International Journal of Engineering Research and Development
e-ISSN : 2278-067X, p-ISSN : 2278-800X, www.ijerd.com
Volume 2, Issue 6 (August 2012), PP. 56-63

Secured Query Registration and Searching for P2P Network

Kodumoori Suma[1], Kothapally Kiran[2], Kasula Saidulu[3], Dr. G. Manjunath[4]
[1,2,3]First Student, Dept. of IT, JNIT, Hyderabad.
[4]Head, Dept. of CSE & IT, JNIT, Hyderabad

Abstract*—Now a days large research groups and alike entities uses P2P system for keeping track of research activities and downloading the files. But P2P is insecured as the communication takes place between the end pair of nodes and there is no control or security of the sessions. P2p is a system of acquiring data directly from the clients using a discovery process monitored by the server. As such, in such a system only information about the data and the nodes are maintained at the server and the communication is in Peer to peer manner between the clients.In a network where data is being changed and new data is released continuely, tracking, and managing the repository is a huge task. Updating the end peers about the availibility of the new data or changes in the existing content is time consuming. Hence in this work we propose a unique system whereby the clients notifies a P2P server about the changes in their file release or about the new files. Server handles the query from existing peer. If the queries are unanswered, then the server stores the queries in an intermediate node called beacon. Often this approch leads to data leakage. Therefore we use BLS for end to end authentication scheme between the server and the query generating client and the queries are digitally signed by the private key of the client and stored.When new data is available, the registered queries are decrypted with the public key stored at the P2P server and clients are alerted about the new data. If the new data contains an Item of interest, it is sent to the client as announcement.

***Keywords*—P2P Network ,Query Registration, Query Processing, Searching in P2P, Maximum Coverage, BLS.**

I. INTRODUCTION

In recent years unstructured peer-to-peer (P2P) systems have evolved as a popular paradigm for content/resource distribution and sharing [1, 6]. Owing to the simplicity of design and flexibility towards transient node population, the real-world P2P systems are invariably unstructured. However, most unstructured P2P content distribution systems only support a very simple model for data sharing and discovery called the ad hoc query model. A peer that is interested in discovering data items initiates a query with a set of search parameters, which is then circulated among the peers according to the specific query forwarding mechanism employed by the network. A peer receiving a query responds to the query initiator, if it has any content satisfying the search criterion. Once a query has been processed at a node, it is removed from the local buffers (some systems cache recently received queries, but for a very short duration and in an ad hoc fashion). Therefore, a query exists within the P2P network only until it is propagated to various nodes and processed by them (or for a short duration thereafter, if the network employs caching). Once a query completes its circulation, the system essentially forgets it.

While the ad hoc query model for data discovery is essential for P2P content distribution networks, it suffers from two serious limitations. First, due to its very nature, an ad hoc query is only capable of retrieving content that exists in the P2P network during the time period when it is actively propagated and processed in the network. Further, an ad hoc query can never reach a peer that joins the network after the query has completed its circulation, and hence cannot discover matching data-items on the new peer. In this scenario, the only way for a peer to discover newly added data-items would be to repeatedly issue the same query, thereby imposing unnecessary overheads on the network. Second, the ad hoc query model provides no support for peers to advertise or announce the data-items they own to other interested peers. Such capabilities are important for P2P communities where peers trade content.

These shortcomings limit the utility of the ad hoc query model for several advanced collaborative applications, such as a community of researchers sharing their recent research results or a community of amateur musicians and their patrons who are interested in buying the music produced by the musicians. In applications such as these, participating peers would not only be interested in searching for existing content, but would also want to be pro-actively informed when content matching their interests is added to the network. Further, some communities also need a mechanism through which peers can advertise their content to other interested peers. Blind broadcast of advertisement would not only result in high overheads, but could also annoy participants who would be receiving large numbers of advertisement about data-items that they are not interested in.

An approach that can partially mitigate these limitations would be to implement a publish-subscribe (pub-sub) system on top of the unstructured overlay network. A generic pub-sub system enables its users to register subscriptions expressing their interests and to announce the occurrence of certain events by publishing them. The pub-sub system matches incoming announcements to the existing subscriptions and notifies the users that have registered the matching subscriptions. An important point to note is that the pub-sub systems attempt to provide guaranteed notification service (although it might not be possible always due to system failures). Researchers have studied the problem of implementingP2P-based pub-sub systems on unstructured overlay networks[7, 17]. However, most of these systems require the underlying P2P networks to be organized according to specific architectures, and hence they cannot be used in generic overlays. Many of these systems also require the peers to maintain intricate index structures which add significant complexity to the design of the P2P network. This additional complexity can adversely affect the flexibility, efficiency, and scalability of the unstructured P2P system. Furthermore, it also makes the design, implementation, and management of P2P content distribution networks harder.

