Super Peer Deployment in Unstructured Peer-to-Peer Networks

R.Venkadeshan,
Assistant Professor / CSE Department,
Chettinad College of Engineering & Technology,
Karur, Tamilnadu, India

M.Jegatha,
Assistant Professor / CSE Department,
Chettinad College of Engineering & Technology,
Karur, Tamilnadu, India

Abstract: **Two-layer hierarchy unstructured peer-to-peer (P2P) systems, comprising an upper layer of super-peers and an underlying layer of ordinary peers, are commonly used to improve the performance of large-scale P2P systems. However, the optimal super-peer network design involves several requirements including super-peer degree, network diameter, scalability, load balancing, and flooding performance. A perfect difference graph has desirable properties to satisfy the above design rationale of super-peers overlay network. This paper proposes a two-layer hierarchical unstructured P2P system in which a perfect difference graph (PDG) is used to dynamically construct and maintain the super-peer overlay topology. In addition, the broadcasting performance of the P2P system is enhanced through the use of a PDG-based forwarding algorithm which ensures that each super-peer receives just one lookup query flooding message. The theoretical results show that the proposed system improves existing super-peer hierarchical unstructured P2P systems in terms of a smaller network diameter, fewer lookup flooding messages, and a reduced average delay and the experimental results show that the proposed two-layer hierarchy P2P system performs very well in the dynamic network environment**.

Keywords-Super Peers, forwarding Algorithms, Peer-to-Peer.

I. INTRODUCTION

Peer-to-peer (P2P) overlay networks are massively-distributed ad-hoc computing systems in which the participating peers directly distribute their tasks and share their resources without any form of hierarchical organization or centralized control [1-4]. Such networks offer numerous advantages, including a robust wide -area routing architecture, an efficient search capability, ano-nymity, excellent fault tolerance, a massive amount of redundant storage, and so forth. Furthermore, since each peer in the system is not only a client, but can also per-form the role of a server, the capacity and scalability of P2P systems are far higher than those of traditional client-server systems. Consequently, P2P overlay networks pro-vide an excellent solution for real-time applications, ad-hoc collaborative projects, and content sharing in large-scale distributed environments.

Although various P2P overlay networks have been proposed in recent years, decentralized, unstructured P2P systems such as Gnutella [1] and KaZaA [2] are the most commonly used in current Internet-based applications. In contrast to structured networks, content placement in P2P networks is unrelated to the overlay topology, and thus such networks are better equipped to deal with the prob-lem of high-churn peer populations. However, content lookup is an important issue in unstructured P2P net-works since the system lacks any indexing rules with which to store the information in a convenient form for search purposes, and thus the content search procedure requires the use of brute force techniques to flood the lookup query amongst the peers (e.g. Gnutella) or super-peers (e.g. KaZaA) until the desired content has been found.

KaZaA and the newest version of Gnutella (Gnutella v0.6) both create a two-layer hierarchical unstructured P2P system comprising an upper layer of "super-peers" (KaZaA) or "ultra-peers" (Gnutella) and an underlying layer of ordinary peers. In both systems, the super (or ultra) peers are chosen from amongst the participating nodes having a fast Internet connection and cannot be blocked by a firewall. These peers are chiefly responsible for servicing a small subpart of the peer network by in-dexing the files shared by all the ordinary peers con-nected to them and performing proxy search requests on their behalf. In practice, all of the lookup queries issued by an ordinary peer are directed initially to the associated super-peer, which (assuming that it does not possess the relevant information itself) then floods a lookup message to the other super-peers in the network.

