Diseño de una red P2P optimizada para la privatización de consultas en WSEs

Damià Castellà *, Cristina Romero-Tris *, Alexandre Viejo *,
Jordi Castellà-Roca *, Francesc Solsona † Francesc Giné †
* Dpt. d'Enginyeria Informàtica i Matemàtiques, UNESCO Chair in Data Privacy,
Universitat Rovira i Virgili, Av. Països Catalans 26,
E-43007 Tarragona, Spain
Email: {damia.castella,cristina.romero,alexandre.viejo,jordi.castella}@urv.cat
† Dep. Informàtica i Enginyeria Industrial, University of Lleida,
C/ Jaume II 69, E-25001 Lleida, Spain.
Email: damia.castella@udl.cat, {francesc,sisco}@diei.udl.cat

***Resumen*—Los motores de búsqueda de Internet (Web Search Engines - WSE) guardan las consultas efectuadas por los usuarios (search logs). A partir de los *logs* crean perfiles de usuario que permiten mejorar las respuestas que ofrecen a los usuarios (sugerencias, correcciones, etc..). No obstante, los *logs* pueden contener información confidencial de los usuarios además que pueden permitir su identificación. Para resolver esta amenaza de privacidad se han realizado diferentes propuestas. En este artículo se presenta un entorno P2P híbrido diseñado para agrupar los usuarios con perfiles similares y realizar la protección del perfil de los usuarios mediante un método de privacidad. A diferencia de anteriores propuestas de privacidad que están basadas en entornos P2P, nuestra propuesta está diseñada para soportar un gran número de usuarios y su movilidad.**

I. Introducción

Los motores de búsqueda de Internet (Web Search Engines - WSEs) son una herramienta básica para muchos usuarios. Los WSE almacenan información de las páginas Web, la indexan y responden a las consultas de los usuarios con una lista de resultados que corresponden a los enlaces a las páginas donde se encuentra la información deseada por estos.

Las consultas de los usuarios son una valiosa fuente de información. Los WSE almacenan y agregan las consultas de cada usuario. Con este registro de consultas (*logs*) los WSE crean un perfil de usuario que permite personalizar o mejorar las respuestas que ofrece. Por ejemplo, hay términos que pueden ser ambiguos y el perfil permite escoger el que seguramente quiere el usuario. Si buscamos el término `Mercurio` puede hacer referencia al planeta del sistema solar o al elemento químico de número atómico 80. Si antes hemos hecho una consulta sobre el sistema solar seguramente que estamos interesados en el planeta.

No obstante, aunque los WSEs son de gran ayuda, también pueden ser una amenaza para la privacidad. Alguna de las consultas pueden contener información personal que permita identificar de forma única a los usuarios, por ejemplo si un usuario se busca a si mismo introduciendo su nombre completo, su número de seguridad social, su residencia, ocupación, etc. Además las consultas pueden contener información sensible como problemas de salud, la orientación sexual, política, religión, etc.

Estos son motivos suficientes para exigir una correcta protección de los *logs* de los WSE. Sin embargo, el escándalo de AOL [11] demostró que los usuarios no pueden confiar en la protección ofrecida por los WSE. En este caso, se dieron a conocer públicamente 20 millones de consultas realizadas por 658.000 usuarios que habían sido anonimizadas. No obstante, fue posible identificar a algunos de los usuarios que habían realizado las consultas.

La protección de los perfiles de los usuarios es importante, pero también lo es obtener un buen servicio. Los usuarios pueden ser reticentes a utilizar un sistema que les proporcione mucha privacidad pero que la respuesta sea lenta, o los resultados que les interesen no estén en las primera páginas. La obtención de todas estas propiedades es compleja. Cualquier método de protección será más lento que la consulta directa al WSE, y las respuestas dependerán directamente del nivel de conocimiento de los perfiles de los usuarios (privacidad). Recordemos que las respuestas se generan a partir de los perfiles. Por lo tanto, en general, si un usuario ofusca su perfil de manera muy significativa obtendrá una buena protección de su privacidad, pero también recibirá un peor servicio, y viceversa.