II. RELATED WORK

The work presented in this paper is primarily related to two fields, namely P2P networks [6, 9, 14] and publish subscribe systems (event-delivery systems) [3,5], both of which have been very active areas of research in the past few years. Pub-sub systems can be classified into two broad categories: (1) topic-based-wherein users join specific topic groups in which all the messages related to the topic are broadcast; and (2) content-based-wherein users specify their interests through predicates. With the aim of enhancing scalability, efficiency and scalability several distributed pub-sub systems have been proposed [3, 5]. Recently, P2P computing models have been utilized for this purpose. Researchers have explored two strategies for constructing P2P-based pub-sub systems, namely (a) adopting a structured P2P network like Chord [15] or CAN [13] as the underlying substrate, and utilizing its indexing schemes for mapping subscriptions and events to nodes of the P2P systems [10, 16]; (b) organizing the nodes of the P2P system into specialized topologies and/or embedding application specific distributed index structures within nodes of the P2P network [7, 17, 19]. The Sub-2-Sub system [17] organizes the peers into clusters using an epidemic-style algorithm such that nodes with similar subscriptions are put into the same cluster. The publisher of an event joins the corresponding cluster and disseminates the event to the cluster members. The our system differs from the above works in terms of motivation, goals and approach. The goal of the above systems is to improve the various performance parameters of pub-sub systems and they use P2P-based techniques as a means towards this end. In contrast, our goal is to enhance the P2P data sharing systems, and continuous queries (that bear similarity to pub-sub model) is a means towards that end. Second, the above pub-sub systems cannot be implemented on top of generic P2P networks; they need specialized overlays (specific topologies and/or indexing mechanisms). Contrastingly, our system does not need any complex distributed indexing structures, nor does it impose any topological constraints on the overlay network. Finally, the above systems are essentially pub-sub systems, and hence guaranteed notification is one of their design goals. Our system provides best-effort notification, which is in tune with design principles of unstructured P2P networks. P2P-DIET [11] supports both ad-hoc and continuous queries, however, it assumes a super peer-based overlay.

In short, the work presented in this paper has several unique aspects, and it addresses an important problem in the area of P2P data sharing systems. Along with the same, the query registration at the intermediate nodes leaks information about the node generating query and makes the identity of the client as vulnerable. Therefore the paper also introduces a BLS based scheme for securing the queries by digitally signing them.

III. OUR APPROACH

A. Problem Formation

Peer to peer networks uses different computers or peers to share the files amongst themselves rather than keeping the files in a single server. Hence same file may be downloaded from different nodes in a peer. But due to non-central nature of the communication, such network looses control and finding out group of nodes relevant to data sharing is low. Therefore here we propose a system to

1) Index the files at the server
2) A search engine to respond to the queries.
3) Maximum neighborhood based beacon selection for registering queries
4) Clients generates the queries which are digitally signed by their private Key. It is decrypted at the server using the publick key and re encrypted with the same if the query needs to be registered. Hence exposure of the clients at the beacons is NULL.
5) Chunking based parallel file download for load balancing
6) Achieve scalability for new information.

B. Proposed System

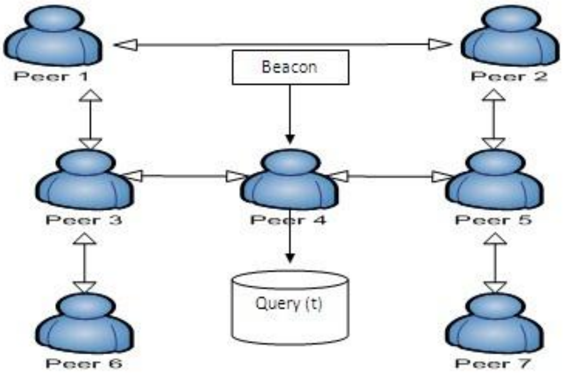

Fig. 1 Pure Decentralized P2P Content-Sharing Architecture

The above fig1shows decentralized P2P content sharing system where a user register a query at peer4 called beacon node at time t. Once the data is available by some client, the beacon announces the same to the requesting client. Further in present system, a client can download only the information that is available at that instance of time. But in the proposed system, a client can download the information after the information is added at a later time. Further the system proposes a chunking based technique with maximum coverage. Generally in P2P system a client downloads the information from the best possible path, so the coverage is least. Hence if any intermediate client seeks the information in between a session, then without having to re-establish a fresh session, nodes can join the existing transmission and start downloading the needed chunks.

