Análisis del desempeño de una red P2P bajo ataques de negación de servicio con nodos coalicionados

por

Biol. Armando de Jesús Ruiz Calderón

Tesis

Presentada al Programa de Graduados de la

Escuela de Tecnologías de Información y Electrónica

como requisito parcial para obtener el grado académico de

Maestro en Ciencias

especialidad en

Tecnología Informática

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

Diciembre de 2007

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

Escuela de Tecnologías de Información y Electrónica

Programa de Graduados

Los miembros del comité de tesis recomendamos que la presente tesis de Armando de Jesús Ruiz Calderón sea aceptada como requisito parcial para obtener el grado académico de **Maestro en Ciencias**, especialidad en:

Tecnología Informática

Comité de tesis:

Dr. Jorge Carlos Mex Perera
Asesor de la tesis

Dr. Gerardo Antonio Castañon Ávila
Sinodal

Dr. Cesar Vargas Rosales
Sinodal

Dr. Graciano Dieck Assad
Director del Programa de Graduados

Diciembre de 2007

Quiero dedicar este trabajo A mis Padres Armando y Delfina con todo mi cariño, admiración y reconocimiento, a mi querida hijita Maria del Rocio, a mis hermanos Aleyda y Efren, a mis lindos sobrinos Santiago y Mariana. Con todo mi cariño y agradecimiento por su apoyo Armando

Agradecimientos

Quiero Agradecer a mi Asesor el Dr. Carlos Mex todas las atenciones que recibí, además de sus consejos, ayuda y paciencia que tuvo conmigo para poder realizar este trabajo, ya que sin su ayuda no podría haber culminado esta etapa; de manera muy especial quiero dar las gracias a mi amigo Alberto Martinez Herrera por todas sus atenciones y apoyo que me brindó.

También quiero dar las gracias al Dr Raúl Perez quien me apoyo desde que llegue al Campus.

A mis amigos Primitivo, Ivan, Juan Paulo, Arturo por su apoyo en estos tiempos tan difíciles.

Finalmente quiero agradecer a mi amigo Oscar Morales ya que sin su oportuna ayuda, no habría podido alcanzar esta meta.

A todos ustedes Muchas Gracias

Armando de Jesús Ruiz Calderón

Instituto Tecnológico y de Estudios Superiores de Monterrey
Diciembre 2007

Análisis del desempeño de una red P2P bajo ataques de negación de servicio con nodos coalicionados

Armando de Jesús Ruiz Calderón, M.C.
Instituto Tecnológico y de Estudios Superiores de Monterrey, 2007

Asesor de la tesis: Dr. Jorge Carlos Mex Perera

El uso de las computadoras, el trabajo en red, el acceso a Internet son actividades cotidianas en nuestros dias, y representan un medio de comunicación de gran importancia; normalmente, se utiliza la arquitectura cliente - servidor para el trabajo en red, aunque este modelo ha sido muy estudiado, presenta una serie de limitaciones, que afectan su desempeño, frente a estas limitaciones existen alternativas de solución como las redes de arquitectura descentralizada o P2P, las cuales tienen características, que pueden mejorar el desempeño del trabajo en red, la alta aceptación que han tenido éstas redes, ha propiciado el desarrollo de aplicaciones que aprovechan las características de éstas y la utlización de sustratos especializados como Pastry para el envío y recepción de mensajes es una alternativa muy adecuada para la implementación de aplicaciones de distribución amplia, sin embargo existen problemas como, la seguridad que se encuentran en investigación. Los ataques como el de negación de existencia, son un problema de seguridad que requiere de estudio, estos ataques provocan pérdida en los mensajes enviados y aumento en el tráfico de la red. El análisis al desempeño de una red P2P bajo ataques de negación de existencia muestra que con una pequeña modificación al algoritmo de ruteo de Pastry se obtiene una mejora muy importante en el desempeño de la red y la cantidad de mensajes que se pierden se reduce significativamente, con lo que este tipo de redes sirven de manera adecuada para el despliegue de aplicaciones de amplia distribución.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Las computadoras, el trabajo en red y el acceso a Internet son, hoy en día, actividades cotidianas en nuestra sociedad. Internet y el acceso a la WEB representan un medio de comunicación de gran importancia.

La arquitectura tradicional de trabajo en red es conocida como modelo "Cliente-Servidor", y aunque es un modelo muy estudiado y ampliamente utilizado, presenta una serie de limitaciones, que afectan su desempeño, entre ellas: la congestión en el tráfico, la centralización de los recursos, y la poca robustez del modelo.

Frente a estas limitantes, se presentan algunas alternativas de solución como las redes de arquitectura descentralizada o P2P[1], las cuales tienen características que pudieran representar una mejora en el desempeño de la red, son descentralizadas, auto organizables, escalables, y robustas.

Como ejemplo de lo anterior se pueden encontrar diversas aplicaciones que corren en redes P2P como son *PAST*, *Scribe*, *Squirrell*, *CAN*, *Chord*, *Pastry*, *Tapestry* etc. las cuales son aplicaciones para el almacenamiento y recuperación de mensajes en redes de tipo P2P y en el caso de las últimas tres, son sustratos que sirven de base para las aplicaciones antes mencionadas, y tienen características propias en sus algoritmos.

No obstante las ventajas que presentan las redes de arquitectura descentralizada, uno de los problemas serios al cual los investigadores han puesto mucha atención es el que concierne a la seguridad, debido a que este problema tiene repercusiones muy importantes en el uso de este tipo de redes, como son la autenticidad, la integridad y la confiabilidad de la información.

Existen diversos problemas que afectan la seguridad de las redes, sin embargo unos de los mas comunes son los ataques que éstas sufren.[2] S. Ariyapperuma et.al., en su

[1]Peer to Peer, por sus siglas en inglés

[2]En este trabajo se consiera como ataque, a las intromisiones de cualquier tipo, que sufre una red

trabajo "Security vulnerabilities in DNS and DNSSEC", presentan una clasificación de los ataques nombrando las siguientes categorías: *man in the middle*, *packet sniffing*, y *Denial of Service*, sin embargo estas no son las únicas categorías de ataques existentes, pues se puede incluir *DDoS* y al ataque conocido como *Spoofing*.

La última categoría de ataque involucra los siguientes tipos: el ataque de un nodo con control limitado, el ataque por difusión de información falsa en un nodo comprometido, ataque por múltiples nodos sin coordinación, y el ataque coordinado con múltiples nodos coalicionados. El resultado de este último tipo de ataques deriva en una pérdida de mensajes que tiene un alto costo, pues con ello aumenta el tráfico en la red, como consecuencia de las retransmisiones que se puedan dar y por consiguiente, aumenta el costo computacional.

Una alternativa de solución a este problema, es la utilización de sustratos especializados como Pastry para las redes de arquitectura descentralizada con el objeto de disminuir la pérdida de mensajes cuando reciben ataques, esto es, cuando se tienen muchos nodos coalicionados, como ejemplo se tiene el ataque de negación de existencia, o también cuando hay nodos en falla dentro de la red. Sin embargo haciendo una pequeña modificación al algoritmo de ruteo de Pastry se reduce de manera substancial la pérdida de mensajes, y aunque se eleva la cantidad de saltos promedio entre dos nodos, éste aumento no es significativo, pero en cambio, la reducción en la pérdida de mensajes si resulta muy satisfactorio.

1.1. Definición del Problema

Los problemas de seguridad en las redes de arquitectura descentralizada, como la falta de control sobre las actividades y protocolos donde participan nodos, el ingreso a la red de nodos maliciosos, la manipulación de la información, además de los ataques a los protocolos de comunicación o a los servicios que causan daño a la integridad, confidencialidad y autenticidad de la información, representan un aspecto importante en las redes con arquitectura descentralizada ya que el envío y recepción de paquetes o mensajes, y el buen funcionamiento de ésta tarea resulta una actividad crítica para todos los sistemas o servicios que trabajan en red y constituye un componente crítico en la infraestructura de Internet.[20] Los problemas de seguridad, autenticidad e integridad, han propiciado modificaciones y ajustes a los protocolos de ruteo y seguridad.

La utilización de nuevas arquitecturas para la implantación de protocolos de ruteo, tal como la P2P representa una opción viable que ha sido estudiada por diversos

por un agente externo, o por coalicón de elementos propios de la red. ver [22]

autores[3, 4, 6, 10, 11], para ser utilizada en diversas aplicaciones de ruteo a través de plataformas de amplia distribución como Internet, sin embargo a la fecha la literatura especializada no reporta la existencia de un análisis de robustez de una red P2P frente a ataques de negación de existencia[3] con diferentes grados de coalición en los nodos, que permita una evaluación mas concreta de la conveniencia de utilizar la arquitectura P2P como una opción adecuada para el envío y recepción satisfactorio de los mensajes a lo largo de una red.

Consideremos ahora el siguiente ejemplo que nos permite ver la dimensión del problema.

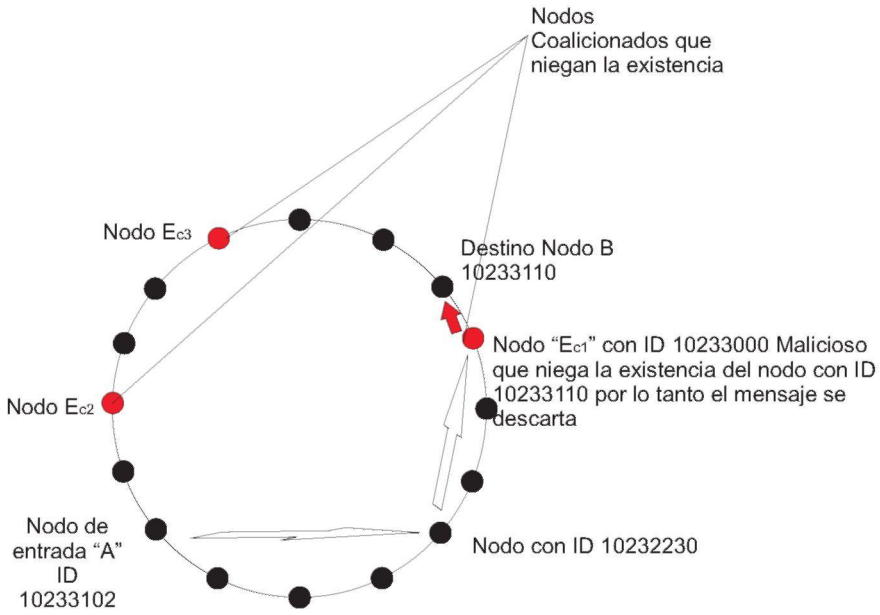

Figura 1.1: Esquema que muestra un ataque de negación de existencia

En la figura 1.1 se muestra una red P2P en la que existen 3 nodos coalicionados, en la figura se muestra como se realiza una búsqueda de información y se envía una consulta de información del nodo de entrada "A" al nodo destino "B"; el procedimiento de búsqueda de información es el siguiente.