Una opción razonable es la utilización de métodos que ofrezcan un balanceo (*trade-off*) entre privacidad, utilidad del perfil y tiempo de respuesta.

I-A. Contribución y organización del artículo

En este trabajo se presenta una red P2P que agrupa a los usuarios con perfiles similares. Cuando los usuarios quieren enviar una consulta al WSE ejecutan el protocolo presentado en [13] para ofuscar sus perfiles. En [13] cada usuario cuando quiere enviar una consulta decide si la envía directamente al WSE o a uno de sus amigos en función de su historial de consultas prévias y una heurística. Si la envía a un amigo, éste, a su vez, decide si la envía al WSE o a otro de sus amigos. Este procedimiento se va repitiendo hasta que la consulta se envía al WSE. La respuesta del WSE sigue el camino inverso. Se puede suponer que las consultas de los amigos de un usuario son similares a las suyas (mismos intereses), y por lo tanto

no alteraran de forma perjudicial el perfil. No obstante, esta suposición puede no ser cierta, y aceptar una consulta de un amigo de otro amigo puede alterar mucho el perfil. Por este motivo, en este caso se propone la creación de esta red P2P. La agrupación de los perfiles se realiza manteniendo al máximo la privacidad de los usuarios. Además, en [13] no se contemplaba esta agrupación ni tampoco soportaba un gran número de usuarios o la tolerancia a fallos.

En la Sección II se describen brevemente las principales propuestas para proteger la privacidad de los usuarios de los WSE. En la Sección III se presenta la arquitectura propuesta y en la Sección IV las operaciones de la red P2P. La Sección V contiene una breve descripción del proceso que siguen los usuarios al realizar una consulta. Finalmente, las conclusiones y el trabajo futuro se introducen en la Sección VI.

II. Estado del arte

La protección de los usuarios de los WSEs ha sido estudiado, y las soluciones propuestas se pueden clasificar teniendo en cuenta el número de usuarios que participan en el protocolo: protocolos *single-party* y *multi-party*. Los protocolos *single-party* permiten que un usuario proteja su privacidad de forma individual frente los protocolos *multi-party* que requieren que un grupo de usuarios colaboren.

Los protocolos single-party generan consultas falsas[18] o modifican [17] las consultas que son sometidas a los WSEs. Sin embargo, algunas propuestas (p.e. [19], [20]) demuestran que es posible diferenciar las consultas reales de las consultas generadas con una probabilidad de error muy baja (alrededor de $0{,}02\,\%$).

Otra opción es usar un canal anónimo como por ejemplo *Tor* [15]. No obstante, en este caso el proceso de someter una consulta es 25 veces más lento que una conexión directa [16]. Además el usuario no tiene perfil de manera que obtiene un peor servicio del WSE (ver [21] para un estudio más detallado).

Por otro lado, los protocolos *multi-party* no están afectados por los errores de clasificación, y generalmente son más rápidos que los esquemas basados en canales anónimos. En estos protocolos, los usuarios ofuscan su perfil sometiendo las consultas de otros usuarios. La obtención de las consultas de otros usuarios se realiza mediante la creación de grupos de usuarios, estos grupos pueden ser creados de forma dinámica cuando un usuario quiere enviar una consulta al WSE [21], o si el grupo ya está creado [13], [14].

Los protocolos con grupos dinámicos utilizan un servidor para agrupar a los usuarios. Este nodo puede ser un cuello de botella, o sufrir ataques de denegación de servicio. En los protocolos con grupos estáticos no sufren este tipo de ataques pero existe la posibilidad que los usuarios puedan crear un perfil de los miembros del grupo. Por este motivo es necesario un protocolo para protegerse del resto de usuarios.

Considerando las ventajas y desventajas de las soluciones con grupos dinámicos o estáticos en este trabajo se propone una solución híbrida. El sistema presentado en este artículo clasifica a los usuarios de los WSE en grupos que tienen perfiles similares. Este grupo puede ser diferente en cada sesión del usuario.