C. Concepts and Notations

Consider an unstructured P2P system comprising of peers (P0,P1,…..PN-1). Let (L0,L1……LM-1) represent the logical links (connections) in the network. For simplicity, we assume that the links are bidirectional. Two peers Pi and Pj are said to be neighbors of each other if there exists a link Lv = (Pi; Pj) connecting them.

We assume that each data item Dr in the system has associated metadata (represented as MData(Dr)) that describes it. In the current context, the metadata is a list of keywords describing the data item.

Continuous query is the means through which a peer can register its interests with the network. A continuous query, represented as Q =(SID; Predicate; V Time), is essentially a tuple of three components, namely, source ID (SID) , query predicate (Predicate) and validity time (V Time). The source ID uniquely identies the peer issuing the query. The query predicate is the matching condition of the query, and is used by the source peer to specify its interests. In general, the predicate can be of any form such as range predicates or even a regular expression. We assume that the predicate is a list of keywords describing the content the source peer is interested in. Validity time (V Time) represents the time until which the source node is interested in receiving notifications. Peers announce their new data items through announcements.

An announcement is represented as Ad =(AID;MData). The announcing peer ID (AID) identifies the advertising peer and the metadata (MData) is the metadata of the content being advertised. A data item Dr (and analogously its announcement) is said to match a continuous query Qm, if Dr's metadata contains all the keywords in Qm's predicate. We use the word query and continuous query[8,12] interchangeably.

D. System Design

Algorithm: The proposed system is explained as bellow.

* Generate Random set of nodes.
* Index a set of information-TF-IDF(term frequency–inverse document frequency).
* Randomly distribute the information to the Nodes.
* One or more clients query for a specific information. The queries are encrypted using BLS scheme with their private keys.
* Search and locate the clients where the information is available.
* Request the Seeder for downloading
* divide the data into equal pieces equivalent to number of lecher.
* Data is sent to the clients through a route from seeder to lecher. Each lecher than connects with each other and downloads the missing chunks.(Maximum Coverage Technique).
* If there is no data available for a query, it is registered at a node called a beacon node by first reencrypting the data with the public key of the server.
* Once the data is available, the information is announced to the lechers and the downloading begins as above.
* After certain time period the query is expired to maintain the integrity of the freshness of the information .
* For query processing, exponential time-maximum likelihood estimation query registration and forwarding is used.

IV. METHODOLOGY

A. BLS Pairing Function

A gap group is a group in which the computational Diffie–Hellman problem is intractable but the decisional Diffie–Hellman problem can be efficiently solved. Non-degenerate, efficiently computable, bilinear pairing functions permit such groups.

Let $e\colon G \times G \rightarrow G_T$ be a non-degenerate, efficiently computable, bilinear pairing function where G, G_T are groups of prime order, r. Let g be a generator of G. Consider an instance of theCDH problem, g,g^x , g^y. Intuitively, the pairing function e does not help us compute g^{xy} , the solution to the CDH problem. It is conjectured that this instance of the CDH problem is intractable. Given g^z , we may check to see if $g^z = g^{xy}$ without knowledge of x, y, and z, by testing whether $e(g^x, g^y) = e(g, g^z)$ holds.

By using the bilinear property $x + y + z$ times, we see that if $e(g^x, g^y) = e(g,g)^{xy} = e(g,g)^z = e(g, g^z)$, then since G_T is a prime order group, $xy = z$.

A signature scheme consists of three functions, *generate*, *sign*, and *verify*

1) Key generation

The key generation algorithm selects a random integer x in the interval [0, $r - 1$]. The private key is x. The holder of the private key publishes the public key, g^x .

2) Signing

Given the private key x, and some message m, we compute the signature by hashing the bitstring m, as $h = H(m)$. We output the signature $\sigma = h^x$.