De forma inicial se envía un mensaje, se incorpora a la red a través del nodo de entrada "A" este mensaje deberá ser enviado al nodo destino "B" en el primer salto el mensaje es enviado al nodo con el ID "10232230" ese nodo se encuentra en la tabla de

[3]De forma general se puede decir que consiste en el hecho de que cuando a un nodo malicioso o coalicionado, le llega una petición de ruteo para un nodo, este niega su existencia, siendo el mensaje descartado

ruteo del nodo "A" se revisa la tabla de ruteo, y el mensaje es enviado al nodo "E_{c1}" sin embargo ese nodo con el ID "10233000" es un nodo malicioso coalicionado con algunos otros. Este nodo nodo malicioso niega la existencia del nodo destino, entonces el mensaje es descartado y se pierde sin completar el servicio ofrecido por la red, con las respectivas consecuencias que trae consigo la pérdida de mensajes como son, entre otras: las retransmisiones, el incremento en el tráfico en la red, aumento en el costo computacional.

Resulta importante señalar que mientras mas nodos se encuentren coalicionados, mayor es la probabilidad de fracaso en el ruteo[16], y mayores serán los costos que se tengan que enfrentar para lograr un ruteo satisfactorio.

Como se puede observar una parte importante en la solución de este problema es encontrar un mecanismo que mitigue el efecto causado por el ataque; el cual debe generar una reducción significativa en la pérdida de mensajes a causa de los ataques de negación de existencia.

1.2. Objetivos

- Encontrar un mecanismo que ayude a mitigar los ataques de negación de existencia en una red P2P con nodos coalicionados.

1.2.1. Objetivos Específicos

- Analizar la robustez de una red P2P bajo ataques de negación de existencia con nodos coalicionados.
- Analizar la probabilidad de acceder con éxito a la información distribuida en una red P2P bajo diferentes niveles de ataque de negación de existencia con nodos coalicionados.
- Determinar los efectos de los ataques de negación de existencia en el desempeño de una red P2P.

1.3. Hipótesis

La explotación de información redundante contenida en las tablas de ruteo, puede ayudar a un mecanismo de defensa, a mitigar los ataques de negación de existencia.

1.4. Justificación

Las redes P2P representan avances tecnológicos recientes, que tienen entre sus características, bajo costo, su escalabilidad, y su tolerancia a fallos de cómputo y comunicación.[18]

Por otra parte, la arquitectura P2P no depende de servidores centralizados para proveer el acceso a los servicios y por lo tanto ofrece una alternativa interesante al modelo cliente servidor, especialmente por las aplicaciones desplegadas a gran escala.[12]

Estos factores han propiciado que esta arquitectura de red cuente con una amplia aceptación que ha llevado al desarrollo de aplicaciones para almacenamiento, compartición y recepción de archivos en diversos dominios de aplicación.

Debido a la gran aceptación que esta arquitectura está teniendo en los sistemas distribuidos, ha propiciado que los aspectos concernientes a la seguridad, sean un tema importante en el desarrollo de la investigación.

Cabe señalar que los trabajos basados sobre el análisis de las propuestas de sustratos como CAN, Chord, Pastry y Tapestry [4], generaron diferentes propuestas en cuanto sus algoritmos para el envío y recuperación de mensajes.

A la fecha se continuan trabajos de investigación que analizan la complejidad de los algoritmos utilizados en esos sustratos, en los cuales al parecer todavía no se ha reportado algún mecanismo que busque mitigar el efecto de los ataques por negación de existencia para esta arquitectura.

Por lo cual consideramos que la propuesta de este trabajo de investigación puede ser una aproximación significativa que contribuya a la solución de los problemas que conciernen, a la seguridad, para este tipo de redes.

1.5. Contribución

La evaluación de la conveniencia de utilizar una arquitectura basada en redes P2P como una alternativa de solución a los problemas de seguridad, autenticidad e integridad que enfrenta el envío y recepción de mensajes a través de una red de arquitectura descentralizada, implica la verificación y consideración de varios factores.

[4] [24, 10, 4, 6]

Este trabajo de investigación se enfoca en verificar la robustez de una red P2P bajo ataques de negación de existencia con nodos coalicionados, utilizando como estrategia la simulación de un ataque selectivo que permita hacer un análisis de la cantidad de mensajes perdidos contra la cantidad de nodos coalicionados y determinar la probabilidad de ruteo exitoso, además de proponer una solución para mitigar los efectos de este tipo de ataque, logrando así reducir el tráfico en la red ya que mientras mas mensajes lleguen de manera satisfactoria a su destino, menor será el tráfico innecesario y mejor será el desempeño de la red, mejorando así la eficiencia de la red.

Capítulo 2

Marco Teórico

El uso de computadoras y diversos dispositivos electrónicos, es una práctica común en nuestros días, el trabajo en red se considera cotidiano, estar en red permite estar conectado con el resto del mundo además de tener la facilidad para poder intercambiar información o compartir recursos.

Una red de computadoras se puede considerar como un sistema de comunicaciones, que permite comunicarse con otros usuarios, compartir información y recursos estos pueden ser ancho de banda, procesador, disco duro etc. Es decir es un sistema de comunicaciones que conecta a varias computadoras y que les permite intercambiar información.

La conexión no necesariamente se hace a través de un hilo de cobre, también puede hacerse mediante el uso de fibra óptica o tecnologías inalámbricas como puede ser infrarrojo, bluetooth o microondas en sus diferentes variantes, como se puede apreciar en la figura 2.1.

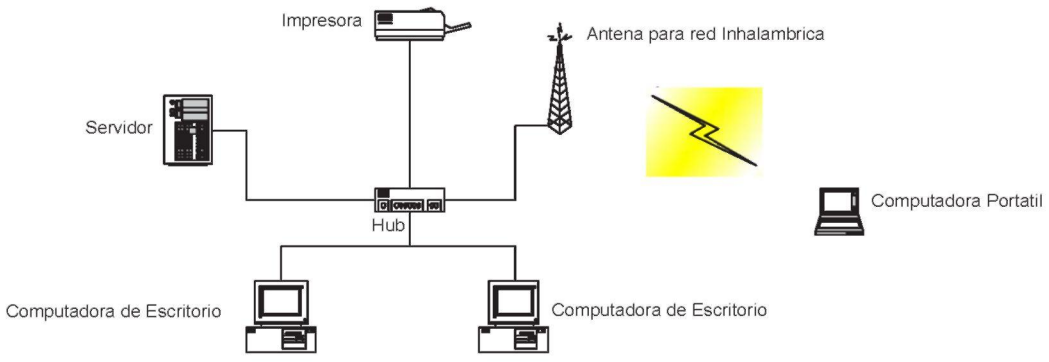

Figura 2.1: Esquema general de una red de computadoras

2.1. Clasificación de las Redes

Las redes han sido clasificadas de muchas formas siguiendo diferentes criterios entre las diferentes clasificaciones para las redes tenemos:

- **Por su ubicación**
  1. **PAN** Personal Area Network
  2. **LAN** Local Area Network
  3. **MAN** Metopolitan Area Network
  4. **WAN** Wide Area Network
- **Por su relación funcional**
  1. Cliente - Servidor
  2. P2P
- **Por su topología**
  1. Estrella
  2. Anillo
  3. Bus
  4. Estrella extendida
  5. Malla

2.2. Ataques

En la actualidad con el crecimiento explosivo de la WEB, se tienen diversos problemas de seguridad, autenticidad e integridad de la información, provocado por agentes hostiles que utilizan diversos mecanismos y procedimientos para generar ataques hacia las redes, utilizando las vulnerabilidades de las mismas.

Entre los principales tipos de ataques que se tienen, se encuentran los siguientes:

1. **Man in the middle**.- Este ataque consiste en que el recipiente de datos no tiene forma de autentificar el origen de los datos, ni tampoco puede verificar la integridad de la información[8]

2. **Packet Sniffing**.- Este ataque consiste en que alguna aplicación, envía una petición o una respuesta completa hacia la red, en un solo paquete de UDP, este datagrama no esta firmado ni esta encriptado. Con esto se pueden capturar los paquetes de las peticiones rápidamente y se puede generar una respuesta falsa para mandarla a quien solicito la información antes de que llegue la información correcta del servidor. Posteriormente, comprometiendo otros elementos de la red, el atacante puede capturar un paquete de respuesta y modificarlo, como no hay una forma de autentificar la fuente o la integridad de los datos este robo de datos no puede ser detectado.

3. **DoS attacks**.- Este ataque consiste en negar el servicio de múltiples formas y tiene un impacto significativo en los usuarios; este tipo ataque esta dirigido usualmente a los servidores raíz.

4. **DDoS**.- Este tipo de ataque se da cuando varios miembros de un sistema distribuido se encuentran comprometidos, usualmente uno o varios servidores web, con frecuencia se utiliza software malintencionado para generar este tipo de ataques, utilizando herramientas automatizadas, para explotar las debilidades de los sistemas atacados como por ejemplo "hoyos" de seguridad del sistema operativo.

   La mayor ventaja de un atacante al utilizar este tipo de ataque es el hecho de que se puede iniciar el ataque desde varias computadoras, con lo que generan mayor tráfico que el que puede generar una sola computadora, y consecuentemente cuesta mas trabajo poder detectar y eliminar el tráfico de varias computadoras que el de una y si de aumenta el ancho de bando en host objetivo, facilmente se puede aumentar el tráfico, lo que puede no ser de gran ayuda.