En trabajos previos [7], [8] D. Castella et al. propuso el diseño de una arquitectura P2P de computo distribuido, llamada DisCoP, optimizada para la búsqueda de recursos computacionales, como la capacidad de CPU, memoria y/o disco, de una manera completa, distribuida, escalable y tolerante a fallos. DisCoP clasifica y ordena los recursos computacionales de todos los nodos en diferentes *mercados* para soportar búsquedas de consultas basadas en *rango*, *multi-atributo* y *aproximadas*.

La propuesta aprovecha las capacidades *multi-atributo* de la red para clasificar a los usuarios según las múltiples preferencias de los perfiles de los usuarios.

III. Red P2P propuesta

El objetivo principal de nuestra propuesta es construir una red P2P híbrida para clasificar y agrupar los perfiles de usuario y que al mismo tiempo contemple un alto grado de escalabilidad, rendimiento de búsqueda y muy especialmente, un servicio de privacidad eficiente para enmascarar las consultas de los usuarios. En esta sección se explica en detalle el diseño de la arquitectura que compone la red Peer-to-Peer.
A modo de ejemplo, en la Figura 1 se muestra el *outline* principal de la arquitectura P2P, representando los diferentes niveles o capas de interconexión que componen el sistema. En la Figura 1 se puede ver como la arquitectura P2P esta compuesta por tres niveles jerárquicos (suboverlays): *Hilbert Space Filling Curve (SFC)* como primer nivel, *de Bruijn* como segundo y los *Clusters de Perfil* (*CP*) como tercer nivel.

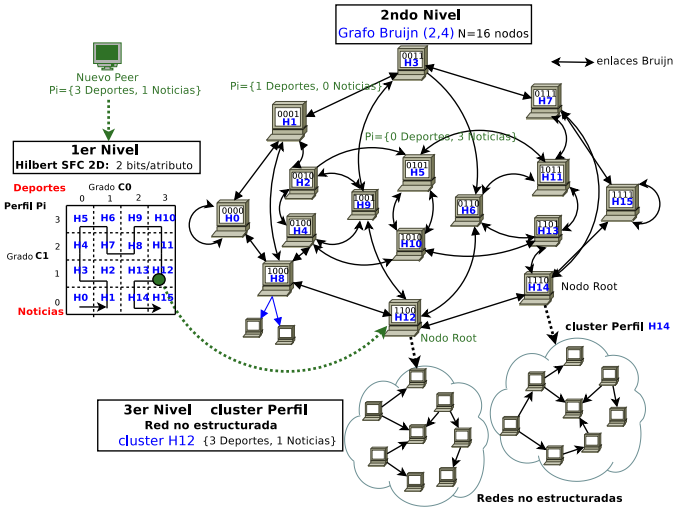

Figura 1. Propuesta de red P2P actual.

El objetivo de esta topología es distribuir los múltiples perfiles de usuario que existen por la red estructurada de Bruijn y agrupar los usuarios con el mismo perfil en las redes no estructuradas del tercer nivel llamadas *Clusters de Perfil* (*CP*).

III-A. *El vector Perfil*

Un peer cuando entra en la red está categorizado con los siguientes parámetros (P_i, L_i). P_i es un vector que representa el *Perfil de Usuario* y L_i un índice que indica su localidad geográfica. Teniendo en cuenta la localidad se mejorará el

rendimiento de las consultas consiguiendo reducir el RTT (Round Trip Time) en las comunicaciones entre peers. Este artículo solo se centrará en el diseño de la topología P2P y sus algoritmos. La localidad L_i queda fuera de esta propuesta y se considera como trabajo futuro.
El modelo del Perfil del Usuario P_i, es un vector que contiene k-coordenadas, $\{C_1, .., C_i, ..., C_k\}$, donde cada componente C_i indica el grado de preferencia de una determinada categoría en el i-esimo atributo. Las k categorías del usuario C_i, como por ejemplo *noticias, deportes, turismo, etc.* son seleccionadas de acuerdo con el principal interés de los usuarios.