3) Verification

Given a signature σ and a public key g^x , we verify that $e(\sigma, g) = e(H(m), g^x)$.

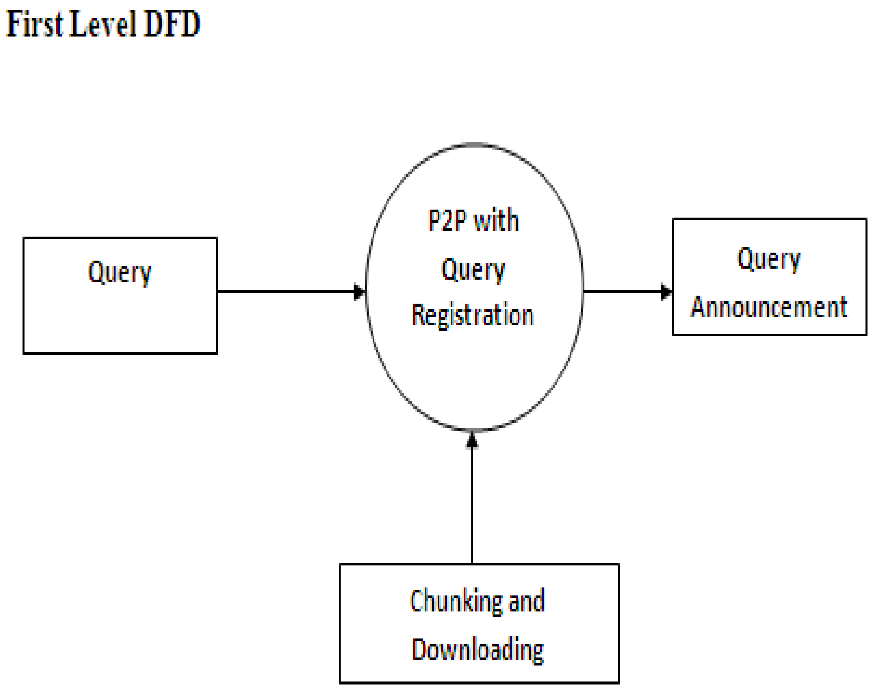

Fig. 2 First Level Data Flow Diagram

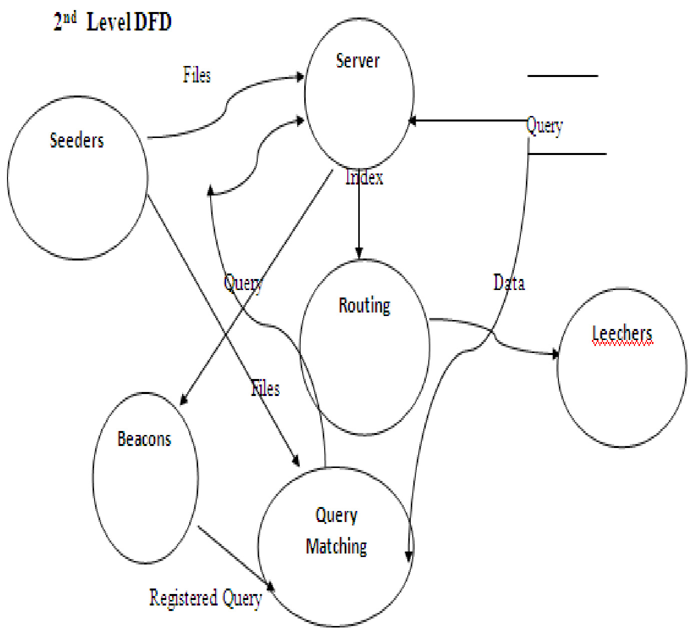

Fig. 3 Second Level Data Flow Diagram

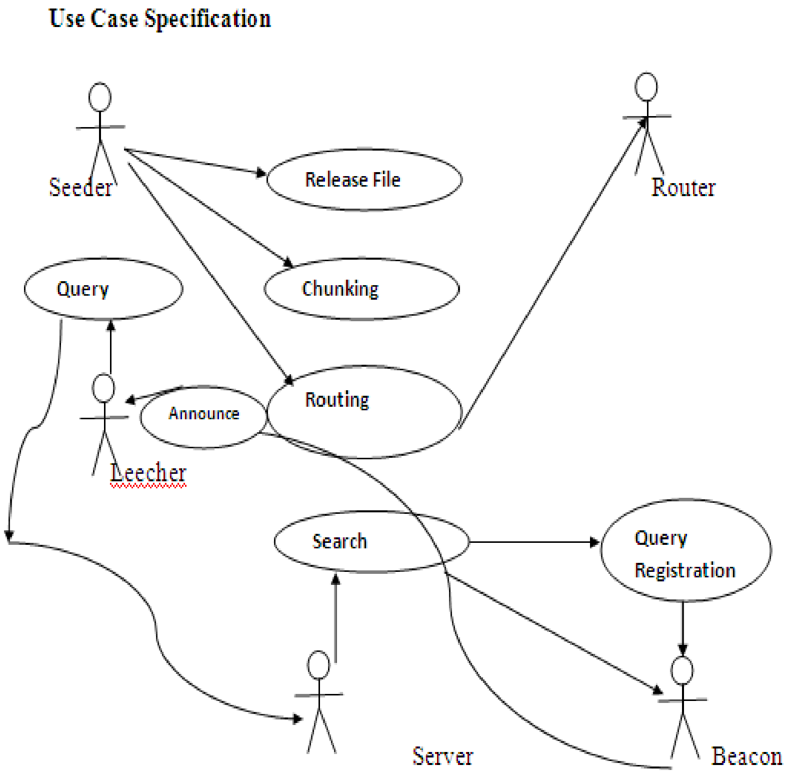

Fig. 4 Use Case Specification Diagram

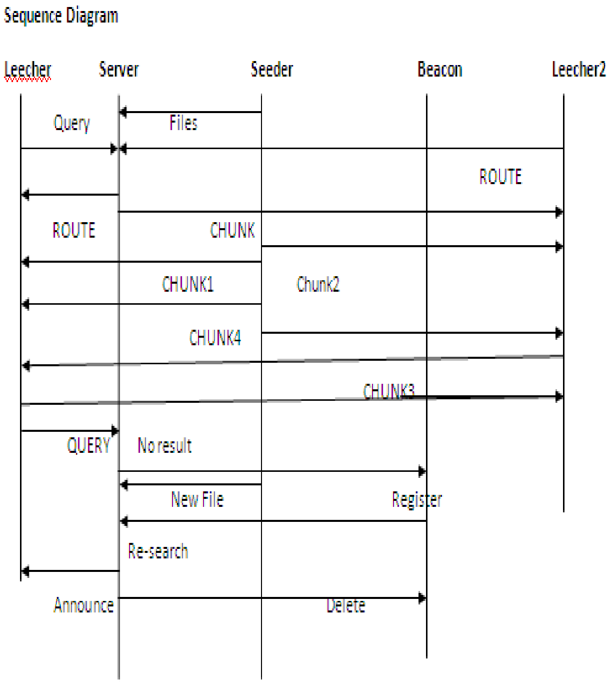

Fig.5 Sequence Diagram

V. RESULTS