5. **ID spoofing**.- Este tipo de ataque también conocido como robo de identidad, se puede describir en el contexto de la seguridad de las redes, como una situación en la que una persona o programa se enmascara de forma satisfactoria y se hace pasar por otra, falsificando datos y obteniendo ventajas ilegítimas por esta situación.

   Existen algunas variantes de esta categoría de ataques, pues el robo de identidad se puede dar por teléfono, por correo electrónico, a través de páginas web, como por ejemplo las páginas de los bancos, en la que los atacantes reproducen la página web atacada en su apariencia externa, las victimas creen que están en la página confiable, sin embargo están conectados a páginas fraudulentas que solo recaban información para delitos posteriores.

2.3. Arquitectura Cliente - Servidor

Basado en la importancia de la funcionalidad de las redes, se puede observar que la arquitectura mas utilizada para trabajo en red, es la conocida como "*Cliente - Servidor*" ésta se puede definir como:

- Arquitectura de red, que generalmente consiste de un proveedor de servicios, (Servidor), y muchos usuarios, (Cliente), que generalmente solo acceden a la red a través de una computadora personal. [11]

En esta arquitectura se tiene un control centralizado de los recursos de la red los cuales son administrados por un servidor, el cliente solicita el acceso a la red y a los servicios que el servidor proporciona.

El esquema básico de la arquitectura Cliente - Servidor se puede apreciar en la figura 2.2

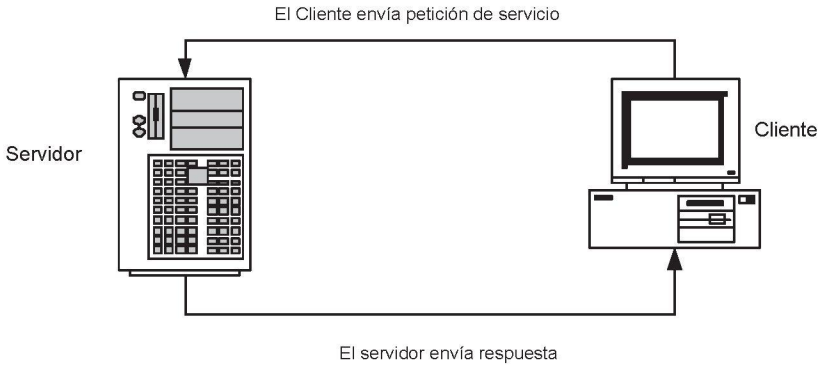

Figura 2.2: Esquema del modelo Cliente Servidor

Los roles de cada uno (Cliente y Servidor) se encuentan perfectamente definidos:

Servidor El servidor tiene las siguientes funciones:

- Esta escuchando el canal para recibir peticiones.
- Espera las peticiones.
- Recibe las peticiones.
- Procesa las peticiones y las contesta.
- Tiene el sistema operativo de RED que es el que realiza la administración de los recursos.

Cliente El cliente tiene las siguientes funciones:

- Se firma en la red.
- Envía peticiones.
- Espera la respuesta del servidor.
- Utiliza los servicios y recursos que le proporciona el servidor.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola computadora ni tampoco es necesariamente una sola aplicación, considerado que es muy frecuente en la arquitectura "Cliente- Servidor"que la función del servidor se descomponga en diferentes aplicaciones que pueden ser ejecutadas por diferentes computadoras aumentando así el grado de distribución del sistema como se puede observar en la figura 2.3, en la que diferentes servidores son utilizados para diferentes funciones bien determinadas.

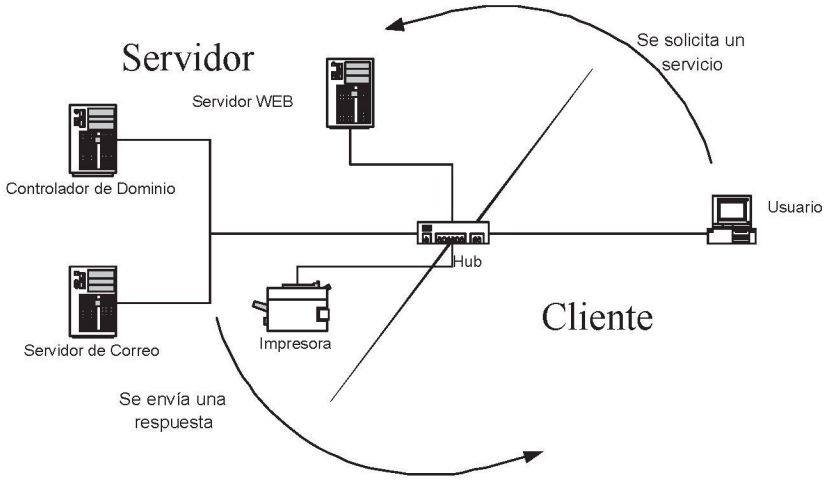

Figura 2.3: Esquema de una red bajo la arquitectura Cliente Servidor

2.3.1. Ventajas y Desventajas de la arquitectura Cliente-Servidor

Ventajas

- Los datos se almacenan en el servidor.
- Se tiene buen control de la seguridad.
- El servidor controla el acceso y los recursos.

- Existen tecnologías bien estudiadas y diseñadas, que brindan alta seguridad.
- Cualquier elemento de la red puede ser actualizado.

Desventajas

- La congestión del tráfico ha sido un problema el cual aún no se puede resolver.
- Cuando muchos clientes envían peticiones al mismo servidor de forma concurrente, se ocupa gran ancho de banda pudiendo generarse retardos en las respuestas hacia los clientes.
- El modelo Cliente - Servidor no es robusto, pues cuando el servidor tiene algún conflicto ó fallo y deja de funcionar, la red completa deja de funcionar.

2.4. Distributed Hash Tables

Las Tablas de Hash Distribuido (Distributed Hash Tables, DHT) son un tipo de sistemas distribuidos descentralizados que proveen el servicio de búsqueda y recuperación similar a una tabla "Hash"[1] a través de pares (nombre, valor), y reparten el conjunto de claves (keys) entre los nodos que participan en una red. Cada nodo es como una celda de una tabla hash. Las DHT son diseñadas para tratar un número grande de nodos además de estar procesando altas y bajas continuas de los nodos. La interfase de las DHT se puede utilizar para construir servicios más complejos y una larga variedad de tareas[17], como sistemas de almacenamiento y recuperación de archivos en redes P2P, almacenamiento cooperativo en Web entre otros.

2.5. Arquitectura P2P

Las redes P2P o *Peer to Peer*[2] (por sus siglas en inglés) de manera informal se pueden definir como un grafo conectado en el que dos nodos aleatoriamente seleccionados se comunican en una cantidad pequeña de saltos[12].

De manera mas formal se puede decir que una red con arquitectura P2P, (por sus siglas en inglés), es un sistema distribuido que sirve para implementar DHT, con el

[1]Estructura de datos, la cual mapea de manera eficiente las llaves k en valores y sirve como un "núcleo" para la construcción de aplicaciones de almacenamiento, búsqueda y recuperación de información.[24]

[2]En el contexto de redes, *peer* se denota como igual, por lo que al hacer mención al término *Peer to Peer* o P2P, se entenderá como de igual a igual

objeto de poder almacenar y recuperar información rápidamente[13], además contiene, una serie de nodos interconectados con capacidad de auto organizarse dentro de la red, con el propósito de compartir recursos, con una alta capacidad de recuperarse frente a fallas, y es adaptable al flujo de la población de nodos existentes dentro de la red.[21]

Dentro de las características mas relevantes de esta arquitectura destaca el hecho de que no tiene clientes ni servidores definidos, ya que los nodos se comportan de manera simultanea como clientes y como servidores, además de que se encuentran distribuidos a lo largo de diferentes dominios en internet[14], lo que trae como resultado que no tienen un control centralizado de los recursos; además las redes P2P, administran y optimizan el uso de el ancho de banda que acumulan de los otros usuarios, obteniendo como consecuencia de esto, mejor rendimiento en las conexiones y transferencias que con algunos métodos centralizados convencionales donde se tiene una cantidad relativamente pequeña de servidores, la cual provee el total de ancho de banda y recursos compartidos para un servicio o aplicación.

Esta arquitectura, intenta tener una alta capacidad de recuperación frente a fallas, su objetivo es tratar de rutear mensajes de forma correcta aún cuando una fracción de nodos fallen, o se desconecten[16], pero esto no significa que sea segura, pues con una pequeña fracción de nodos maliciosos ó en falla, se provoca una deficiencia en el envío de mensajes a los lugares correctos, también conocido como mal ruteo[16]

En la figura 2.4 se puede observar el esquema de una red P2P.

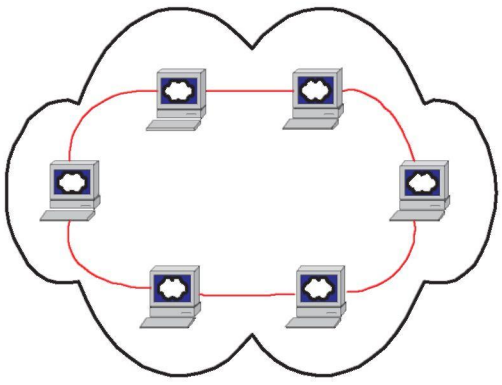

Figura 2.4: Ejemplo de una red P2P

La literatura especializada, considera que dentro de las redes con arquitectura descentralizada, existen dos generaciones de redes P2P.

La primera generación de redes P2P aún requería de un directorio centralizado, como ejemplo de esta generación esta "Napster"[3] ya para la segunda generación no se requería de ese directorio centralizado. En esta segunda generación se pueden encontrar dos categorías de redes P2P de acuerdo a su estructuración[12]:

- Redes no estructuradas
    1. **Gnutella**.- Es un protocolo de red para la distribución de archivos en un sistema distribuido que trabaja entre pares, en una red P2P pura.
    2. **Freenet**.-Es una red P2P diseñada para resistir la censura, la cual utiliza el ancho de banda y espacio de almacenamiento de los nodos para publicar u obtener información de todo tipo en completo anonimato.
- Redes estructuradas
    1. Pastry
    2. Chord
    3. Tapestry

Las redes P2P estructuradas son aquellas que tienen aspectos técnicos tales como:[4]

- La Escalabilidad
- La Robustez
- La Descentralización
- La Autorganización
- La Seguridad

Escalabilidad Las redes P2P pueden llegar a tener un alcance mundial con cientos de millones de usuarios potenciales. En general, se puede decir 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. Siendo esta la principal diferencia con la arquitectura tradicional de tipo cliente-servidor.