III-B. Topologia Peer-to-Peer previa

Tal y como se ha mencionado en la Sec. II, una arquitectura Peer-to-Peer orientada a Computo Distribuido llamada *DisCoP* [7], [8] fue diseñada. La arquitectura de *DisCoP* esta formada por tres niveles jerárquicos (*Hilbert SFC*, *Bruijn* y las topologías *N-ari Tree* también llamadas *Mercados*). El objetivo de la topología propuesta es ordenar y agrupar los múltiples recursos computacionales compartidos por los nodos en diferentes mercados y poder localizar diferentes tipos de recursos en el menor número de saltos. En cambio, en nuestra propuesta el objetivo es obtener ventaja del diseño de DisCoP para clasificar preferencias de búsqueda similar de los usuarios en clusters y reemplazar los mercados por redes P2P no estructuradas llamadas *Clusters Perfil* (CP). El propósito de la selección de este tipo de red es porque consiguen preservar el anonimato de los usuarios y se puede acoplar el protocolo P2P de privatización ya diseñado en [13], [14] sin requerir modificaciones.

III-B1. Primer nivel: Función de Hilbert SFC: La curva multi-dimensional Hilbert SFC es utilizada como una función Hash para mapear un espacio multi-dimensional de datos dentro de una dimensión (1D). Un SFC es un hilo que visita todos los puntos de un espacio multi-dimensional una sola vez. En nuestro caso, la función Hilbert SFC [1] es utilizada para mapear el vector Perfil Pi de un usuario para transformarlo en un único identificador Hi que es utilizado como dirección para acceder a un nodo en el nivel de Bruijn.

La Figura 2 muestra un ejemplo de la transformación de un Perfil de usuario. Tal como se puede apreciar, el vector Perfil P_i esta compuesto de dos categorías ($P_i = \{C_1, C_2\}$), *Deportes* e *Historia* en este caso. Para hacer la transformación del vector, cada categoría C_i del vector se transforma a un eje del espacio multi-dimensional de Hilbert respectivamente y el rango de valores que contiene el grado de preferencia.

El objetivo de usar esta curva como función de mapping es para colocar los perfiles similares en posiciones cercanas en la red de Bruijn. La función de Hilbert SFC es la curva dentro una familia existente de SFC, que preserva más la propiedad de *clustering* colocando los atributos multi-dimensionales en posiciones cercanas dentro del espacio unidimensional.
De este modo, cuando un Cluster Perfil llega a su limite máximo de nodos, se podrá desplazar nodos de un perfil a otros clusters similares en pocos saltos, reduciendo así el coste de comunicación. Esta política de reemplazo se queda fuera de esta propuesta y se considera como trabajo futuro.

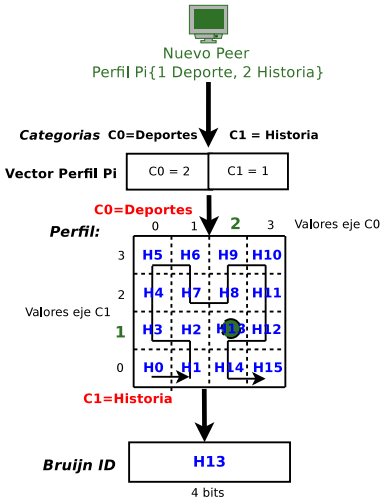

Figura 2. Hilbert SFC de 2 dimensiones y 2 bits/dimensión.

III-B2. Segundo nivel: red de Bruijn: El segundo nivel de la arquitectura está compuesto por una red de topología grafo Bruijn que previamente enlazaba mercados de recursos computacionales. Bruijn tiene óptimas propiedades que pueden ser trasladas a una red distribuida P2P, su diámetro óptimo es de orden $\log_e(N)$, construido con un grado e constante de enlaces. También su morfología es simétrica y multi-direccional lo cual permite multitud de caminos de búsqueda entre dos nodos y así tener una buena tolerancia a fallos. El tipo de grafo Bruijn usado es un grafo directo con los ejes de entrada y salida conectados a cada nodo. Además, cada nodo tiene una única clave fija Hi suministrada por el primer nivel.