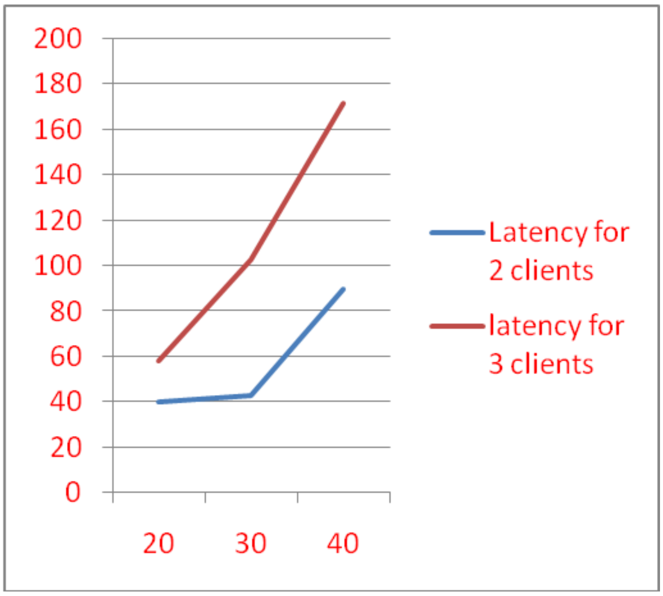

Fig. 6 Number of Node v/s Latency

Latency is defined as the end to end delay for a node to acquire an entire file through collecting of chunks from seeder and through transformed seeders. The graph shows that the latency of the system depends upon the leechers rather than the network size. for limited leechers, the transmission time is low and for the higher leechers, the same is increased.