[3]Es un servicio de distribución y compartición de archivos y pionero de las redes P2P

Robustez La naturaleza distribuida de las redes P2P también incrementa la robustez, ya que en caso de haber fallas en la réplica de los datos hacia múltiples destinos, los sistemas P2P puros permiten a los *peers* encontrar la información sin hacer peticiones a ningún servidor centralizado. Considerando en este último caso, que 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.

Auto-organización Esta característica hace referencia hacia la facilidad con la que un nodo nuevo se puede incorporar a la red así como a la rápida actualización de las tablas de ruteo de los miembros de la red; lo mismo sucede cuando un miembro sale de la red, los nodos que se quedan activos, de manera rápida actualizan las tablas de ruteo.

Seguridad Esta es una de las características deseables de las redes P2P .

Dentro de los objetivos de seguridad en una red P2P segura, están:

- Identificar y evitar los nodos maliciosos.
- Evitar el contenido infectado.
- Evitar el espionaje de las comunicaciones entre nodos.
- La creación de grupos seguros de nodos dentro de la red.
- Integridad de la Información.
- Autenticidad de la información.

Actualmente existen proyectos que apuntan hacia la construcción de aplicaciones P2P y hacia un entendimiento más profundo y adecuado de los detalles y requerimientos de esas aplicaciones y sistemas entre estos trabajos se encuentran:

Pastry es un sustrato utilizado para ruteo de mensajes en una red de tipo P2P el cual utiliza DHT para el almacenamiento y recuperación de la información de cada nodo; la información de cada nodo esta almacenada de forma redundante en las tablas de ruteo de los elementos de la red, los cuales están conectados a internet.

Pastry utiliza un procedimiento especial para las entradas y salidas de nodos de la red, de manera dinámica, por su naturaleza redundante y descentralizada, no se considera que exista un solo punto de falla y cuando algún nodo desee separarse de la red lo puede hacer sin la necesidad de dar aviso alguno.

Chord es un protocolo distribuido de búsqueda que realiza una sola operación, esta operación sirve para tener un ruteo eficiente entre dos nodos, éste protocolo trabaja basado en una llave la cual se utiliza para referenciar a otro nodo, que se encuentra en su tabla de ruteo y utiliza unicamente $O(log^2 N)$ de la información total del ambiente en sus tablas de ruteo[10]

Tapestry es un sustrato dedicado al almacenamiento y localización de información, además de ser una plataforma para el ruteo de mensajes, ya que localiza información y la envía directamente hacia la copia mas cercana del objeto o servicio utilizando unicamente enlaces punto a punto, sin la utilización de recursos centralizados. El ruteo de información, así como la información contenida en los directorios, es de facil administración y reparación; Tapestry es auto administrable, tolerante a fallas y de alta recuperación frente a fallos, así como de baja carga.[6]

Además de los proyectos mencionados existen aplicaciones que utlizan alguno de los sutratos mencionados para sus procesos de almacenamiento y recuperación de infomación, entre estas aplicaciones tenemos:

Squirrel es una aplicación P2P,vdescentralizada, auto organizable, y con una alta capacidad de recuperación frente a fallas o ataques; en la que la idea básica es la de permitir a los navegadores de los nodos que estan conectados a internet, el que puedan compartir su cache, con el objeto de poder formar un web cache eficiente y escalable, sin la necesidad de tener que incorporar hardware adicional, con su consecuente aumento de el costo computacional. Squirrel aligual que otras aplicaciones de almacenamiento y recuperación de información al algoritmo de Pastry.[23]

PAST es una aplicación P2P dedicada al almacenamiento global y recuperación de archivos, que esta compuesta por nodos conectados a internet, cada nodo es capaz de iniciar y rutear peticiones del cliente las cuales pueden ser de almacenamiento y/o recuperación de archivos; los nodos de PAST están auto organizados y forman una red P2P de tipo estructurado; cabe señalar que dentro de sus características de funcionamiento PAST utiliza el algoritmo de ruteo de Pastry, cuando algún cliente realiza una petición determinada.[3]

No obstante lo arriba señalado, uno de los problemas clave de las redes P2P de gran escala, es el de proveer seguridad adecuada[4].

2.5.1. Ventajas y desventajas del modelo P2P

Ventajas

- Minimización de la congestión debido a que las conexiones se realizan punto a punto, no existen cuellos de botella.
- Es descentralizado
- Es anónimo
- Al añadir un nodo a la red no es necesario reestructurar la red
- Escalabilidad más sencilla al tener una menor congestión y autoorganización.

Desventajas

- Dificil de administrar.
- Incrementa la complejidad del entorno debido a la heterogeneidad de los *peers*.
- No existe ningún tipo de filtro.
- La seguridad depende de cada *peer* por lo que podría no ser confiable.

2.6. Pastry

Pastry es una arquitectura P2P estructurada, que utiliza DHT, para la búsqueda y localización de objetos, genera un anillo lógico con capacidad máxima de 2^{128}, en la que el elemento 1, está, junto al último elemento generado, ésta arquitectura posee un esquema de ruteo basado en la autorganización y nodos conectados a Internet.

Pastry es una topología completamente descentralizada, con una alta capacidad de recuperación frente a fallas, escalable y confiable; la literatura considera que Pastry tiene buenas caraterísticas de ruteo.

Esta topología intenta ser un sustrato general para el diseño y la implementación de una gran variedad de aplicaciones P2P que trabajan en internet como son:

- Compartición de archivos
- Almacenamiento de archivos
- Comunicación entre grupos.

Dentro de las características que hacen a Pastry un sustrato adecuado para las aplicaciones que requieren de una plataforma P2P se pueden observar las siguientes:

- Cada nodo tiene un número de identificador único "ID".
- Cada nodo envía eficientemente un mensaje, y cada mensaje contiene una llave numérica única.
- Cada nodo tiene su propia tabla de ruteo.
- Utiliza DHT para su organización en el anillo lógico que forma

Como un ejemplo de de estas aplicaciones tenemos a "PAST" y a "Squirrel"[3]

2.6.1. Descripción del Nodo

Cada nodo tiene un identificador único; llamado ID, el cual es asignado aleatoriamente de un espacio de 128 bits, el ID esta basado en 2^b, donde "b" es un parametro de configuración, el cual modifica las dimensiones de la tabla de ruteo, los nodos están uniformamente distribuidos en su espacio que va desde (0) hasta $(2^{128} - 1)$.

En la figura 2.5 se muestra el ejemplo de una tabla de ruteo completa, incluyendo sus tres elementos que la conforman, considerando a $b = 2$.

De donde tenemos que:

- N es el número total de nodos en la red
- L es el conjunto de hojas "Leaf Set"
- R es la tabla de ruteo "Routing Table"
- M es el conjunto de vecinos "Neighborhood Set"

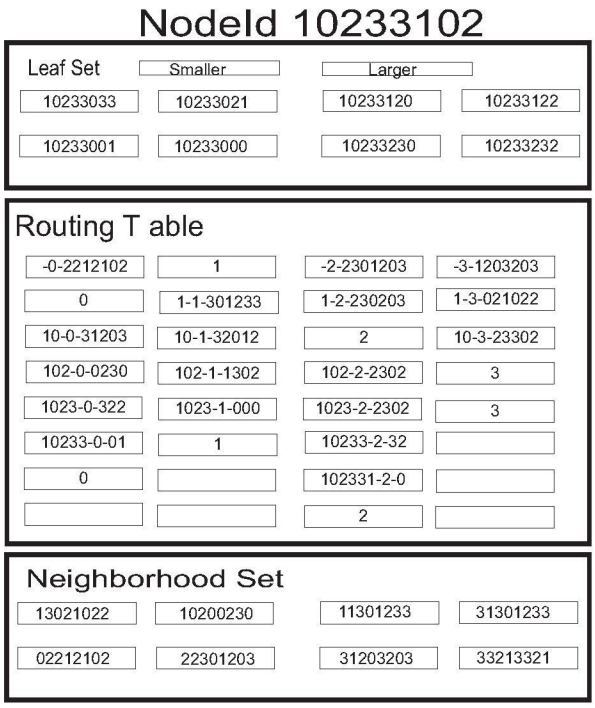

Figura 2.5: Ejemplo de una Tabla de ruteo completa

2.6.2. Leaf Set

El *leaf Set* contiene a los $L/2$ nodos numericamente mas grandes y mas cercanos al nodo actual y además contiene a los $L/2$ nodos numericamente mas pequeños y mas cercanos al mismo. El tamaño de "L" es tipicamente 2^b, o (2 x 2^b); aún cuando la cota superior del leaf set esta dada por un valor típico aproximado de: $\lceil 8 * log_{2^b}(N) \rceil$

Se debe entender claramente que, los nodos de "L" siendo numericamente los mas cercanos no implica que geográficamente sean cercanos.

En la figura 2.6 se muestra como se compone un leaf Set.

El conjunto de hojas "L", esta formado por los $L/2$ nodos mas cercanos y sirve para que de acuerdo a las necesidades de ruteo se pueda ir hacia la derecha o la izquierda en las tablas de ruteo.

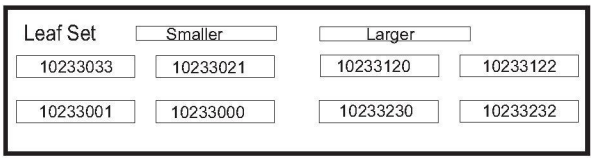

Figura 2.6: Ejemplo de un Leaf Set

2.6.3. Neighborhood Set

El *Neighborhood Set* "*M*" contiene los ID y las direcciones IP de los $|M|$ nodos mas cercanos de acuerdo con la métrica de proximidad[4], típicamente el tamaño de $|M|$ es de 2^b, o (2 x 2^b).

