásticamente con el número de saltos [19]. En [20] se presenta un estudio que determina que la cantidad de mensajes reenviados por los nodos de la red es significativamente menor que lo que sugiere una topología de árbol. Aquí se concluye que no hay crecimiento exponencial en Gnutella y que el tráfico crece mucho más lento que lo esperado.

Sobre la base de este estudio se realizó una comparación contra el modelo de índice distribuido propuesto. Se toman directamente los valores obtenidos por el analizador para diferentes configuraciones de conexiones directas, con TTL = 7 (Tabla 3). Véase en [20] que el valor observado es sensiblemente menor al calculado en base al modelo teórico.

El cálculo de tráfico generado se realiza en base a la estimación del tamaño medio de un mensaje de consulta en Gnutella, que es de 83 bytes. Además, hay que considerar que en el cálculo

de tráfico generado no se tienen en cuenta los mensajes descartados por nodos intermedios en la red ya que fue imposible su medición por la metodología utilizada. No obstante, otros estudios [24, 25] plantean que el 30% de los mensajes son descartados por redundantes, por lo tanto el tráfico generado se incrementaría bajo esta consideración.

Para alcanzar similar cantidad de nodos y – por ende – realizar la consulta, el modelo propuesto requiere consultar inicialmente a los nodos índice y luego a los nodos que potencialmente la pueden satisfacer. El tráfico generado para una situación similar a la planteada sobre la configuración del experimento #2 se presenta en la tabla 4.

# de conexiones	# de nodos alcanzados	Tráfico Generado (Bytes)
1,29	224	18592
5,00	631	52373
11,90	1248	103584
17,84	1711	142013
24,29	1912	158696
40,76	2506	207998

Tabla 3 – Cantidad de nodos alcanzados y tráfico generado por el experimento en [20]

# de nodos alcanzados	Tráfico Generado (Bytes)	
	Exp.#2	Exp.#4
213	1778	2978
639	5334	8934
1278	10668	17868
1704	14224	23824
1917	16002	26802
2556	21336	35736

Tabla 4 – Cantidad de nodos alcanzados y tráfico generado para el Exp#2 y el Exp#4

En el gráfico 3 se presentan los resultados comparativos. Sobre esta base se establece que el modelo propuesto aporta una reducción del 90% en el caso del experimento #2 y del 84% en el caso del experimento #4. Estos valores se modifican al 93 y 88% si se tiene en cuenta el incremento del tráfico en un 30% por mensajes redundantes descartados [SCH+30%].

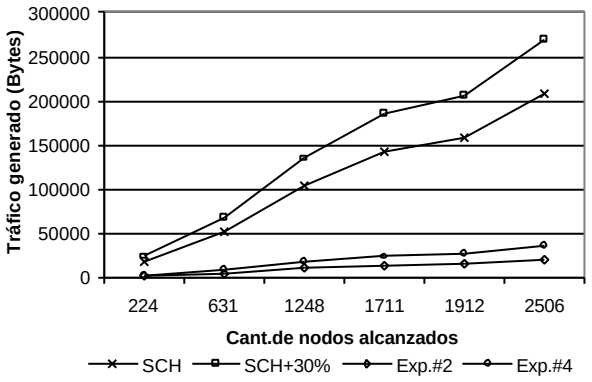

Gráfico 3 – Comparación del tráfico de Gnutella y el modelo propuesto.

6.1 – Determinación de la cantidad de nodos índice a consultar

A partir de la información contenida en la tablas de contactos un nodo que realiza una búsqueda puede obtener diferente cantidad de respuestas de acuerdo a la cantidad de nodos índice que consulte. Por lo tanto, es posible utilizar algún criterio – por ejemplo cantidad de resultados deseados o tráfico máximo generado – para definir un cantidad máxima. Por cada término utilizado en una búsqueda, un nodo tiene la posibilidad de consultar entre 0 y k índices para la letra inicial del término, donde k corresponde al número máximo de entradas que el nodo posee en su tabla de contactos para la letra en cuestión.

El número de resultados retornados por un índice es función de la cantidad de nodos en la red y de la cantidad de documentos distribuidos entre éstos. Por lo tanto, si se desean obtener todos los resultados posibles el tráfico generado crecerá linealmente conforme aumenten los valores para los

parámetros mencionados. Sin embargo, una manera de evitar esta situación es establecer un umbral de corte para la cantidad de nodos índice a consultar, de manera de obtener un costo máximo predecible.

