

Diseño de un sistema de control de acceso estructurado en un sistema p2p usando pastry

E. Estrada¹, J. Wilches²
elsy_estrada0191@hotmail.com¹, ing.jorgewilches@gmail.com²

Resumen—Este documento detalla la propuesta de un sistema peer-to-peer usando pastry con past además este describirá como se realizó el diseño de una red p2p que nos ayude a identificar que nodos pueden acceder a un archivo por medio de políticas de lectura implantando el sistema pastry con past

Palabras clave—Redes, p2p, pastry, past, nodos.

Abstract— This document details the proposal of a peer-to-peer system using pastry with the past also described the design of a P2P network to help us identify which nodes can access a file through implementing policies reading system was conducted pastry with past

Keywords—Networks, p2p, pastry, past, nodes

I. INTRODUCCIÓN

En una red P2P como es la que Pastry crea, la información se almacena en tablas de hash. Por esto denominamos a estos sistemas DHT (Distributed Hash Table). Una tabla de Hash consiste en un array de elementos donde cada uno posee su correspondiente clave. De esta forma cuando añadimos un elemento a la tabla tenemos que calcular un índice a partir de la clave, intentando que sea lo más distinto posible al resto para mejorar la eficiencia en búsquedas futuras.

En este proyecto se utilizara FreePastry el cual esta implementado en Java, por lo que todas las modificaciones y aportaciones realizadas sobre su código se han hecho con este lenguaje. FreePastry es una API (Application Programing Interface) de Java concebida para crear aplicaciones en comunicación P2P, de forma totalmente descentralizada y distribuida. Como hemos explicado en conceptos básicos, se trata de una red P2P – DHT, aunque presenta algunas Características propias que hacen que no se trate de una red Peer - to - Peer pura. Además utilizaremos past que es una aplicación de almacenamiento global peer-to-peer persistente, que nos permite crear la tabla distribuida, añadiendo los nodos con su información, lista nodos, e introduce o recoge los datos.

II. PASTRY

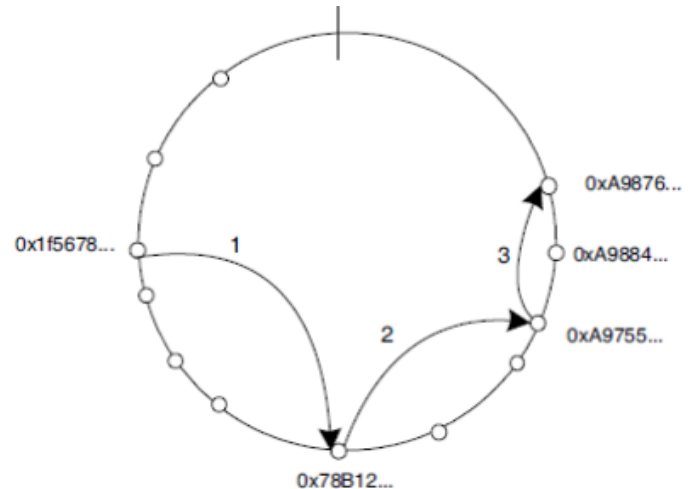


Figura 1. Rutas Pastry mensajes a nodos cuyos

nodeIds son progresivamente más cerca al mensaje clave.

Los nodos de pastry se organizan alrededor de un espacio circular de Identificación. Cada nodo de la red de Pastry se le asigna 128-bit este que es un identificador único que se genera típicamente desde el hash criptográfico, por ejemplo, dirección IP y los usuarios de nombres. Por ejemplo Utilizando el algoritmo de hash seguro (SHA-1) [6], una cadena como "la informática" se generar el código hash "0bbb843c75b8cb93ceb9d5594e208668484448ee".

Pastry tiene la capacidad para enrutar los mensajes entre los nodos cuando se les da una clave de mensaje. El mensaje se dirige al nodo cuyo nodeid se encuentre numéricamente más cercano a la clave del mensaje. Algoritmo de encaminamiento pastry es eficiente, escalable y robusta. La organización de los nodos alrededor del anillo de Pastry es aleatoria. Esto es debido a la forma en que los nodos son asignado a su id. Un algoritmo de cifrado se emplea para el propósito de generar los códigos únicos; este algoritmo

Comúnmente llamado hash que es SHA-1. SHA-1 es irreversible, colisión-resistente, y tiene una buena avalancha efecto.

El efecto de la avalancha de algoritmos hash significa que, dadas dos cadenas muy similares, dos muy diferentes y no numéricamente cercanos códigos hash se produce. Pastry utiliza esta característica como una manera de conseguir una carga equilibrada dentro de la red colocando de forma aleatoria los nodos alrededor de la red.