Normalmente el "*Neighborhood Set*" no se utiliza durante el ruteo de mensajes, sin embargo es útil en el mantenimiento de las propiedades de la localidad por ejemplo cuando se adiciona un nodo en la red, cuando se recupera un nodo de alguna falla ó bien, cuando un nodo sale de la red.

En la figura 2.7 se ejemplifica un "*Neighborhood Set*".

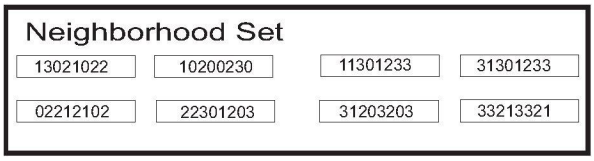

Figura 2.7: Ejemplo de un Conjunto de vecinos

2.6.4. Routing Table

Las tablas de ruteo estan organizadas en $log_{2^b}(N)$ filas y (2^b) columnas[4, 25], conteniendo 2^b entradas. La fila n de la tabla de ruteo contiene las direcciones "**IP**" de los ID que se encuentran en el *Leaf Set* [3] y comparten los primeros n dígitos del ID del nodo actual; el $(n+1)esimo$ dígito del ID del nodo en la columna m de la fila n es igual a m.

En la figura 2.8, se muestra como esta formada la tabla de ruteo.

[4]La métrica de proximidad puede ser la cantidad de saltos que debe dar un mensaje para llegar a su destino

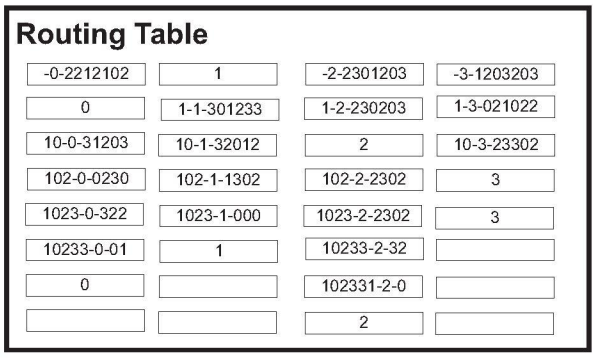

Figura 2.8: Ejemplo de una tabla de ruteo

2.6.5. Envío de Mensajes

Este sustrato tiene un sistema de ruteo muy eficiente, ya que cada nodo rutea un mensaje en $O(logN)$ saltos al destino[4] y el ruteo funciona de la siguiente manera:

Cuando un nodo envía un mensaje a otro nodo; el nodo origen de forma normal, envía el mensaje a otro nodo el cual en su ID comparte al menos un dígito o "d" dígitos de largo que estan en el ID del nodo origen[5]. Si no estuviera un ID conocido, el mensaje será enviado a un nodo el cual en su prefijo de su ID comparta, al menos la misma cantidad de dígitos que el nodo actual, pero que numericamente sea mas cercano al destino.

El procedimiento de ruteo que se muestra a continuación:

El proceso se ejecuta cuando un mensaje con una llave D llega a un nodo con un ID A, definiendo la siguiente notación:

- R_l^{Dl} representa la entrada a la tabla de ruteo R en la columna $i, 0 \leq l \leq 2^b$ y la fila $l, 0 \leq l < \lfloor 128/b \rfloor$
- L_i es el "*iésimo*" ID mas cercano en el *Leaf Set* L, $(L_{-\lfloor |L|/2 \rfloor} \leq D \leq L_{\lfloor |L|/2 \rfloor}$, donde el signo de positivo o negativo, indica que los ID son mas pequeños o grandes que el ID actual
- D_l es el valor de l digitos en la llave D
- $shl(A, B)$: representa la longitud del prefjo compartido entre A y B , expresada en dígitos

[5]El mensaje contiene una llave la cual sirve para el ruteo del mensaje

De donde se tiene:

if $(-L_{\lfloor |L|/2 \rfloor} \leq D \leq L_{\lfloor |L|/2 \rfloor})$
{
// D se encuentra dentro del rango del Leaf set
// envía en mensaje a L_1, el cual es el destino
// en caso contrario
}
else
{
//Si el ID al que se quiere llegar no esta en el Leaf Set entonces
// se utiliza la tabla de ruteo
let $l = shl(D, A)$;
$(R_l^{Dl} \neq null$ {
envía a R_l^{Dl} ;
}
else
{
//Si el ID que se busca no se encuentra en la tabla de ruteo entonces
// se cae en el caso raro y se busca dentro del neighborhood set
envía a $T \in L \cup R \cup M$
$shl(T, D) \geq l$,
$|T - D| < |A - D|$
}
}

El algoritmo arriba expuesto, describe como un mensaje, de manera inicial verifica si el ID destino se encuentra en el *Leaf Set* de ese nodo. Si si está, entonces el mensaje es enviado directamente al nodo destino, y se termina el proceso de ruteo; de otra manera, como el ID no está en *Leaf Set*, se revisa en la tabla de ruteo, buscando cual es el prefijo que mas se acerca al ID destino, entonces el mensaje es enviado a un nodo que comparta un prefijo común con el ID del mensaje en por lo menos la misma cantidad de dígitos que el nodo actual.

Este proceso se repite hasta que el mensaje llega a su destino.

De manera excepcional, se puede dar el caso (caso raro[6]) en que la entrada apropiada en la tabla de ruteo este vacía o el nodo asociado no este disponible, en cuyo caso el

[6]Previamente expuesto en el algoritmo de ruteo

mensaje es enviado a un nodo que comparta la misma cantidad de dígitos en el prefijo que el nodo actual este nuevo nodo será numericamente mas cercano en el ID que el nodo original.

Este simple procedimiento de ruteo siempre converge, pues en cada paso el mensaje se dirije a un nodo que le permita:

1. Compartir con la llave un prefijo mas largo que el nodo original
2. Comparta un prefijo tan largo como el del nodo original, pero que sea el mas cercano a la llave

Para ejemplificar el algoritmo de ruteo anteriormente expuesto, se puede ejemplificar en la figura 2.9.

Inicialmente, un mensaje es enviado a través del nodo "A" que con ID "10233102" este mensaje tiene como destino el nodo "B" con ID "102331100" , el mensaje sale del nodo origen (A) y viaja hacia el nodo con ID "10232230" debido a que el nodo destino no se encuentra dentro del *leaf set* del nodo origen aquí hay que hacer notar que este nodo comparte solo un dígito adicional de su ID con el que tiene marcado como destino el mensaje, en este punto el nodo revisa si en su *leaf set* o en su tabla de ruteo se encuentra el destino al cual el mensaje quiere llegar, al verifcar que no esta entonces manda al mensaje al nodo con ID "10233000" el cual comparte en su ID cuatro dígitos, nuevamente este nodo revisa en su *leaf set* , y al encontralo, entonces lo manda directo al destino terminando así el procedimiento de ruteo.

De donde tenemos que en la figura 2.9:

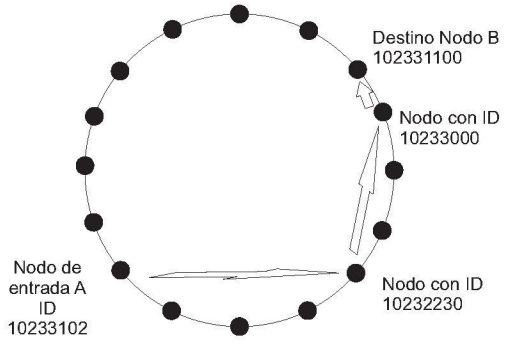

Figura 2.9: Ejemplo En el que un cliente A que busca información usando la DHT, este es un servicio que proveerá la P2P

Capítulo 3

Metodología

3.1. Supuestos

Se considera que se tiene una red de tipo P2P [1], la red se encuentra trabajando de forma normal, esto es:

- Tiene un espacio con capacidad para N posibles nodos.
- Tiene n nodos activos.
- Los mensajes son ruteados de forma adecuada, sin retrasos y a sus destinos.
- Todos los nodos se encuentran trabajando de forma adecuada.
- No existen cuellos de botella.
- Un nodo recibe una petición y este la contesta.
- No tiene tramos malos.
- Las consultas, tienen el mismo tamaño y siempre son los mismos paquetes por consulta.
- Existen E_c nodos maliciosos coalicionados, la información que ellos tienen, se toma como información correcta.
- La información que proveen los nodos maliciosos se considera correcta, ya que los nodos maliciosos no son identificados por la red.

3.2. Generación del ambiente

Para poder simular la red P2P es necesario generar el ambiente, éste se genera de la siguiente manera:

[1] Con las características de Pastry

1. Se Considera a $b = 2$ donde "b" es un parámetro de configuración.[2]
2. Se genera un ambiente con espacio para N elementos.
3. Se considera a N como la capacidad total de elementos que se pueden generar.
4. Se generan los n nodos que se van a utilizar.
5. Con base en el espacio creado anteriormente, y de manera aleatoria, a los nodos generados, se les asigna un "ID".
6. Cada "ID" es único.
7. Se ordenan los nodos en el "anillo" lógico con base en el "ID".
8. Se generan y llenan las tablas de ruteo siguiendo el siguiente procedimiento:
   a) Se genera el *leaf set* tomando los 4 nodos lógicos mas cercanos con valor inferior, (ID), y los 4 nodos lógicos mas cercanos con valor superior al nodo.
   b) Se genera la *tabla de ruteo*; se toman en cuenta el ID actual mas los elementos pertenecientes al *Leaf Set* para que no se repitan ni en el *Leaf Set* ni en la *tabla de ruteo*; la *tabla de ruteo*, se llena considerando los elementos del conjunto n de acuerdo a los prefijos compartidos marcados por las reglas de Pastry para cada renglón.
   c) Se verifica que ninguno de los elementos anexados, al *Leaf Set* ó a la *tabla de ruteo* se repita en cualquiera de las secciones.
   d) Para esta investigación, se considera que no existen entradas o salidas de nodos durante su ejecución como consecuencia, no se considera la implementación del *neighborhood set* y para el proceso de ruteo, en el supuesto de caer en el caso "raro" se considera como conjunto vacío considerandose unicamente al *leaf set* y a la*tabla de ruteo* para encontrar al nodo que tenga mayor coincidencias en el ID para el siguiente salto.