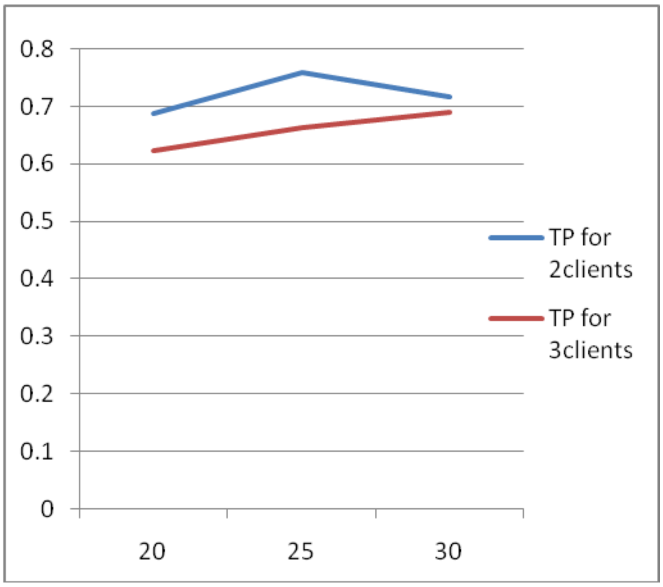

Fig. 7 Number of Nodes v/s Throughput.

The performance graph shows that the chunking and parallel downloading and maximum coverage routing helps in enhancing the throughput. Generally throughput decreases significantly in P2P system. But in the current process, throughput is increased with increase of nodes which suggest that as the network grows, availability of the data also increases which is most desirable property of the P2P system.

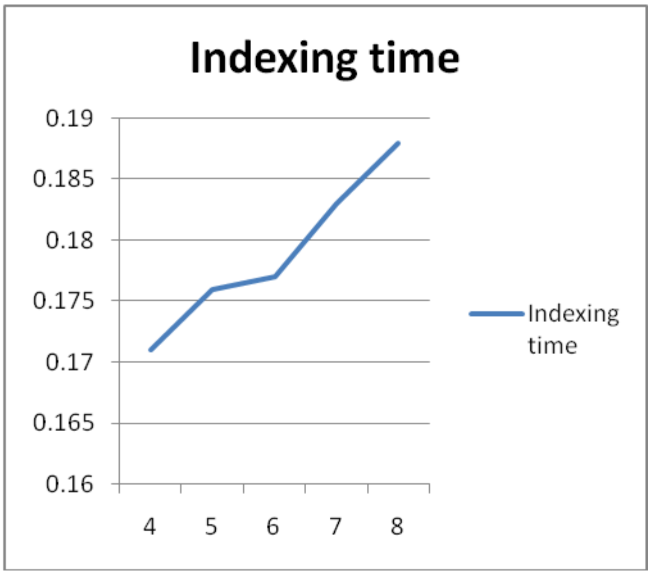

Fig. 8 Number of Data Items v/s the indexing time.

This experiment was conducted by considering different length of text documents. The indexing is considered as TF-IDF score of the documents. Indexing time increases with increase of number of files and is independent of the file size. Therefore as the network grows, indexing the entire set consumes time. Hence the concept of query processing is used which eliminates the indexing for every query. Once a query is unanswered the query is stored. Hence only re indexing is needed once a new data is made available which is relevant to the query.

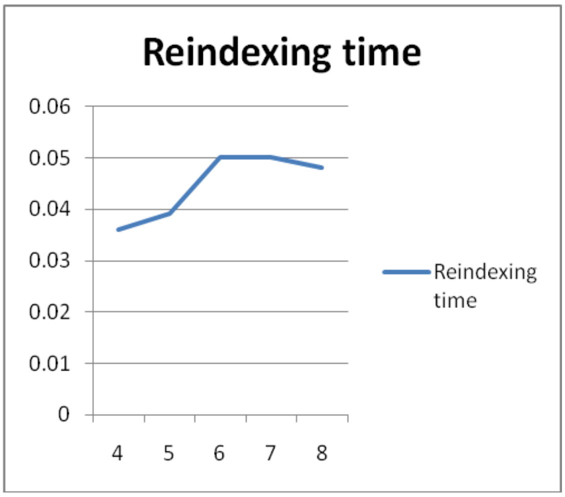

Fig. 9 Number of Files v/s Reindexing time

The performance graph clearly explains the utility of the technique. Once a new data is matched with query it is reindexed. Number of reindexing iterations are limited. Hence the system takes lesser time for announcement of new files.