III. DESARROLLO

A. Términos aplicados:

PAST es un sistema de archivos distribuido en capas en la parte superior de Pastry. Un archivo se almacena en el sistema calculando el hash de su nombre de archivo. A continuación, las rutas de Pastry los contenidos del archivo al nodo en el espacio de claves circular más próximo al hash obtenido de nombre de archivo. Este nodo enviará entonces copias del expediente a los k nodos más cercanos a la clave actual, la mayoría de los cuales es probable que sean nodos hojas de este nodo y por lo tanto accesibles directamente. La recuperación de datos se lleva a cabo refrito el nombre del archivo y el enrutamiento de una solicitud de los datos sobre la masa al lugar que le corresponde en el espacio de claves. El pedido puede realizarse por cualquiera de los k nodos que tienen copias de los datos. Esto logra redundancia de datos y reparto de la carga. Puesto que los nodos adyacentes en el espacio de claves están geográficamente diversa las posibilidades de que todos los k de ellos se van fuera de línea al mismo tiempo es muy pequeño. Más importante, ya que el protocolo de enrutamiento Pastry busca minimizar la distancia recorrida, el nodo más cercano a la máquina que hizo la solicitud (de acuerdo con la métrica) es probable que sea el que responde con los datos.

PASTRY es una superposición de enrutamiento de red y para la aplicación de una tabla de hash distribuida (DHT) similar a Chord. Los pares clave-valor se almacenan en una redundante peer-to-peer red de conectados a Internet hosts. El protocolo se bootstrapped facilitándole el dirección IP de un peer ya en la red y a partir de entonces a través de la tabla de enrutamiento que es dinámicamente construido y reparado. Debido a su naturaleza descentralizada, redundante y no hay punto único de fallo y cualquier nodo solo puede salir de la red en cualquier momento sin previo aviso y con poca o ninguna posibilidad de pérdida de datos. El protocolo también es capaz de usar una métrica de enrutamiento proporcionados por un programa externo, tales como mesa de ping o traceroute, para determinar las mejores rutas para almacenar en su tabla de enrutamiento.

REDES P2P Es una red de computadoras en la que todos o algunos aspectos funcionan sin clientes ni servidores fijos, sino una serie de nodos que se comportan como iguales entre sí. Es decir, actúan simultáneamente como clientes y servidores respecto a los demás nodos de la red. Las redes P2P permiten el intercambio directo de información, en cualquier formato, entre los ordenadores interconectados.

Normalmente este tipo de redes se implementan como redes superpuestas construidas en la capa de aplicación de redes públicas como Internet.

Las redes peer-to-peer aprovechan, administran y optimizan el uso del ancho de banda de los demás usuarios de la red por medio de la conectividad entre los mismos, y obtienen así más rendimiento en las conexiones y transferencias que con algunos métodos centralizados convencionales, donde una cantidad relativamente pequeña de servidores provee el total del ancho de banda y recursos compartidos para un servicio o aplicación.

IV. CARACTERISTICAS:

Seis características deseables de las redes P2P:

- Escalabilidad. Las redes P2P tienen un alcance mundial con cientos de millones de usuarios potenciales. En general, lo deseable es que cuantos más nodos estén conectados a una red P2P, mejor será su funcionamiento. Así, cuando los nodos llegan y comparten sus propios recursos, los recursos totales del sistema aumentan. Esto es diferente en una arquitectura del modo servidor-cliente con un sistema fijo de servidores, en los cuales la adición de clientes podría significar una transferencia de datos más lenta para todos los usuarios. Algunos autores advierten que, si proliferan mucho este tipo de redes, cliente-servidor, podrían llegar a su fin, ya que a cada una de estas redes se conectarán muy pocos usuarios.
- Robustez. La naturaleza distribuida de las redes peer-to-peer también incrementa la robustez en caso de haber fallos en la réplica excesiva de los datos hacia múltiples destinos, y — en sistemas P2P puros— permitiendo a los peers encontrar la información sin hacer peticiones a ningún servidor centralizado de indexado. En el último caso, no hay ningún punto singular de falla en el sistema.
- Descentralización. Estas redes por definición son descentralizadas y todos los nodos son iguales. No existen nodos con funciones especiales, y por tanto ningún nodo es imprescindible para el funcionamiento de la red. En realidad, algunas redes comúnmente llamadas P2P no cumplen esta característica, como Napster, eDonkey o BitTorrent.
- Distribución de costes entre los usuarios. Se comparten o donan recursos a cambio de recursos. Según la aplicación de la

red, los recursos pueden ser archivos, ancho de banda, ciclos de proceso o almacenamiento de disco.

- Anonimato. Es deseable que en estas redes quede anónimo el autor de un contenido, el editor, el lector, el servidor que lo alberga y la petición para encontrarlo, siempre que así lo necesiten los usuarios. Muchas veces el derecho al anonimato y los derechos de autor son incompatibles entre sí, y la industria propone mecanismos como el DRM para limitar ambos.