3.3. Proceso de Envío de Mensajes

Para el desarrollo del proceso de ruteo se siguieron las reglas que señala Pastry para su proceso de ruteo considerando la siguiente modificación al algoritmo original.

Se considera unicamente al *Neighborhood set* como un conjunto vacío, debido a que en la red simulada no se incorporan mas nodos, tampoco se considera que algún

[2]b es un parámetro de configuración utilizado por Pastry para la generación y dimensionamiento de las tablas de ruteo.

nodo salga, tampoco se tiene la posibilidad de dar de baja a algún nodo en la red; lo que trae como consecuencia que si se llegara a caer en el "caso raro" descrito en el algoritmo original de ruteo de Pastry solo se consideran los elementos de el *Leaf Set* y la *tabla de ruteo* para continuar con el proceso de ruteo.

Si se llegara a caer en el caso raro descrito anteriormente, entonces se utilizan para el proceso de ruteo los elementos del *Leaf Set* y de la *tabla de ruteo*, buscando al nodo que tenga mayor similitud en el prefijo del ID destino que lleva el mensaje para poder continuar con proceso de envío y recuperación de mensajes. Esta variación busca encontrar al elemento que se encuentre mas cercano al destino, considerando a esta modificación del algoritmo original como una modificación al algoritmo de ruteo de *PASTRY*.

3.4. Ataques

Una vez generado el ambiente y tomando en cuenta que la red se encuentra trabajando satisfactoriamente, se consideran las siguientes variables a analizar:

- Probabilidad de fracaso en el ruteo P_f, esta es definida como la cantidad de veces que un mensaje se perdió durante el proceso de ruteo, durante los experimentos que se corrieron y esta definido por

$$P_f = m_p/n \tag{3.1}$$

  Donde mp es la cantidad de mensajes perdidos durante la prueba, n es la cantidad de nodos generados.

- El número promedio de saltos esta definido por la cantidad de veces que un mensaje visitó algún nodo antes de llegar a su destino

3.4.1. Generación del ataque con Múltiples Nodos Coalicionados Coordinados

En este escenario de ataque se genera una sincronización entre los nodos coalicionados, $(E_1, \ldots, E_{nc})$, estos nodos inician el ataque de manera coordinada; adicionalmente, este ataque se considera un ataque "Inteligente" ó selectivo por que si el nodo que va a ser negado, (como consecuencia de el ataque), se encuentra en alguna de las secciones de la tabla de ruteo, de alguno de los nodos coalicionados entonces, el mensaje se descarta y se considera como ruteo fallido, (se genera la negación); sin embargo si el destino que se esta buscando, no se encuentra en alguna de las secciones de la tabla de ruteo,

el mensaje se envía al siguiente nodo que le corresponde de acuerdo con las reglas de *Pastry*; esto es con el objeto de que el ó los nodos maliciosos no sean descubiertos facilmente, pues no ejecutan acciones que podrían ser sospechosas, como por ejemplo, el que los nodos coalcionados, mandaran de manera aleatoria el mensaje a un sitio remoto de la red , o simplemente que los nodos negaran todo, con lo cual estos nodos estos nodos podrían ser descubiertos.

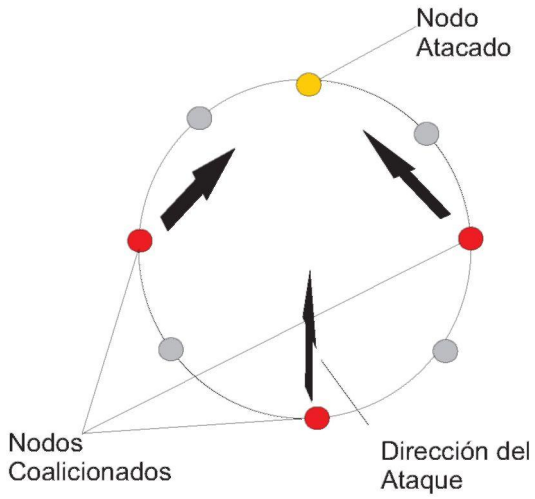

Figura 3.1: Esquematización de ataque con una coalición de nodos maliciosos coordinados en una red P2P

1. Inicialmente la red esta trabajando de manera "normal"
2. En un momento determinado se dispara un *trigger*, los nodos coalicionados, empiezan sus acciones maliciosas:
   - Niegan la existencia de un nodo en particular. como se puede observar en la figura 3.1
3. El nodo que va a ser atacado se obtiene de manera aleatoria.
4. Los nodos coalicionados $(E_1, \ldots, E_{nc})$, se obtienen de manera aleatoria.
5. Se considera que se tienen E_{nc} nodos maliciosos repartidos de manera aleatoria a lo largo de la red.

Capítulo 4

Resultados

Los resultados obtenidos son el producto del desarrollo de una simulación realizada en lenguaje PERL, en la que se generó un ambiente con capacidad $N = 5000$ elementos, de donde se crearon $n = 500$, considerando que se está utilizando unicamente el 10 % de la capacidad total del entorno con lo que además se respeta la cantidad de lugares ocupados que, de acuerdo con la teoría de las DHT recomienda, sea como máximo del 60 % dependiendo del algoritmo .

Además se consideró que utilizando solamente el 10 % de la capacidad total del espacio N se respeta la necesidad de aleatoriedad que pide Pastry para realizar la asignación de los ID siguiendo sus reglas.

Se realizaron las pruebas teniendo diferentes cantidades de nodos coalicionados, en cada prueba se enviaron 50,000 mensajes en cada una.

De las pruebas señaladas, se obtuvieron los siguientes resultados:

Cantidad de mensajes perdidos la tabla 4.1 nos muestra la cantidad de mensajes, (expresada en porcentaje), que se perdieron durante la realización de las diferentes pruebas, con base en la cantidad de nodos coalicionados como consecuencia del ataque de negación de existencia, además se muestra la probabilidad de fracaso en el ruteo expresada en porcentaje.

Como se puede observar en la tabla 4.1 el porcentaje de mensajes pérdidos es directamente proporcional al número de nodos coalicionados tal y como se aprecia en la figura 4.1, considerando diferentes cantidades de nodos coalicionados, habiendose tenido un ataque de negación de existencia, y siguiendo las reglas de Pastry para el procedimiento de ruteo.

Posteriormente se comparó la probabilidad de fracaso en el ruteo de los mensajes

Nodos coalicionados	P_f
0	0 %
1	0.182 %
2	0.348 %
5	1.02 %
10	1.99 %
20	3.78 %
50	9.99 %
100	20.02 %

Tabla 4.1: Muestra el porcentaje de mensajes perdidos con base en la cantidad de nodos coalicionados y la probabilidad de falla en el ruteo n=500 y N=50,000

durante las pruebas, contra la probabilidad de fracaso en el ruteo dado por las fórmulas:

1. Se considera a n como la cantidad de nodos existentes en la red.
2. Se considera E_{nc} como el número total de nodos coalicionados.
3. Se considera a F_f como la fracción de nodos en falla.
4. Se considera a σ como la probabilidad de rutear satisfactoriamente una petición entre dos nodos en funcionamiento, teniendo una cantidad E_{nc} de nodos coalicionados.

   De donde se tiene que

$$F_f = (E_{nc}/n) \tag{4.1}$$

$$\sigma = (1 - F_f)^{h-1} \tag{4.2}$$

   Donde

$$h = log_{2^b}(n) \tag{4.3}$$

   Se considera a P_f como la probabilidad de fracaso en el ruteo donde tenemos:

$$p_f = (1 - \sigma) \tag{4.4}$$

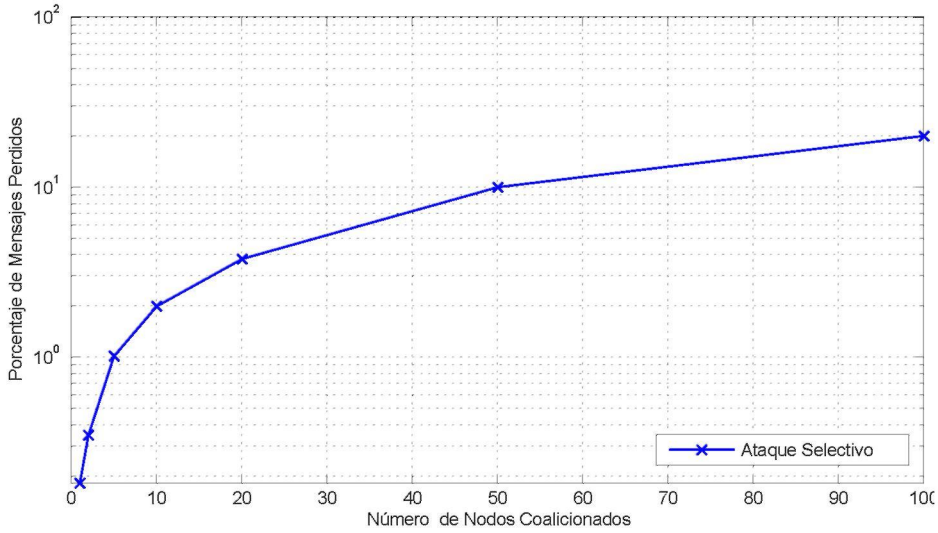

Figura 4.1: Gráfica que muestra el porcentaje de mensajes perdidos, contra el número de nodos coalicionados

dado en [16], pudiendose apreciar que la probabilidad de fracaso en el ruteo es menor en el algoritmo de ruteo con los nodos coalicionados utilizando las fórmulas que la teoría de Pastry señala; es importante mencionar que tal y como se expuso en la sección de "Proceso de Envío de Mensajes" el algoritmo utilizado en las pruebas fue el algoritmo modificado para esta investigación, en la que se encuentra actividad maliciosa con negación de existencia, mientras que en el algoritmo de Pastry los nodos no tienen actividad maliciosa sino solo fallas, con esa consideración se observa que en el algoritmo modificado de Pastry, la probabilidad de fracaso en el envío de mensajes es sensiblemente inferior en los resultados obtenidos por el Algoritmo original de Pastry al ser calculado por [16], como se puede apreciar en la figura 4.2.