Una política inicial es consultar todos los nodos posibles para una determinada letra. En el gráfico 9 se presentan – para los experimentos 2 y 4 – los resultados promediados de consultas sucesivas por un término a todos los nodos índice posibles. Se muestran los valores totales (sucesivos) y los acumulados para cada experimento. Aquí se supone el caso que el nodo que realiza la consulta posee una tabla de contactos completa. Para cada consulta se tomaron los identificadores de los nodos retornados como respuesta y se computaron solo aquellos que no habían sido retornados en respuestas anteriores, es decir, se calculó cuánta información nueva aportó cada nueva consulta. Esta técnica de búsqueda puede ser comparada con la estrategia Profundización Iterativa [24] utilizada para reducir el tráfico en redes de difusión, en la cual se aumenta el horizonte de alcance de un mensaje conforme se requieran más respuestas.

Nótese – a modo de ejemplo – que consultando a 33 nodos índice se obtiene el 50% de las respuestas posibles, con 53 el 70% y con 81 el 90% (Gráfico 10). Suponiendo la configuración del experimento #2 y el cálculo de tráfico generado según la expresión (1), el costo de realizar estas consultas es de 57, 92 y 141 KB, respectivamente. Este tráfico se encuentra muy por debajo de las estimaciones realizadas para otros sistemas P2P, por ejemplo en [20] para Gnutella.

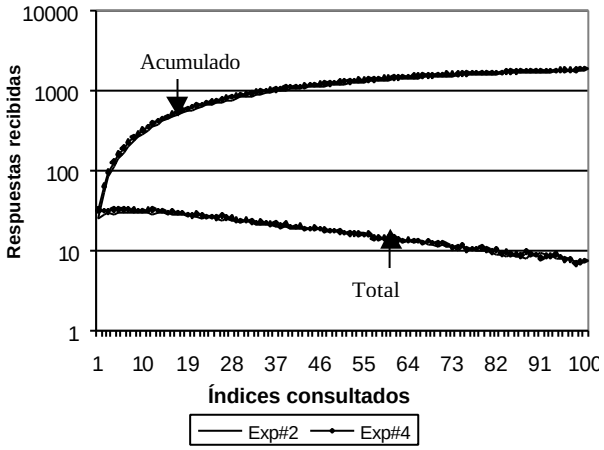

Gráfico 9 – Cantidad de nuevas respuestas (acumulado y totales) por índice

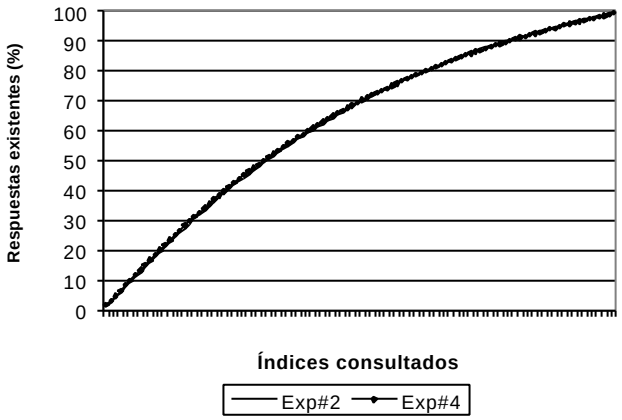

Gráfico 10 – Porcentaje de respuestas recibidas respuesta recibidas por índice

Con estos resultados se determinó una política inicial de selección de los nodos índices a cuáles enviar las consultas y no simplemente seguir el orden “aleatorio” de la tabla de contactos. Se utilizó una simple política consistente en ordenar la tabla de contactos de acuerdo a la cantidad de respuestas que entregó cada nodo índice en consultas anteriores por el mismo término, de manera similar a lo propuesto en [24] para la estrategia BFS Directo. Con esta modificación se ejecutaron nuevamente los experimentos. Los resultados obtenidos fueron moderadamente mejores ya que – en promedio – el incremento de la eficiencia fue del 7%. Esto significa que para obtener la misma performance que la estrategia del caso inicial se necesita consultar menos nodos índice. En el gráfico 11 se muestran – comparativamente – los resultados de las consultas para cada estrategia. Nótese que la mayor diferencia se encuentra entre las consultas a los nodos 17 y 37 donde se obtiene una mejora del 11% respecto que la política original.