- Seguridad. Es una de las características deseables de las redes P2P menos implementada. Los objetivos de un P2P seguro serían identificar y evitar los nodos maliciosos, evitar el contenido infectado, evitar el espionaje de las comunicaciones entre nodos, creación de grupos seguros de nodos dentro de la red, protección de los recursos de la red... La mayor parte de los nodos aún están bajo investigación, pero los mecanismos más prometedores son: cifrado multiclave, cajas de arena, gestión de derechos de autor (la industria define qué puede hacer el usuario; por ejemplo, la segunda vez que se oye la canción se apaga), reputación (permitir acceso sólo a los conocidos), comunicaciones seguras, comentarios sobre los ficheros, etc.

- Scribe Es un organismo descentralizado de publicación / suscripción sistema que utiliza pastry de su gestión ruta subyacente y la búsqueda de acogida. Los usuarios pueden crear temas para que otros usuarios pueden suscribirse. Una vez que el tema ha sido creado, el dueño del tema pueden publicar nuevas entradas en el tema que se distribuirá en una multidifusión árbol para todos los nodos SCRIBE que se han suscrito a este tema. El sistema funciona mediante el cálculo del valor hash del nombre del tema concatenado con el nombre del usuario al que pertenece el tema. Este hash se utiliza entonces como una clave de pastry, al editor entonces enruta paquetes al nodo más cercano a la clave utilizando el protocolo de encaminamiento pastry para crear el nodo raíz del tema en dicho nodo. Entonces la gente suscribirse al tema mediante el cálculo de la clave del tema y el nombre del editor y luego usando pastry para dirigir un mensaje de suscripción con el tema hacia el nodo raíz. Cuando el nodo de raíz recibe el mensaje de suscripción de otro nodo se añade el ID de nodo a la lista de los niños y comienza a actuar como un promotor del tema. [4]

- Generación de nodosids Generación de nodeIds una tarjeta inteligente proporciona la NodeID para un nodo PAST asociado. El ID de nodo se basa en un hash criptográfica de clave pública de la tarjeta inteligente. Esta asignación de nodeIds probabilísticamente asegura una cobertura uniforme del espacio de nodeIds y la diversidad de nodos con nodeIds adyacentes, en términos de localización geográfica, red apego,

la propiedad, el imperio de la ley, etc. Además, los nodos Puede verificar la autenticidad de los demás nodeIds.

- Seguridad: PAST Modelo de seguridad se basa en los siguientes supuestos:(14) Es computacionalmente imposible de romper el cripto sistema de clave pública y la función de hash criptográfica utilizando past , (15) mientras que los clientes, los operadores de nodo y software de nodos no son de confianza y los atacantes pueden controlar el comportamiento de cada uno de los nodos del pasado, se supone que la mayor los nodos de la red superpuesta se comportan bien, y, (16) un atacante no puede controlar el comportamiento de las tarjetas inteligentes. En la siguiente discusión, se supone el uso de tarjetas inteligentes. Como se discute más adelante en esta sección, es posible operar una red PAST sin tarjetas inteligentes. Cada PAST y cada nodo de usuario del sistema mantenga una tarjeta inteligente. El par de claves pública / privada está asociado a cada tarjeta. Cada la clave pública de la tarjeta inteligente se firma con la entidad emisora de la tarjeta inteligente clave privada para fines de certificación. Las tarjetas inteligentes generar y verificar diversos certificados utilizados durante la inserción y recuperar las operaciones y mantener las cuotas de almacenamiento. Siguiendo, esbozamos las funciones relacionadas con la seguridad principal.

- Gestión de almacenamiento y almacenamiento en cachéLa asignación estadística de los archivos a los nodos de almacenamiento en PAST aproximadamente equilibra el número de archivos almacenados en cada nodo. Sin embargo, no uniformes capacidades del nodo de almacenamiento y tamaño de los archivos de almacenamiento requieren de equilibrio de carga más explícita a permitir un comportamiento elegante bajo una alta utilización de almacenamiento global; y, no uniforme popularidad de los archivos requiere el almacenamiento en caché para minimizar la distancia y buscar para equilibrar la carga de la consulta. PAST emplea un esquema de gestión de almacenamiento que consigue una alta utilización de almacenamiento global al tiempo que rechaza algunos presentar solicitudes de inserción. El esquema se basa sólo en la coordinación local entre los nodos de un conjunto de hojas, e impone poco encima de la cabeza. Los resultados experimentales muestran que el pasado puede lograr la utilización del almacenamiento global de más de 95%, mientras que la tasa de rechazados inserciones de archivos se mantiene por debajo del 5% y las inserciones fallidas están fuertemente sesgadas hacia archivos de gran tamaño [16]. Cualquier nodo PAST puede almacenar en caché copias adicionales de un archivo, que logra el equilibrio de carga de consulta, de alto rendimiento para formatos más populares, y reduce la distancia y buscar tráfico de red. [16].