Los resultados obtenidos del ataque selectivo en el que los mensajes son descartados de manera selectiva, se compararon contra otro ataque al cual se le denominó "ataque de fuerza bruta" del cual se puede observar una diferencia significativa en la pérdida de mensajes, esta comparación se realizó para poder observar la diferencia en la cantidad de mensajes que se pueden perder con cada tipo de ataque tal y como se observa en la tabla 4.2, dado que es un ataque muy evidente el ataque denominado "de fuerza bruta" es facilmente detectable, ya que en éste se niega siempre la existencia de los nodos, sin importar que el nodo destino este o no en su información de ruteo; por lo tanto al estar negando siempre los nodos se puede detectar facilmente, la figura 4.3 muestra porcentaje de mensajes perdidos comparados contra el ataque inteligente que

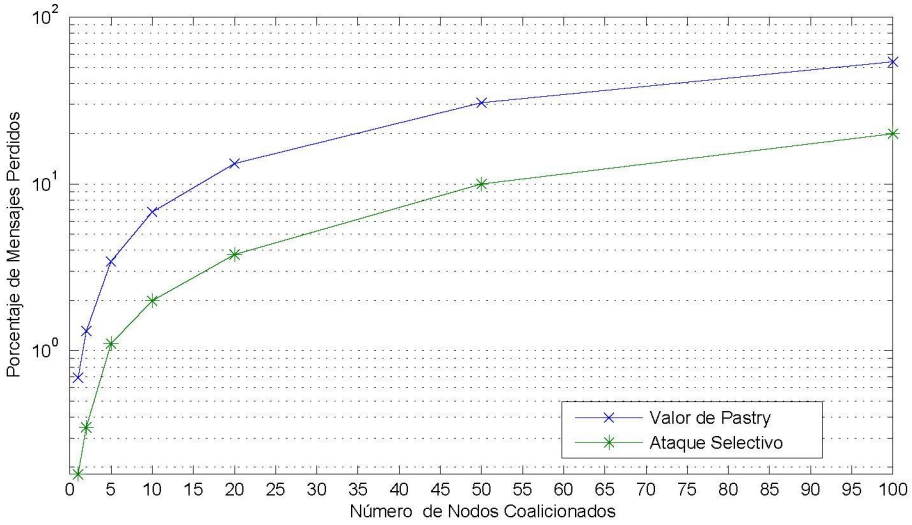

Figura 4.2: Gráfica que muestra la comparación entre la probabilidad de fracaso en el ruteo observado bajo un ataque selectivo y el esperado de acuerdo con Pastry

fue objeto de ésta investigación.

La figura 4.3 muestra el porcentaje de mensajes perdidos bajo un ataque inteligente y con un ataque de fuerza bruta.

Adicionalmente se generó un experimento de control para comprobar la aleatoriedad de los algoritmos, este experimento se desarrollo de la siguiente manera:

- Una prueba de ruteo en la que no existio ataque, el nodo destino es seleccionado de manera aleatoria.

Nodos Coalicionados	Ataque selectivo	% Ataque selectivo	Ataque fuerza bruta	% Fuerza bruta
0	0	0.00 %	0	0.00 %
1	91	0.182 %	460	0.92 %
2	174	0.348 %	923	1.846 %
5	508	1.02 %	2259	4.518 %
10	997	1.99 %	4435	8.87 %
20	1890	3.78 %	8415	16.83 %
50	4995	9.99 %	18702	37.404 %
100	10014	20.02 %	31402	62.80 %

Tabla 4.2: Tabla que muestra la diferencia en la cantidad de mensajes perdidos con dos ataques diferentes

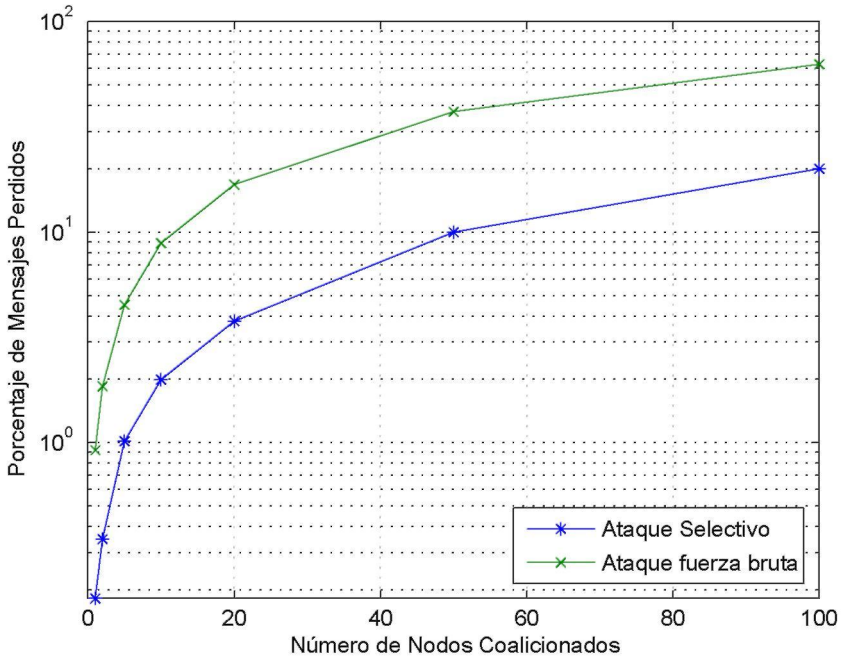

Figura 4.3: Gráfica que muestra la comparación de mensaje perdidos con un ataque de fuerza bruta y un ataque inteligente

4.1. Propuesta de Solución

Como resultado de los experimentos realizados se encontró que se pierden alrededor de un 20 % de los mensajes, cuando se tienen 100 nodos coalicionados de un total de 500, que representan el 20 % de el total de nodos en la red. Se propone una solución para reducir la cantidad de mensajes perdidos como consecuencia de el ataque coordinado con múltiples nodos coalicionados.

La solución propuesta, consiste en que si un mensaje llega a un nodo en el cual no tiene información del destino o se está negando a algún nodo, entonces este mensaje se regresa al nodo inmediato anterior de donde llego, ahi de manera aleatoria, utilizando la información de ruteo se escoge a algún nodo utilizando *Leaf Set* y se va hacia ese nodo, considerando unicamente que no sea el nodo del que llegó antes de este punto para evitar que el algoritmo se pueda ciclar; una vez que se selecciono un nuevo nodo se continua apartir de ese momento con las reglas de Pastry, el procedimiento descrito se puede observar en la figura 4.4.

Es importante señalar que el algoritmo de solución tiene la siguiente limitante,

y ésta es la siguiente: Cuando se encuentra con un nodo coalicionado regresa al nodo origen y se selecciona otro nodo que se encuentre en el *leaf set* de ese nodo pero si en el segundo salto cae en otro nodo coalicionado, el mensaje se pierde; esta limitante es con el objetivo, de que no se genere tráfico innecesario en la red, y pueda se simular el comportamiento de tráfico que se podría tener en un sistema de distribucón amplia, ya que si el mensaje esta brincando de un nodo a otro se puede corromper en el camino y se genera mas tráfico.

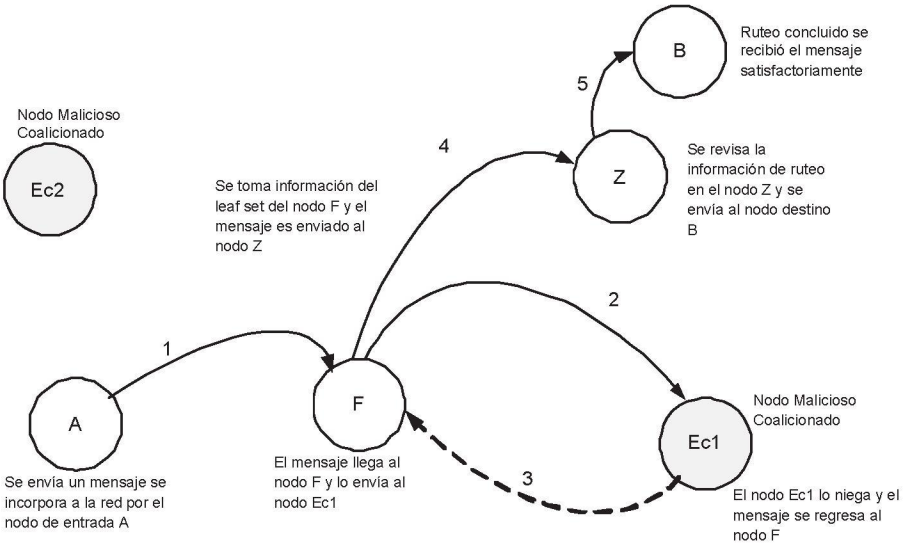

Figura 4.4: La figura muestra la propuesta de solución para el algoritmo de ruteo

El resultado de los experimentos con la propuesta de solución muestra la cantidad de mensajes perdidos y se pueden apreciar en la tabla 4.3

La figura 4.5 muestra el porcentaje de mensajes perdidos, cuando se recibe un ataque de fuerza bruta, cuando se recibe un ataque normal y cuando se aplica la solución propuesta

Adicionalmente para verificar que la solución propuesta no fuera a ser contra producente, por ejemplo generando un número excesivo de saltos y que como consecuencia de eso se genera mas tráfico en la red, se contaron la cantidad de saltos promedio que se tienen durante un ataque inteligente y después cuando se aplica el algoritmo con la solución propuesta los resultados del conteo de saltos se pueden apreciar en la tabla 4.4.