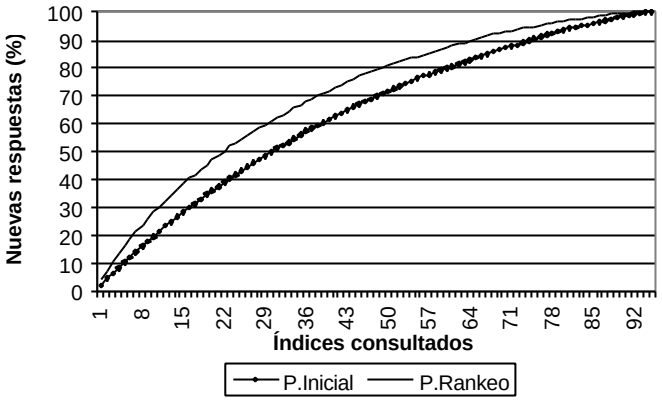

Gráfico 11 – Cantidad de nuevos nodos de acuerdo a cada política de selección
(P.Inicial: Selección Aleatoria,
P.Rankeo: Clasificación por cantidad de respuestas entregadas previamente)

Se presume que este incremento moderado se debe a la uniformidad en la distribución de los índices sobre la red y que – si bien es altamente deseable para balancear las cargas y el volumen del índice en cada nodo – esta situación se irá modificando con el dinamismo de la red y con la entrada y salida de nuevos nodos. En tal caso, la política debería contemplar otros parámetros de rankeo como por ejemplo tiempo de permanencia de un nodo índice en la red, latencia ó una combinación de todas, como se sugiere en [24].

7 – Conclusiones y trabajos futuros

Este trabajo presenta un modelo de índice distribuido sobre una red compañero a compañero que soporta la indexación completa de documentos y permite búsquedas por contenido. Se evaluó su comportamiento mediante una serie de experimentos y del análisis de los resultados se determinó que el modelo propuesto mejora la eficiencia entre un 84% y 93% – para las situaciones propuestas – respecto de sistemas similares a Gnutella.

En cuanto a las búsquedas, el modelo soporta expresiones de consulta con múltiples términos y la utilización palabras incompletas. Esta característica brinda un mecanismo de búsqueda más expresivo y flexible, adecuado para operar sobre el contenido de documentos de texto. Los resultados de las pruebas sugieren que el modelo es una alternativa válida a las Tablas de Dispersión Distribuidas y otras estrategias basadas en el reenvío de consultas para intentar resolver el problema de las búsquedas distribuidas sobre los sistemas P2P actuales.

Los nodos cuentan con la posibilidad de almacenar información de varios nodos índices a los efectos de realizar búsquedas incrementales. Esto permite ampliar la cantidad de nodos a los cuales consultar ante un requerimiento. Para la selección de contactos se determinó que la utilización de una política de rankeo de los nodos índice de acuerdo a la cantidad de respuestas que retornaron anteriormente para la misma consulta mejora moderadamente la eficiencia. El incremento fue – en promedio – del 7%, con un máximo del 11% cuando se consultaron los nodos 17 a 37.

Si bien los resultados iniciales permiten verificar la validez del modelo se prevé realizar pruebas sobre una red con una mayor cantidad de nodos y diferentes configuraciones, variando la cantidad de documentos, su distribución y las capacidades de los nodos de la red. Además, se considera que es necesario continuar la investigación en esta línea a los efectos de mejorar sus

prestaciones y definir parámetros particulares orientados a diferentes aplicaciones posibles.

Otra cuestión importante a estudiar es el comportamiento de diferentes políticas de selección de nodos índice a los efectos de determinar cuál brinda la mayor eficiencia, tanto en términos de tráfico como de tiempos de respuestas. Finalmente, en base a modelos de comportamiento de los usuarios de redes P2P se pretende determinar una fórmula que describa el costo de la consulta como función de la cantidad de nodos, índices y documentos en el sistema.

8 – Referencias

[1] Bawa, M.; Bayardo, R.; Rajagopalan, S.; Shekita, E. "Make it Fresh, Make it Quick – Searching a Network of Personal Web Servers". En 12th International WWW Conference. Mayo, 2003.

[2] Baquero, C. y Lopez, N. "Towards Peer-to-Peer Content Indexing". ACM Operating Systems Review, Vol. 37 No. 4. Octubre, 2003.

[3] Bordignon, F. y Tolosa, G. "Gnutella: Distributed System for Information Storage and Searching. Model Description". JIT – Journal of Internet Technology. Taipei (Taiwan). Vol.2, No.5. 2001.