• Napster fue un servicio de distribución de archivos de música (en formato MP3), la primera gran red P2P de intercambio creado por Shawn Fanning. Su popularidad comenzó durante el año 1998. Su tecnología permitía a los aficionados a la música compartir sus colecciones de MP3 fácilmente con otros usuarios, lo que originó las protestas de las instituciones de protección de derechos de autor. La primera versión de Napster fue publicada a finales de 1999. Fue el primero de los sistemas de distribución de archivos entre pares de popularidad masiva, y era una red centralizada, ya que utilizaba un servidor principal para mantener la lista de usuarios conectados y archivos compartidos por cada uno de ellos. Las transferencias de archivos, sin embargo, eran realizadas entre los usuarios sin intermediarios. Fue el primero de los sistemas de distribución peer-to-peer masivamente populares, aunque no era totalmente peer-to-peer debido a que usaba servidores centrales para mantener una lista de todos los sistemas conectados y los archivos que eran distribuidos, mientras que las transacciones eran de hecho realizadas entre las máquinas. Aunque ya existían redes que facilitaban la distribución de archivos a través del internet como IRC, Hotline y USENET, Napster se especializaba directamente en música en la forma de archivos MP3, presentados a través de una interfaz amigable al usuario. El sistema back-end fue diseñado por el principal arquitecto, Jordan Mendelson.¹⁰ El resultado fue un sistema cuya popularidad generó una enorme selección de música para descargar.

V. TRABAJOS RELACIONADOS

Durante la consulta del estado del arte hemos encontrado varios trabajos entre los cuales tenemos [9], este proyecto muestra a un multiusuario completamente descentralizado de lectura y escritura peer-to-peer sistema de archivos con las propiedades de la localidad buenas. En este sistema todos los datos están contenidos en bloques almacenados utilizando el past la tabla de hash distribuidas (DHT), aprovechando así la tolerancia a fallos y las propiedades de la localidad de Past y Pastry. También se introduce una modificación a la DHT del past que permite aumentar aún más el rendimiento cuando se utiliza un modelo de consistencia relajado, pero útiles sin embargo, en este proyecto se desarrollo un prototipo con el fin de evaluar el desempeño de este diseño. Este prototipo está programado en Java y utiliza el FreePastry implementación en código abierto del past y Pastry. Permite que las aplicaciones que elegir entre dos grados de consistencia. Los resultados preliminares obtenidos a través de la simulación sugieren que nuestro sistema es aproximadamente dos veces tan lento como NFS.

Otro de los trabajos realizados tenemos, el diseño del past a gran escala, basada en Internet, la utilidad de almacenamiento

global que proporciona escalabilidad, alta disponibilidad, persistencia y la seguridad [10]. PAST es una aplicación de Internet peer-to-peer, y es totalmente auto-organización. Estos Nodos past sirven como puntos de acceso para los clientes, participar en el enrutamiento de las solicitudes de los clientes, y contribuir al sistema de almacenamiento. Sin embargo, este diseño de sistema es capaz de ofrecer garantías sólidas, acceso a almacenamiento eficiente, balanceo de carga y escalabilidad. Entre los aspectos más interesantes del diseño del past son (1) la ubicación de Pastry y el esquema de enrutamiento, lo que de forma fiable y eficiente las solicitudes de cliente rutas entre los nodos pasado, tiene buenas propiedades de las localidades de la red y resuelve automáticamente fallos en los nodos y las adiciones de nodos, (2) el uso de asignación al azar para garantizar la diversidad en el conjunto de nodos que almacenan réplicas de un archivo y para proporcionar equilibrio de carga, y (3) el uso opcional de tarjetas inteligentes, que están en poder de cada usuario PAST y emitidos por un tercero llamado un agente de la ayuda tarjetas inteligentes un sistema de cuotas que equilibra la oferta y la demanda de almacenamiento en el sistema.

También tenemos [11], este trabajo se realizo por medio de investigaciones sobre redes de sensores inalámbricos además se estudiaron e implementaron en muchos lugares. A pesar de esto Recientemente diversos trabajos para compartir los datos de los sensores existentes han sido juzgados. Pero En este trabajo se propuso realizar un sistema basado en P2P para el intercambio de datos de los sensores. Se utiliza una estructura P2P – Pastry para proporcionar un enrutamiento basado DHT sin servidor y se ha desplegado en Koren. Se clasifican los datos de los sensores en tiempo real de datos y los datos anteriores, y se escriba en pastry estos se aplican a los datos de multidifusión en tiempo real y el past en pastry se utiliza para almacenar y suministrar los datos anteriores. Además, como Pastry sí misma no proporcione la función de búsqueda, se propuso utilizar PAST para proporcionar la función de búsqueda mediante palabras clave.

Podemos ver que Muchas de las redes P2P que ofrecen diferente superposición de varias funcionalidades, dirigidos a aplicaciones específicas, se han propuesto en los últimos cinco años. Ahora es razonable considerar que las superposiciones múltiples pueden ser desplegadas en un amplio conjunto de nodos para que la plantilla más adecuada pudiera ser elegida en función de la aplicación.