Nodos Coalicionados	Porcentaje de Mensajes Perdidos
0	0 %
1	0.008 %
2	0.0.16 %
5	0.032 %
10	0.1 %
20	0.288 %
50	1.31 %
100	4.56 %

Tabla 4.3: Tabla que muestra la cantidad de mensajes perdidos aplicando la solución propuesta

Nodos Coalicionados	Núm. de saltos con ataque	Núm de saltos con solución
0	2.83	2.83
1	2.8298	2.8513
2	2.8385	2.9127
5	2.8346	3.0192
10	2.8384	3.0802
20	2.8389	3.2438
50	2.8386	3.5274
100	2.84	3.7219

Tabla 4.4: Tabla que muestra la cantidad de saltos promedio bajo un ataque inteligente y la solución propuesta

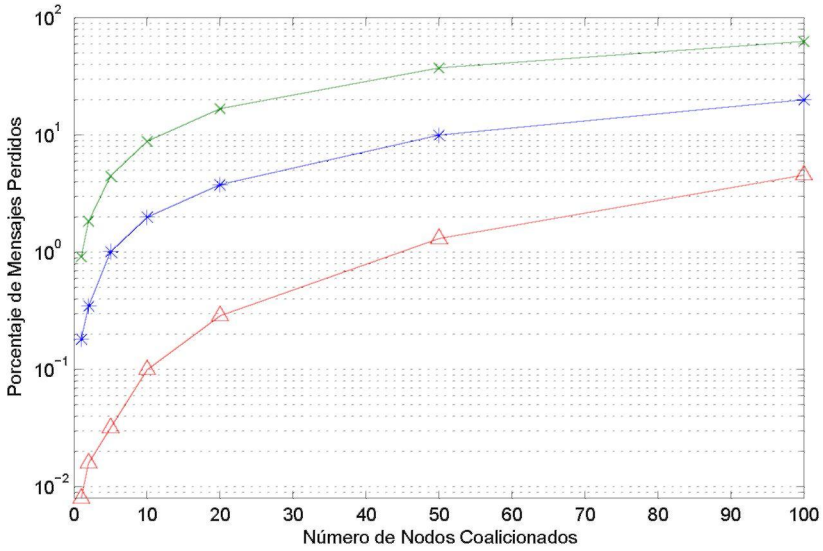

Figura 4.5: Gráfica que muestra el porcentaje de mensajes perdidos, contra el número de nodos coalicionados, mostrando un ataque de fuerza bruta, un ataque normal y la solución propuesta

Capítulo 5

Conclusiones

Con base en los experimentos realizados se puede concluir que las redes P2P son una alternativa viable para aplicaciones de almacenamiento y recuperación de mensajes.

La utilización de sustratos especializados para el almacenamiento, envío y recuperación de mensajes como Pastry, es una alternativa de solución a los problemas almacenamiento central, que se tienen en las redes del tipo Cliente- Servidor, ya que con las características de las redes P2P, la manipulación de la información es mas eficiente.

El balanceo de la red es adecuado, la aleatoriedad que se genera en el ambiente es adecuada, se observa en los resultados que la pérdida de mensajes es en la misma proporción que el porcentaje de nodos coalicionados.

La propuesta de solución es muy eficiente ya que el efecto del ataque de negación de existencia se reduce a un 4.56 % cuando se tienen 100 nodos coalicionados, lo cual resulta muy satisfactorio, ya que inicialmente con el algoritmo diseñado sin la propuesta de solución, se perdían un 20.02 % de los mensajes enviados, esta reducción de mensajes perdidos, es una reducción significativa.

La propuesta de solución es eficiente, ya que no se aumenta de forma significativa la cantidad de saltos que el mensaje debe de dar para poder llegar a su destino cuando se encuentra con un nodo coalicionado, ya que aumenta de 2.81 saltos promedio, a 3.7219 saltos promedio, que un mensaje debe de dar, para poder llegar al destino cuando se tienen 100 nodos coalicionados, pudiendose observar que este aumento en la cantidad de saltos promedio es mínima y no afecta el desempeño de la red, considerando la cantidad de nodos coalicionados.

Este tipo de redes podría servir para montar aplicaciones de amplia distribución como DNS, sin embargo hay que establecer la distribución de la información, cuando se quiera utilizar en aplicaciones donde los recursos sean utilizados de manera no uniforme.

5.1. Trabajo Futuro

Para futuras investigaciones se debe realizar una prueba de comportamiento del ambiente incluyendo el "neighborhood set" para poder observar el comportamiento del algoritmo de ruteo cuando se tiene completa la tabla de ruteo, y poder observar la cantidad de saltos promedio que se dan cuando esta el neighborhood set, además se deben de realizar pruebas en las que se contemplen entradas y salidas de nodos, tanto aleatorias como definidas.

Se deben realizar investigaciones con otros tipos de ataques, también se debe comparar el desempeño de la red, cuando se realizan diferentes tipos de ataques y de diferente intensidad.

Desarrollar un algoritmo que mejore el desempeño, como por ejemplo buscar el nodo mas cercano al destino, lo que implicaría una modificación al algoritmo de Pastry.

Determinar el desempeño de una red con fallas comunes, tales como desconexión imprevista de nodos, errores en las tablas de ruteo, entre otras.

Bibliografía

[1] A. B.; Dreger, H.; Feldmann, A.; Predicting the DNSSEC overhead using DNS traces;Information Sciences and Systems, 2006 40th Annual Conference on 22-24 March 2006 Page(s):1484 - 1489

[2] A. Passarella, Franca Delmastro, Marco Conti; XScribe: a stateless, cross-layer approach to P2P multicast in multi-hop ad hoc networks; Proceedings of the 1st international workshop on Decentralized resource sharing in mobile computing and networking MobiShare '06; September 2006

[3] A. Rowstron and Peter Druschel; Storage management and caching in PAST, a large scale, persistent peer to peer storage utility; In Proc ACM SOS'P 1; Banff Canada Oct 2001.

[4] A. Rowstron and Peter Druschel; Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems; In Proc. IFIP/ACM Middleware 2001, Heidelberg, Germany; November 2001.

[5] A. Friedlander, Allison Mankin, W. Douglas Maughan, Stephen D. Crocker; DNSSEC: a protocol toward securing the internet infrastructure; Communications of the ACM, Volume 50 Issue 6; June 2007

[6] Ben Y. Zhao, John D. Kubiatowicz, and Anthony D. Joseph.; Tapestry: An infrastructure for fault-tolerant wide-area location and routing; Technical Report UCB//CSD-01-1141, U. C. Berkeley, April 2001.

[7] D. Ervin Khuth; The art of computer programming; Stanford University Addison Wesley; Vol 3 sorting and searching; U.S.A. 1973.

[8] E. Damiani, Stefano Paraboschi, Pierangela Samarati, Fabio Violante; Peer to peer networks: A reputation-based approach for choosing reliable resources in peer-to-peer networks; Proceedings of the 9th ACM conference on Computer and communications security CCS '02; November 2002

[9] G. Alexis; DNSSEC Operational Impact and Performance;Computing in the Global Information Technology, 2006. ICCGI '06. International Multi-Conference on Aug. 2006 Page(s)55 - 63

[10] I. Stoica, R. Morris, D. Karger, M Frans H. Balakrishnan; Chord: a scalable Peer-to-Peer lookup service for Internet applications; SIGCOMM 01 ; ACM; San Diego California; 2001

[11] J. Kyeong Kim a, Hyea Kyeong Kim a, Yoon Ho Cho; A user-oriented contents recommendation system in peer-to-peer architecture; Expert Systems with Applications 34 (2008) 300-312.

[12] K. Hui, John C.S. Lui, David K.Y. Yau; Small-world overlay P2P networks: Construction, management and handling of dynamic flash crowds; computer networks num 50 2006.

[13] L. Liu, N. Antonopolus, S. Mackin; Fault tolerant peer-to-peer search on small-world networks; Future Generation Computer Systems; vol 23; march 2007; 921-931.

[14] L. Zambenedetti Granville, D. Moreira da Rosa, C. Melchiors, M. J. Bosquiroli Almeida, and L. M. Rockenbach Tarouco; Managing Computer Networks Using Peer-to-Peer Technologies; IEEE Communications Magazine ; October 2005

[15] M. Castro and Barbara Liskov. Practical byzantine fault tolerance. In Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDl'99), New Orleans, Louisiana, February 1999.

[16] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. Wallach. Secure Routing for Structured Peer-to-Peer Overlay Networks. Symposium on Operating Systems Design and Implementation, Boston MA, Dec 2002.

[17] M. Naor and Udi Wieder. Novel Architectures for P2P Applications: the Continuous-Discrete Approach. Proc. SPAA, 2006

[18] Ñ.Saxena, G. Tsudik, J. Hyun Yi.; Threshold cryptography in P2P and MANETs: The case of access control; Computer Networks vol 51; 2007

[19] P. Fältström, Daniel Massey, Vasileios Pappas, Lixia Zhang; Distributed DNS troubleshooting; Proceedings of the ACM SIGCOMM workshop on Network troubleshooting: research, theory and operations practice meet malfunctioning reality NetT '04; September 2 004 .

[20] R. Venugopalan, Emin Gün Sirer; The design and implementation of a next generation name service for the internet; ACM SIGCOMM Computer Communication Review; Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications SIGCOMM '04, Volume 34 Issue 4; August 2004

[21] S. Andr outsellis-Theotokis, Diomidis Spinellis; A survey of peer-to-peer content distribution technologies; ACM Computing Surveys (CSUR), Volume 36 Issue 4; December 2004.

[22] S. Ariyapperuma, C.J Mitchell; Security vulnerabilities in DNS and DNSSEC; Availability, Reliability and Security, 2007; The Second International Conference on 10-13 April 2007 Page(s):335 - 342; ARES 2007

[23] S. Iyer, Antony Rowstron, Peter Druschel; Squirrel: a decentralized peer-to-peer web cache; Proceedings of the twenty-first annual symposium on Principles of distributed computing PODC '02 Publisher: ACM Press; July 2002

[24] S. Ratnasamy, Paul Francis, Mark Handley, Richard Karp, Scott Schenker; A scalable content-addressable network;; ACM SIGCOMM Computer Communication Review , Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications SIGCOMM '01, Volume 31 Issue 4; August 2001

[25] Zhang Rongmei and Charlie Hu; Borg: A Hybrid protocol for scalable application level multicast in peer to peer networks; Nossdaw 03; Monterey California; June 03.