[4] Bordignon, F. y Tolosa, G. "Redes Compañero a Compañero (P2P): conceptos y tendencias de aplicación. Novática – Revista de la Asociación de Técnicos en Informática, Número 166, pág. 57. Noviembre – Diciembre, 2003.

[5] Chawathe, Y.; Ratnasamy, S.; Breslau, L.; Lanham, N.; Shenker, S. "Making Gnutella-like P2P Systems Scalable". Interl Research Report. IRB-TR-03-014. Abril, 2003.

[6] Dabek, F.; Brunskill, E.; Kaashoek, M.; Karger, D.; Morris, R.; Stoica, I.; Balakrishnan, H. "Building Peer-to-Peer Systems with Chord, a Distributed Lookup Service". En Proceedings of the 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII). Mayo, 2001.

[7] Fasttrack. http://www.fasttrack.nu. 2003.

[8] Garces-Erice, L; Felber, P.; Biersack, E.; Urvoy-Keller, G. y Ross, K. "Data Indexing in Peer-to-Peer DHT Networks" Proceedings of the IEEE 24th International Conference on Distributed Computing Systems (ICDCS), 2004.

[9] Gummadi, K.; Dunn, R.; Saroiu, S.; Gribble, S.; Levy, H.; Zahorjan, J. "Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload". Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP-19), Bolton Landing, NY. October 2003.

[10] Kazaa Media Desktop. http://www.kazaa.com. 2003.

[11] Lv, Q. "Search and Replication in Unstructured Peer-to-Peer Networks". En Proceedings of the 16th ACM International Conference on Supercomputing. 2002.

[12] Napster. http://www.napster.com. 2001.

[13] National Science Foundation. “NSFResearch Awards Abstracts 1990-2003”. http://kdd.ics.uci.edu/summary.data.alphabetical.html.

[14] Ratnasamy, S.; Francis, P.; Handley, M.; Karp, R. ”A Scalable Content-Addressable Network (CAN)”. En Proceedings of ACM SIGCOMM. 2001.

[15] Reynolds, P. y Vahdat, A. “Efficient Peer to Peer Keyword Searching”. Middleware 2003. ACM/IFIP/USENIX International Middleware Conference. Junio, 2003.

[16] Ritter, J. “Why Gnutella Can’t Scale. No http://www.darkridge.com/~jpr5/doc/gnutella.html

[17] Rowstron, A.; Druschel, P. “Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems”. En IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), p329-350. Noviembre, 2001.

[18] Saroiu, S.; Gummadi, K. P.; Gribble, S. “Measuring and analyzing the characteristics of Napster and Gnutella Hosts”. Multimedia Systems 9: 170-184. Springer–Verlag 2003.

[19] Schollmeier, R. y Schollmeier, G. “Why Peer-to-Peer (P2P) Does Scale: An Anlalysis or P2P Traffic Patterns”. En Second International Conference on P2P Computing (P2P’02). Septiembre, 2002.

[20] Schollmeier, R. y Hermann, F. “Topology-Analysis of Pure Peer-to-Peer Networks”. En Proceedings of the Fachtagung Kommunikation in Verteilten Systemen (KiVS 2003). Alemania. Febrero, 2003.

[21] Sen, S; Wang, J. “Analyzing Peer-to-Peer Traffic Across Large Networks”. Proceedings of ACM SIGCOMM Internet Measurement Workshop. Noviembre, 2002.

[22] Tolosa, G., Bordignon, F., Peri, Jorge. “Indice distribuido sobre una red compañero a compañero para búsquedas por contenido de documentos”. Jornadas de la Ciencia y la Tecnología, UNLu. Septiembre de 2003.

[23] Yang, B.; Garcia-Molina, H. “Comparing Hybrid Peer-to-Peer Systems”. En VLDB’01 Conference. 2001.

[24] Yang, B.; Garcia-Molina, H. “Improving Search in Peer-to-Peer Networks”. En 22nd International Conference on Distributed Computing Systems (ICDCS). Julio 2002.

[25] Zeinalipour-Yazti, D. “Information Retrieval in Peer-to-Peer Systems”. Master Thesis. Universidad de California Riverside. Junio, 2003

[26] Zhuang, Z; Liu, Y; Xiao, L.; Ni, L. “Hybrid Periodical Flooding in Unstructured Peer-to-Peer Networks”. International Conference on Parallel Processing. Taiwan. Octubre, 2003.