Hierarchical P2P systems such as KaZaA and Gnutella have two major advantages compared to pure decentral-ized systems, namely a reduced discovery time and an improved ability to exploit the inherent heterogeneity of the participating nodes. Therefore, super -peer overlay networks offer the potential for building efficient and scalable file-sharing systems. However, establishing the optimal super-peer network design necessarily involves making various performance tradeoffs and raises a num-ber of key questions. For example, how should the super- peers connect with one another? How should a suitable topology be chosen for the super-peer overlay network? How should the network design utilize an efficient broadcasting algorithm to avoid broadcast storms and redundant messages? To what extent does the design provide a reliable service given the possibility that a hier-archical super-peer represents a potential point of failure for multiple associated clients?

In an attempt to address some of these questions, this study presents a method for dynamically constructing and maintaining the super-peer overlay topology of a two-layer hierarchical P2P system using a perfect differ-ence graph (PDG) method. In addition, a PDG-based for-warding algorithm is developed to improve the broad-casting efficiency of the P2P system by ensuring that each super-peer receives just one lookup query flooding mes-sage.

A. Outline of proposed super-peer selection and broadcasting scheme

In the super-peer overlay network construction and broadcasting scheme developed in this study, a super-peer table is maintained by a bootstrap server. Any peer joining the P2P network and wishing to become a super-peer must first issue a request to the bootstrap (BS) server. After

examining the bandwidth connectivity quality such as over an upload speed of 1 MB/s and a download speed of a 2 MB/s, the server either selects the peer as a super-peer, and sends the peer the corresponding for-ward and backward connections, or registers the peer as a redundant super- peer, and provides the peer with a list of super-peers from which it can use to connect to the sys-tem.

In the event that a new super- peer joins the overlay network or an existing super-peer leaves the system, the BS server automatically extends or shrinks the configuration of the super-peer topology by the proposed request process algorithm. Having established the overlay topology, a PDG-based forwarding algorithm is used to flood the lookup messages from the originator super-peer to the other super-peers in the network in such a way that each super-peer receives just one message.

B. Main contributions

The main contributions of this paper can be summarized as follows:

1. Super-peer overlay configuration scheme enables the dynamic, low-cost construction of balanced, low-diameter unstructured P2P systems.
2. The PDG-based flooding algorithm eliminates the problem of redundant lookup query flooding messages in the super-peer layer of the P2P sys-tem. Specifically, each super-peer receives just one broadcast message in place.
3. Performance evaluation demonstrates that the proposed super-peer hierarchical system outper-forms existing super-peer hierarchical unstruc-tured P2P systems in terms of a smaller network diameter, a lower number of lookup query flood-ing messages, and a lower average delay.

The remainder of this paper is organized as follows. Section II presents a brief overview of the literature relat-ing to super-peer and cluster-based P2P systems, while section III introduces the basic principles of the perfect difference graph (PDG) method and broadcasting proto-col in the context of super-peer overlay networks. Section IV describes the proposed PDG-based super-peer over-lay-construction algorithm to deal with the super-peers request and maintain the super-peer overlay topology. Section V presents a theoretical evaluation of the per-formance of the PDG-based super-peer topology configu-ration method and broadcasting scheme. Experimental results on our testbed for the proposed prototype system are shown in Section VI. Section VII discusses the com-munication overheads incurred between the BS server and the super-peers in the P2P network and considers the need for multiple BS servers to minimize the impact of single-point failures. Finally, Section VIII presents some brief conclusions.

II. RELATED WORK

In recent years, various hierarchical two-layer unstruc-tured P2P systems have been proposed as a means of scaling up conventional unstructured P2P systems. Such systems, of which Gnutella vs. 6 [1] and KaZaA [2] are the most widely used, comprise super-peers and ordinary peers and have a number of key advantages for the execution of large-scale distributed applications, including a higher search efficiency and the ability to harness the power and resources of multiple heterogeneous nodes. However, they also suffer the problems of a heavy work-load and the risk of single-point failures, i.e. the failure or departure of a single super-peer causes all of its children (ordinary peers) to lose their connections to the system until they are reassigned to a new super-peer.