Por eso también se realizo un estudios sobre [12] que es Un par físico, y este puede alojar varias instancias de sus compañeros lógicos que pertenecen a diferentes redes superpuestas. En este trabajo, nos muestran que la coexistencia de una superposición estructurada P2P y uno no estructurado puede ser aprovechada para que, mediante la construcción, la otra se construye automáticamente también. Más

específicamente, nos muestran que la aleatoriedad estructurada proporcionada por un gadget basado en la superposición puede ser utilizado para construir la tabla de enrutamiento de una superposición estructurada P2P y la aleatoriedad en los vínculos de proximidad numérica en las redes estructuradas proporciona la toma de muestras al azar de pares requeridos por el gadget-basado en superposiciones no estructuradas. En este trabajo, se muestran que el mantenimiento del conjunto de la hoja de pastry y los vínculos de proximidad de una plantilla estructurada es suficiente para construir las plantillas completas.

Demás se puede ver que IP Multimedia Subsystem (IMS) es considerada como la principal tecnología de red de próxima generación, en este trabajo [13] se usa un amplio uso de peer-to-peer (P2P) en Internet promovido la investigación de la combinación de IMS y P2P, este trabajo propuso un intercambio de archivos P2P sistema basado en IMS. Los temas a ser considerados incluyen la combinación de P2P y SIP (Session Initiation Protocol), NAT (Network Address Translators) y firewall transversal. El marco de este sistema es en SIP sobre redes P2P-construir en IMS y adoptar la estructura P2P híbrida, la superposición de P2P es DHT Pastry (Distributed Hash Table) de la red.

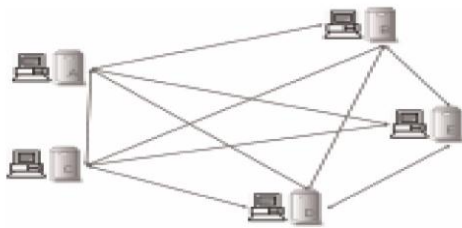


Figura 5 Red p2p descentralizada.

Cuando el nodo de interés se encuentra en este vector, se envía directamente el mensaje, sino se utiliza la tabla de enrutamiento de la siguiente forma: Los dígitos de los GUIDs funcionan como llaves en las tablas de enrutamiento.

0	0	1	2	3	4	5	6	...	F
1	10	11	12	13	14	15	16	...	1F
2	120	121	122	123	124	125	126	...	12F
3	1230	1231	1232	1233	1234	1235	1236	...	123F

Figura 6: cuatro primeras filas de la tabla de enrutamiento del nodo con GUID: 1234 [7]

La figura 6 muestra las primeras filas de la tabla de enrutamiento (para GUIDs de 128 bits las tablas tendrán 32 filas) de un nodo cuyos primeros dígitos de su GUID son 1234.

En las celdas de cada fila se hace variar el último dígito, conservando los anteriores que corresponden al nodo al que pertenece la tabla, así pues, en la fila cero, se tiene una variación de 0 a F, en la uno todas las celdas empiezan con 1 y variamos entonces el segundo dígito, la fila 2 tiene fijo el 12 y variamos el tercer dígito una vez más, y continuamos así hasta el final. La búsqueda de los datos sobre la ubicación del nodo se realiza comparando, hasta donde sea posible, dígito por dígito entre los GUIDs del Nodo actual y el de destino en la tabla de enrutamiento, acercándose así al el cada vez más, si no se puede continuar recorriendo la tabla del nodo actual y no se ha encontrado el GUID exacto del destino, se envía la búsqueda al nodo que se encuentre en la celda con la llave más cercada. Por ejemplo: En una celda A379F1 se podría encontrar la IP del nodo A379F14F y podemos inferir que si el mensaje es enviado a este nodo, la celda pertenecía a la tabla de un nodo con GUID A379FXYZ donde 0? X? F tal que X?1 y 0? Y, Z? F. O gráficamente: Si enviamos un mensaje del nodo 1234 al nodo 134D, el nodo 1234 busca en su tabla la celda con el GUID más cercano al deseado, como podemos ver, no es posible que la búsqueda pase de la segunda fila, ya que los nodos comparten solamente un dígito de sus GUIDs:

0	0	1	2	3	4	5	6	...	F
1	10	11	12	13	14	15	16	...	1F
2	120	121	122	123	124	125	126	...	12F

Figura 7: tabla de enrutamiento del nodo 1234 [7]

La celda 13 contiene la información del nodo más cercano en términos de su GUID al nodo 134D, ahora supongamos que el GUID de la celda 13 contiene la IP del nodo 135A, nos vamos para ese nodo:

0	0	1	2	3	4	5	6	...	F
1	10	11	12	13	14	15	16	...	1F
2	130	131	132	133	134	135	136	...	13F
3	1350	1352	1352	1353	1354	1355	1356	...	135F

Figura 8: tabla de enrutamiento del nodo 135A [7]

Supongamos que llegamos a la celda 134 y en ella encontramos la IP del nodo 134D, aunque podríamos continuar por ejemplo si se trata del nodo 134F. Por tratarse de una estructura de tipo árbol, en Pastry se lograra llegar al nodo deseado con complejidad $O(\log N)$. Cuando un nodo nuevo quiere formar parte de la red, este adquiere su propio GUID o nodeId, luego notifica de su existencia a otro nodo en su vecindad "Tal nodo puede ser localizado automáticamente, por ejemplo, usando "expanding ring IP multicast" o ser obtenido por el administrador del sistema a través de canales externos" [6], otra opción para la inicialización de nodos en este y otros

modelo similares es la propuesta de crear un "anillo universal" que provea este tipo de servicios[8]. El nodo contactado (A) reenvía el mensaje al nodo más cercano (en términos de GUID) en su tabla (B), y además podemos asegurar que la primera fila de la tabla de enrutamiento de este nodo (A) sirve como base para el nuevo nodo (por ser un nodo cercano en cuanto a una métrica de red), y además: el segundo (B) y tercer nodo contactado (C) comparten su segunda y tercer fila con el nuevo nodo, y así sucesivamente hasta formar su tabla de enrutamiento completa. Además, el vector de vecinos del último nodo le sirve al nuevo nodo ya que son vecinos. Cada nodo continuamente notifica a sus vecinos que está activo, si el nodo falla, los nodos en su tabla dejarán de recibir la señal y asumirán que se presentó un error, entonces, actualizarán su tabla para remplazarlo por otro nodo activo. Por supuesto si un nodo desea abandonar la red, este notificará voluntariamente a los demás para que cambien sus tablas, en todas las filas de la tabla de enrutamiento excepto la última, se hace referencia a uno de varios nodos posibles, si existe un nodo que pueda ocupar una posición y este se encuentra más cerca (métrica de red), se cambiara la tabla para hacer referencia al más cercano para mejorar la eficiencia. Con lo anterior se tiene una idea global del funcionamiento de Pastry como modelo de enrutamiento. No es el único foco de trabajo para este tipo de redes, pero es una aproximación interesante para la solución del tipo de problemas que se pueden presentar en las redes p2p antes descritas. La mayoría de estos trabajos son relativamente nuevos lo que hace que las implementaciones sean poco difundidas y con algunas limitaciones, para el caso de Pastry se tiene disponible FreePastry desarrollado por Rice University, Houston, USA bajo el modelo de software libre. [3]

VI. FUNCIONAMIENTO DE LA APLICACIÓN

Con la realización de esta aplicación se obtendrán muchos beneficios respecto a seguridad, ya que con este se busca un control a los accesos que realizan los clientes a la hora de almacenar o descargar información a través de la red, este sistema cuenta con una amplia disponibilidad de la información que es replicada en varios nodos, dichos nodos donde es replicada la información cuenta con una administración, conectividad, reglas y una ubicación cercana a los nodos que necesiten dicha información.

Este sistema es importante porque en él se pueden crear un conjunto de reglas para poder identificar si el nodo que solicita dicha información es confiable o no, con esto se busca establecer un control de seguridad sobre quien trabaja en dicha información.

La información almacenada en este sistema cuenta con un id casi único que se genera a la hora de almacenarlo por medio de una función hash, por lo siguiente no puede insertarse la misma información con el mismo id, esta información almacenada puede contener un sistema de encriptado en el cual el propietario podrá decidir si compartirla con alguien o no, este sistema no permite la eliminación de ningún tipo de información por motivos de seguridad, por el cual siempre estará disponible, este sistema p2p con past está basado en pastry el cual es el anillo de red en el que se busca un nodo responsable de la información a almacenar y el que genera dicho id.

Para la realización de este sistema empezaremos utilizando una clase llamada "Mynode" la cual se encargara de crear un nodo con el cual se podrá dar comienzo a un anillo de conexión o unirse a uno ya existente, los pasos a seguir para la realización de un nodo son:

1. Realizaremos la creación de un nodo de pastry.
 2. Creamos un método para la conexión de los nodos el cual se basara en 4 parámetros importantes que son:
 - Bindport: el cual es un puerto local disponible desde el cual escuchara el cliente, no es necesario que sea igual en todos los nodos conectados en dicho anillo.
 - InetAddress: el cual es el puerto a través del cual se realizara el anillo de conexión, este puerto tiene que ser igual en todos los nodos que quieran conectarse a un anillo en específico.
 - IpAddress: es la ip de cualquier nodo conectado en el anillo al cual se desea unirse o en caso dicho crear.
 - Environment: es el que se encarga de colocar el ambiente y las propiedades dentro del nodo pastry.
 3. Además de los 4 parámetros mencionados anteriormente se requiere de una clase llamada NodeIdFactory la cual se encargara de crear un id aleatorio para el nodo que será conectado dentro del anillo.
 4. También se requiere una clase llamada PastryNodeFactory la cual contiene todos los parámetros mencionados al principio, lo que realizara esta clase es en sí la creación de los nodos usando la información ingresada (Id del nodo, Puerto Local, Environment).
 5. También contiene unos métodos adicionales como por ejemplo el boot el cual se encarga de otras funciones dentro de la clase.
 6. Estos pasos anteriores son los necesarios para la creación de un nodo utilizando pastry.
- Luego de la creación del nodo viene la creación de los directorios en el cual se almacenara la información enviada por medio de la aplicación Past. Lo siguiente es la realización de un procedimiento llamado "Send" el cual será el encargado de enviar la información a través del anillo pastry, este procedimiento viene acompañado de otro procedimiento

llamado “getBytesFromFile” el cual su función es coger el archivo físico que se desea enviar y cargarlo en un array de bytes en memoria para poder realizar posteriormente el envío del mismo a través del anillo, el envío se realizara cogiendo el array de bytes creado con anterioridad y llevarlo a una clase llamada “MypastContent” la cual le asignara el id del nodo responsable de dicho array para poder localizarlo posteriormente (como se muestra en la figura 2).

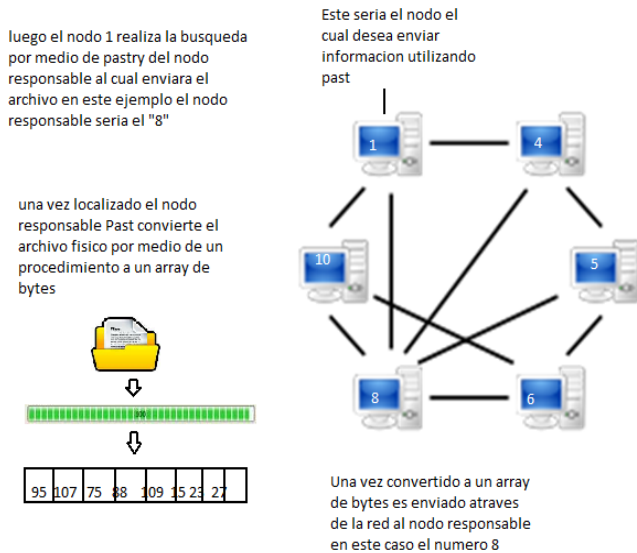


Figura 2

Para la localización de un archivo guardado anteriormente usare un procedimiento llamado “Find” el cual se encargara de realizar la búsqueda dentro del anillo, buscando el nodo responsable del archivo solicitado, una vez localizado dicho archivo que en este caso siempre será un array de bytes, es enviado a un procedimiento llamado “escribirarchivo” en el cual por medio de dos parámetros enviados con anterioridad que son el array y la ubicación donde se desea almacenar dicho archivo, realizara la conversión, este procedimiento es totalmente inverso al que realiza el procedimiento “getBytesFromFile” en este caso lo que realizara es que el array de bytes es convertido nuevamente en un archivo físico el cual será almacenado en la ubicación dada anteriormente (como se muestra en la figura 3).

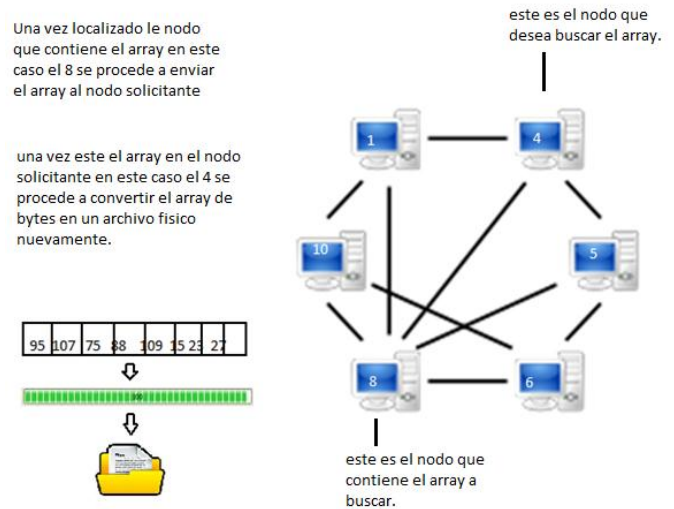


Figura 3

VII. RESULTADOS Y CONCLUSIONES

Los resultados obtenidos por medio de esta aplicación se pudo conseguir el envío y la respectiva recuperación de un archivo enviado a través de una red Pastry con la ayuda de la aplicación Past. Con el siguiente proyecto se pudo demostrar que al trabajar en una red P2P con PASTRY la cual funciona en la capa de aplicación, se consiguió la creación de un anillo de conexión con PASTRY el cual se encarga de crear y organizar los nodos dentro de la red y crear una aplicación utilizando PAST para el envío de información por medio de la red PASTRY. Actualmente, en Internet el ancho de banda o las capacidades de almacenamiento y cómputo son recursos caros. En aquellas aplicaciones y servicios que requieran una enorme cantidad de recursos pueden usarse las redes P2P.