No obstante, nuestro propósito es reemplazar la interconexión de Mercados de cómputo por entidades llamadas *Cluster Perfil* (*CP*). Estas están diseñadas como redes no estructuradas P2P, usadas para clusterizar los peers de un mismo perfil y de este modo proteger el perfil de los usuarios malintencionados. (ver Sec. III-B3).

En el grafo Bruijn, el número máximo de nodos es $N = e^D$, donde D es el diámetro (máxima distancia entre cualquiera de los dos nodos), y e el número de conexiones que debe mantener cada nodo.

En la Figura 3 (segundo nivel) se muestra una red clásica de Bruijn para $e = 2$ y $N = 8$. Para obtener el grafo, simplemente para cada clave Hi de un nodo se desplaza un dígito a la izquierda y después se suma el índice del enlace correspondiente. En el caso de la clave $H4$ se obtienen los enlaces dirigidos a $H0$ y $H1$ que resultan de la operación $(H4 \equiv (001_2) << 1) +$ Índices Enlaces $\{0, 1\} = \{H0, H1\}$.

Hay que destacar que para la creación de la red de Bruijn se diseñaron dos mejoras y que están detalladas en los artículos [7], [8]. La primera mejora consiste en *la virtualización de nodos*, técnica usada para crear nodos virtuales dentro de los nodos existentes cuando la red Bruijn no este completa. La segunda mejora es la utilización de *los enlaces sucesores*

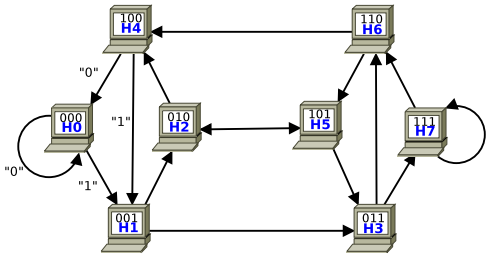

Figura 3. Grafo de Bruijn con los parámetros $e = 2$ y $N = 8$.

Hilbert a cada nodo consiguiendo así mantener la localidad de los recursos multi-atributo similares en posiciones cercanas a la red Bruijn. En nuestro caso, se obtienen los mismos resultados si aplicamos el vector Pi del perfil de usuario. Finalmente, hay que comentar que los algoritmos de búsqueda empleados para la localización de nodos en la red Bruijn estan descritos también en [7], [8].

III-B3. Cluster de Perfil (CP): Una red CP esta diseñada como una red distribuida no estructurada. Las redes no estructuradas no siguen ningún mecanismo para ordenar o dispersar los nodos dentro la red, como por ejemplo en cambio son las *DHTs* (*Distributed Hash Tables*) [5], [4], [2], tablas de dispersión distribuidas usadas en redes estructuradas. Además proporcionan más resistencia ante fallos de nodos y ataques. No obstante, los mecanismos de búsqueda no están bien escalados porque los nodos no están ordenados por ninguna característica. Sin embargo, para nuestro caso los mecanismos de anonimato y privacidad están mejor soportados.

La razón de su diseño es poder agrupar peers con un perfil Pi similar. Mediante esta agrupación se consigue que los nodos / usuarios que utilizan el protocolo obtenga un perfil ofuscado pero que mantenga su usabilidad. Es decir, un buen trade-off entre usabilidad y privacidad.

IV. Mecanismos de la red P2P

La siguiente Sec. IV describe los métodos y políticas utilizadas para construir, mantener y desconectar los peers del sistema.

IV-A. Construcción del sistema

Como muchas redes P2P, estas se construyen gradualmente de modo ad-hoc a medida que van entrando los nodos. Los pasos para insertar un nuevo peer en el sistema se describen a continuación:

IV-A1. Inserción en la red Bruijn: Inicialmente un nuevo peer H_i tiene una configuración de entrada con los siguientes parámetros (Pi, L_i) necesarios para entrar en la red. La inserción de un nuevo peer en la red Bruijn puede realizarse de dos maneras: Si existe el *CP* asociado al perfil Pi del nuevo peer, solo deberá localizar su CP asociado al perfil mediante uno de los métodos de búsqueda de Bruijn y formar parte de esta red. En el caso de que no exista, se procederá a la creación de un nuevo CP y se le asignará al nuevo peer como nodo *root* de la red. A continuación se describen los pasos necesarios para ser insertado:

1. El usuario selecciona el grado de preferencia de cada categoría C_i del *ODP (Open Directory Project)* y lo codifica con una longitud fija de c bits. De este modo, construye el vector Perfil Pi. Se ha utilizado el ODP porqué es una clasificación ampliamente aceptada de las diferentes categorías que existen de páginas Web [10].
2. El peer Hi conecta al nodo *Bootstrap* para obtener las direcciones físicas de algunos nodos *root* existentes en la red Bruijn. El nodo *Bootstrap* es usado como puerta de entrada a la red P2P y así permitir que cualquier usuario pueda localizar la red. Este nodo esta aislado del resto y es administrado por creadores y/o voluntarios de la red P2P.
3. El peer solicita a un nodo *root* conocido la petición de búsqueda de su CP Hi.
4. Si existe entonces se conecta y finaliza la inserción. Sino se le asigna como nodo *root* del Perfil Hi.
5. El peer Hi mapea el vector Pi con la función Hilbert SFC y obtiene el ID Hi del *CP* a localizar.
6. El peer Hi procede a actualizar sus enlaces de red. Por lo tanto, localiza y envía e notificaciones a sus vecinos derechos correspondientes de Bruijn con las claves ($\{Hi << 1+0, Hi << 1+1, ..., Hi << 1+e-1\}$).
7. Del mismo modo, el peer $Hi+1$ envía otras e notificaciones a sus vecinos izquierdos ($\{Hi >> 1+0, Hi >> 1+1, ..., Hi >> 1+e-1\}$).
8. Finalmente, el peer Hi localiza y enlaza con los peers vecinos sucesor $Hi+1$ y predecesor $Hi-1$.

IV-A2. Inserción en el CP: Los CP son la tercera capa de la arquitectura y está formada por una red no estructurada. Las condiciones para construir esta red son:

1. Cada nodo *root* de la capa Bruijn mantiene una *cola de entrada* guardando las direcciones físicas de los peers recién insertados. Esta cola aplica una política *FIFO* y se actualiza cada vez que entra un peer. Esta lista es ofrecida a cada peer para que se pueda incorporar rápidamente a la red.
2. Cada nodo puede estar conectado a cualquier peer de la red. Este conjunto de peers es escogido aleatoriamente mediante la *cola de entrada* proporcionada por el *root* peer.
3. Cuando un peer entra en el sistema tiene que conectarse como mínimo a σ nodos aleatorios en la red. La red tiene que tener grado σ como mínimo por propósitos de privacidad. El rango de valores del grado σ puede variar entre entre $[\sigma, \sigma + \frac{1}{\lambda}]$. Es necesario mantener este número mínimo de enlaces para acoplar el mecanismo de privacidad citado en [13], ya que son requeridos para ocultar las consultas. En la Sec. V se comenta el mecanismo.
4. Los k nodos aleatorios son seleccionados aplicando un método de búsqueda no informada, [2], [3], como por ejemplo el *Random Walk*. *Random Walk* es un de los protocolos de búsqueda ampliamente utilizado en este tipo de redes. Esta estrategia se basa en enviar

un consulta y a cada salto seleccionar aleatoriamente un nodo vecino. Para acotar el número de pasos de la consulta se le añade un contador TTL (*Time To Live*) al mensaje y se decrementa en cada paso. Una vez alcance cero, la consulta deja de enviarse y el nodo final es el candidato.

IV-B. Mantenimiento de la conectividad

Para mantener la conectividad, la red P2P requiere un intercambio de mensajes *Alive* entre nodos vecinos, en cada periodo T con la intención de examinar si existen los nodos vecinos. Es importante examinar periódicamente los enlaces para no perder la conectividad. El valor del periodo T debe estar balanceado para que sea suficientemente pequeño y detectar rápidamente un caso de desconexión y a la vez suficientemente grande para no congestionar la red en el intercambio de mensajes.
Después de un periodo T, cuando la notificación de un mensaje *Alive* no ha llegado al vecino derecho, el peer actual procederá a llamar al mecanismo de salida de peers descrito en la Sec. IV-C teniendo en cuenta el nivel de la red donde se produjo.