VI. CONCLUSIONS

Peer-to-peer systems have become a popular media for sharing large amount of information among millions of users. While previous research efforts are focusing on supporting search in P2P systems, obtaining hidden and valuable knowledge from these data through data mining techniques is essential for scientific findings and many other applications. In this work, we investigate searching and query registration for fast information announcement to the nodes. We provide complexity analysis on the transmission incurred by the system. The analytic result indicates that proposed system can efficiently mitigate data to the nodes seeking the information. The system can be further improved by adopting passive replication technique.

REFERENCES

[1]. Gnutella P2P Network. www.gnutella.com.

[2]. Kazaa P2P Network. www.kazaa.com.

[3]. G. Banavar, T. Chandra, B. Mukherjee, J. Nagarajarao, R. E. Strom, and D. C. Sturman. An Efcient Multicast Protocol for Content-Based Publish-Subscribe Systems. In Proceedings of ICDCS 1999, 1999.

[4]. N. Bisnik and A. Abouzeid. Modeling and analysis of random walk search algorithms in P2P networks. In Proceedings of HOT-P2P, 2005.

[5]. A. Carzaniga, D. S. Rosenblum, and A. L.Wolf. Design and evaluation of a wide-area event notication service. ACM Transactions on Computer Systems, 19(3):332–383, 2001.

[6]. Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making Gnutella-like P2P Systems Scalable. In Proceedings of ACM SIGCOMM 2003, 2003.

[7]. P. Chirita, S. Idreos, M. Koubarakis, and W. Nejdl. Publish/Subscribe for RDF-based P2P Networks. In Proceedings of the 1st European Semantic Web Symposium, May 2004.

[8]. Lakshmish Ramaswamy, Member, IEEE, and Jianxia Chen, Student Member, IEEE. The CoQUOS Approach to Continuous Queries in Unstructured Overlays.IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL.23, NO. 4, April 2011

[9]. S. Androutsellis-Theotokis and D. Spinellis. A Survey of Peer-to-Peer Content Distribution Technologies. ACM Comput. Surv., 2004.

[10]. A. Gupta, O. D. Sahin, D. Agrawal, and A. E. Abbadi.

[11]. Meghdoot: content-based publish/subscribe over P2P networks. In Proceedings of Middleware 2004, 2004.

[12]. S. Idreos, M. Koubarakis, and C. Tryfonopoulos. P2P-DIET: One-Time and Continuous Queries in Super-Peer Networks. In Proceedings of EDBT, 2004.

[13]. L. Ramaswamy, J. Chen, P. Parate, and A. Meka. Lightweight Support for Continuous Queries in Unstructured Overlays. Technical report, The University of Georgia, 2006.

[14]. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A Scalable Content-Addressable Network. In Proceedings of ACM SIGCOMM 2001, Aug 2001.

[15]. P. Reynolds and A.Vahdat. Efcient peer-to-peer keyword searching. In Proceedings of Middleware 2003.

[16]. I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In Proceedings of ACM SIGCOMM 2001, Aug 2001.

[17]. P. Triantallou and I. Aekaterinidis. Content-Based Publish-Subscribe Over Structured P2P Networks. In Proceedings of the International Workshop on Distributed Event-Based Systems (DEBS), 2004.

[18]. S. Voulgaris, E. Riviere, A.-M. Kermarrec, and M. van Steen. Sub-2-Sub: Self-Organizing Content-Based Publish Subscribe for Dynamic Large Scale Collaborative Networks. In Proceedings of the 5th international workshop on peer-topeer systems, Feb 2006.

[19]. B. Yang and H. Garcia-Molina. Improving search in peer to peer systems. In Proceedings of ICDCS 2002.

[20]. C. Zhang, A. Krishnamurthy, and R. Wang. Combining fexibility and scalability in a peer-to-peer publish/subscribe system. In Middleware 2005.