Algunos ejemplos de aplicación de las redes P2P son los siguientes:

- Intercambio y búsqueda de ficheros. Quizás sea la aplicación más extendida de este tipo de redes. Algunos ejemplos son BitTorrent o la red eDonkey2000.
- Sistemas de ficheros distribuidos, como CFS o Freenet.
- Sistemas para proporcionar cierto grado de anonimato, como i2p, Tarzan o MorphMix. Este tipo de tecnologías forman parte de la llamada red oscura y constituyen el llamado peer-to-peer anónimo.
- Sistemas de telefonía por Internet, como Skype.
- A partir del año 2006, cada vez más compañías europeas y norteamericanas, como Warner Bros o la BBC, empezaron a ver el P2P como una alternativa a la distribución convencional de películas y programas de televisión, y ofrecen parte de sus contenidos a través de tecnologías como la de BitTorrent.[2]
- Cálculos científicos que procesen enormes bases de datos, como los procedimientos bioinformáticos.
- Monedas virtuales para transacciones entre partes. Bitcoin.

Aplicaciones creadas en Pastry: Pastry específica cómo las claves se distribuyen entre los nodos y cómo el nodo responsable puede encontrar la llave. Usando esto como un sustrato para un mayor protocolo Pastry permite implementar la funcionalidad tal como un sistema de archivos distribuido, un sistema de suscripción y publicidad o cualquier otro sistema que se puede reducir a almacenar valores y recuperarlos posteriormente.

VIII. REFERENCIAS

- [1] Ian Foster, Carl Kesselman *La Grilla 2: Planos Para una Nueva Infraestructura Informática (The Grid 2: Blueprint for a New Computing Infrastructure)*. Morgan Kaufmann Publishers. ISBN (1999).
- [2] [Online]. Available: RTVE adopta el P2P
- [3] A. Rowstron and P. Druschel "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems". IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany: 329–350. (Nov 2001).
- [4] A. Rowstron, A-M. Kermarec, M. Castro and P. Druschel (Nov 2001). "SCRIBE: The design of a large-scale event notification infrastructure". NGC2001 UCL London.
- [5] C. G. Plaxton, R. Rajaraman, and A. W. Richa. "Accessing nearby copies of replicated objects in a distributed environment". In *Proceedings of the 9th Annual ACM Symposium on Parallel Algorithms and Architectures*, Newport, Rhode Island, pages 311-320, June 1997.
- [6] A. Rowstron, P Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. En 18th IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg, Germany, (Middleware 2001). November 2001.
- [7] G. Coulouris, *Distributed Systems: Concepts and design*. Estados Unidos: Addison-Wesley, pp.397-430. 2005.
- [8] Castro, M., Druschel, P., Kermarec, A., y Rowstron, A. One ring to rule them all: service discovery and binding in structured peer-to-peer overlay networks. En *Proceedings of the 10th Workshop on ACM SIGOPS European Workshop: Beyond the PC*, Saint-Emilion France. 2002.
- [9] Picconi, F. Exploiting network locality in a decentralized read-write peer-to-peer file system.
- [10] Druschel, P. PAST: a large-scale, persistent peer-to-peer storage utility.
- [11] Wang-Cheol Song .Pastry Based Sensor Data Sharing.
- [12] Maniyamaran, B. Build One, Get One Free: Leveraging the Coexistence of Multiple P2P Overlay Networks.
- [13] Huasong Wang. Design of IMS-based P2P file-sharing system architecture.
- [14] The Gnutella protocol specification, <http://dss.clip2.com/GnutellaProtocol04.pdf>. 2000
- [15] R. Anderson. The Eternity service. In *Proc.PRAGOCRYPT'96*, pages 242–252. CTU Publishing House,Prague, Czech Republic. 1996.
- [16] A. Rowstron and P. Druschel. Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility. In *Proc. ACM SOSP'01*, Banff, Canada, Oct. 2001.
- [17] D. Eastlake, 3rd and P. Jone. RFC 3174: US secure hashing Algorithm 1, Sept. 2001.
- [18] Napster. <http://www.napster.com/>.
- [19] J. G. Kreifeldt, "An analysis of surface-detected EMG as an amplitude-modulated noise," presented at the 1989 Int. Conf. Medicine and Biological Engineering, Chicago, IL.
- [20] J. Williams, "Narrow-band analyzer (Thesis or Dissertation style)," Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.
- [21] N. Kawasaki, "Parametric study of thermal and chemical nonequilibrium nozzle flow," M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.