IV-C. Salida de peers

Cuando un peer desconecta del sistema, sea voluntaria o involuntariamente, es necesario reestructurar el sistema para mantener su conectividad y funcionalidad. Si la red quedase desconectada entonces su rendimiento bajaría, aumentando el número de saltos para la localización de un nodo y la posibilidad de existir grupos de peers aislados del resto. A continuación se describe el mecanismo de salida de peers de los dos niveles:

IV-C1. Salida en la red Bruijn:

1. El peer Hi se desconecta del sistema.
2. Después de T unidades de tiempo, el peer sucesor Hilbert $Hi+1$ detecta el enlace roto.
3. El peer $Hi+1$ se hace sucesor de Hi virtualizando el nodo Hi, añade su clave a su rango y guarda sus enlaces.
4. Para ello, localiza y envía e notificaciones a los vecinos derechos de Hi para reemplazar la dirección y apuntar a $Hi+1$.
5. Del mismo modo, envía otras e notificaciones a sus vecinos izquierdos.
6. Finalmente, notifica el nuevo cambio al peer predecesor $Hi-1$.

IV-C2. Salida en los CP: Cada vez que un peer desconecta de la red de Perfiles los σ vecinos que detectan los enlaces rotos deberán conectarse a otro nodo. Para encontrar nodos aleatorios dentro de la propia red existen dos posibles maneras:

- Seleccionando otros nodos aleatorios de la cola *FIFO* subministrada al entrar por el *superpeer*. Se escogen los nodos aún no visitados y disponibles y una vez que se finalice la lista se aplica el segundo método.
- Aplicando el algoritmo de búsqueda no informada *Random Walk*. De este modo, se envía una consulta y se redirige sucesivamente a un número limitado de vecinos con el fin de encontrar un nodo aleatorio.

V. Método de Privacidad

Para proteger la privacidad de los usuarios de los mecanismos de monitorización de perfil de los WSEs se diseño un protocolo para redes P2P en anteriores trabajos [13], [14]. El esquema propuesto fue diseñado utilizando redes sociales como P2P para obtener un perfil de usuario distorsionado al motor de búsqueda web. El método considera la presencia de usuarios que no siguen el protocolo (p.e. adversarios) de tal modo que evalúa la privacidad de los usuarios. En el protocolo propuesto las consultas se someten de forma estandard al motor de búsqueda con lo que no requiere ningún cambio por parte del servidor. Es decir, no se requiere ninguna colaboración del servidor con los usuarios. No obstante, la red P2P que se propuso no estaba diseñada para soportar un gran volumen de usuarios, ni tampoco permitía la agrupación de estos a partir de su perfil.

Este protocolo mejora las soluciones existentes en términos de tiempo de respuesta. Además, los perfiles distorsionados aún permiten a los usuarios tener un servicio adecuado de los motores de búsqueda web. Para ver una descripción mas detallada de la implementación ver [13], [14].

VI. Conclusiones

Los actuales servidores de búsqueda de Internet crean *logs* de los resultados de búsqueda de los usuarios para construir un perfil personalizado. Esta información amenaza la privacidad del usuario. En este trabajo se ha propuesto una arquitectura P2P híbrida, formada por tres niveles jerárquicos, Hilbert, Bruijn y Clusters de Perfil con la intención de agrupar los peers con perfiles similares y adaptar el protocolo de privatización ya implementado fácilmente a la red. Además se han descrito los mecanismos para crear el sistema, el mantenimiento de la conectividad y la salida de peers.
Como trabajo futuro se propone simular la red P2P propuesta, adaptar el protocolo P2P de privatización de búsquedas y evaluar su rendimiento en términos de dinamismo de entrada y salida de peers, envío de consultas y la laténcia producida. Este estudio permitiría valorar su impacto en escenarios reales. Finalmente, también se estudiará cuales serían los niveles aceptables de privacidad y tiempo de respuesta en distintos escenarios.

Referencias

[1] Mokbel, Mohamed F. and Aref, Walid G. and Kamel, Ibrahim, "Analysis of Multi-Dimensional Space-Filling Curves", *Geoinformatica*, Vol. 7, 179–209, September 2003.

[2] E. Meshkova, J. Riihijrvi, M. Petrova, and P. Mahonen, "A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks", *Computing Networks, 52, 11, pp. 2097–2128, 2008.*

[3] K. Lua, J. Crowcroft, M. Pias, R. Sharma and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes", Communications Surveys & Tutorials*, S. 72–93, 2005.*

[4] Sarmady, Siamak "A Survey on Peer-to-Peer and DHT", 2007.

[5] Y-J. Joung, L-W. Yang and C-T. Fang, "Keyword search in DHT-based peer-to-peer networks", IEEE Journal on Selected Areas in Communications*, 25, pp. 46–61, 2007.*

[6] G. Sakaryan, M. Wulff and H. Unger, "Search Methods in P2P Networks: A Survey", pp. 59–68, 2006.

[7] D. Castella, H. Blanco, F. Gine, and F. Solsona, "Combining Hilbert SFC and bruijn graphs for searching computing markets in a P2P system", In Proceedings of the 16th international Euro-Par conference on Parallel processing: Part I (EuroPar'10), *Springer-Verlag, Berlin, Heidelberg, pp. 471–483, 2010.*

[8] D. Castella, F. Gine, F. Solsona, J.L. Lerida, "A Resilient Architecture Oriented to P2P Computing", IEEE 10th International Symposium on Network Computing and Applications*, pp. 41–50, 2011.*

[9] G. Conti, E. Sobiesk, "An honest man has nothing to fear: user perceptions on web-based information disclosure", Proceedings of the 3rd symposium on usable privacy and security*, pp. 112–121, 2007.*

[10] Open Directory Project, 2011. `http://www.dmoz.org/`.

[11] M. Barbaro, T. Zeller, "A Face is Exposed for AOL Searcher No. 4417749", New York Times*, August 2006.*

[12] K. Hafner, M. Richtel, "Google Resists U.S. Subpoena of Search Data", New York Times*, January 2006.*

[13] A. Viejo, J. Castellà-Roca, "Using social networks to distort users' profiles generated by web search engines", Computer Networks*, vol. 54, no. 9, pp. 1343–1357, 2010.*

[14] A. Erola, J. Castellà-Roca, A. Viejo, J.M. Mateo-Sanz, "Exploiting social networks to provide privacy in personalized web search." In The Journal of Systems and Software*, vol. 84, no. 10, pp. 1734(12), 2011.*

[15] R. Dingledine, N. Mathewson, P. Syverson, "Tor: the second-generation onion router", Proceedings of the 13th conference on USENIX Security Symposium*, pp. 21–21, 2004.*

[16] F. Saint-Jean, A. Johnson, D. Boneh, J. Feigenbaum, "Private Web Search" Proceedings of the 2007 ACM workshop on Privacy in electronic society – WPES'07*, pp. 84–90, 2007.*

*[17] J. Domingo-Ferrer, A. Solanas, J. Castellà-Roca, "*h(k)*-private information retrieval from privacy-uncooperative queryable databases",* Journal of Online Information Review*, vol. 33, no. 4, pp. 1468–4527, 2009.*

[18] TrackMeNot, 2011. `http://mrl.nyu.edu/dhowe/trackmenot`.

[19] R. Chow, P. Golle, "Faking contextual data for fun, profit, and privacy", Proceedings of the 8th ACM workshop on Privacy in the electronic society – WPES'09*, pp. 105–108, 2009.*

[20] S. T. Peddinti, N. Saxena, "On the privacy of web search based on query obfuscation: a case study of TrackMeNot", Proceedings of the 10th international conference on Privacy enhancing technologies – PETS'10*, pp. 19–37, 2010.*

[21] J. Castellà-Roca, A. Viejo, J. Herrera-Joancomartí, "Preserving user's privacy in web search engines", Computer Communications *vol. 32, no. 13-14, pp. 1541–1551, 2009.*