In an attempt to address these issues, Yang *et al*. proposed several rules of thumb for accomplishing the major trade-offs required in super-peer networks and introduced a k-redundancy concept for improving the system reliability and reducing the workload imposed on the individual super-peers. Watababe *et al*. presented a method for reducing communication overheads in a two-layer hierarchical P2P system by allowing ordinary peers within designated clusters to communicate directly with one another rather than through a super-peer.

Gia improved the performance of unstructured P2P systems by using a dynamic scheme to select appropriate super-peers and to construct the topology around them in an adaptive manner. Furthermore, a search-based random walk mechanism was proposed for directing the lookup messages issued by the ordinary peers towards the high-capacity nodes in the system. However, the effi-ciency of the search procedure relies fundamentally on the matching data being found very quickly. In the worst case scenario, the random walk search mechanism either gives up without finding a match or may have to traverse a very long path.

Pyun presented a protocol designated as SUPs for constructing the super-peer overlay topology of scalable unstructured P2P systems using a random graph method. The results showed that SUPs was not only more compu-tationally straightforward than the scheme presented in , but also was much compatible with existing system and was likely be adopted. Although the resulting over-lay network had a lower diameter and the topologies produced were low cost and almost regular, the authors didn't discuss the content search procedure in detail.

Xiao *et al*. [10] presented a workload model for estab-lishing the optimal size ratio between the super-layer and the leaf-layer, and proposed an efficient dynamic layer management (DLM) scheme for super-peer architectures. In the proposed approach, the DLM algorithm automati-cally selects the peers with larger lifetimes and capacities as super-peers and designates those with shorter lifetimes and capacities as leaf peers. However, the DLM algo-rithm inevitably incurs a substantial traffic overhead in exchanging information amongst neighboring peers and a peer adjustment overhead is incurred when a super-peer is demoted to be a leaf-peer. Moreover, they did not ex-amine which topology is suitable for super-peers to maximize their benefits.

III. SUPER-PEER OVERLAY NETWORKS AND BROADCASTING PROTOCOLS

Since super-peers have a fast Internet connection, they can accommodate a high traffic demand. The topology for super-peers can be modeled by a graph with higher de-gree, in which vertices represent individual super-peers while undirected edges stand for connections between super-peers. Since all of the super -peers are regarded as being of equal importance in terms of their ability to route traffic, it is suitable to construct the topology of super-

peers into a regular graph, which the degree of each ver-tex is the same, to easily achieve load balancing. Besides balancing the load within the P2P system, it is also desir-able to minimize the diameter of the super-peer overlay topology in order to limit the length of the paths which a lookup query generated by any super-peer must traverse to reach the other super-peers in the network. Finally, the degree of the super-peers in the overlay topology should be such that the P2P system is both practical and scalable.

Table 1 summarizes the vertex degree and graph di-ameter of various well known graph methods. As shown, the complete graph models a regular *n*-vertex network in which the vertex degree is $d= O(n)$ and the diameter is $D=1$. (Note that the diameter indicates the maximum number of hops in the path between the source-destination vertices in the graph.) Although the complete graph provides a simple approach for modeling a net-work, it is impractical for large *n* and lacks the scalability required to support network growth. Therefore, it is gen-erally preferable to relax the maximum hop-count pa-rameter to D=2 for practical large-scale systems and to model the network using a perfect difference graph (PDG).

Each vertex in a PDG has a degree $O(\sqrt{n})$, and thus the network is significantly more scalable than that mod-eled by a complete graph (i.e. $O(n)$). Furthermore, even though the vertices in the PDG have a lower degree than those in the complete graph, the performance of a PDG-based network is similar to that of a complete graph-based system. In addition, Table 1 shows that the other common graph methods have both a lower vertex degree than the PDG method and a greater diameter. Thus, the PDG-like graph is an ideal solution for the dynamic super-peer overlay construction scheme presented in this study.

TABLE 1
COMPARISION OF VERTEX DEGREE AND GRAPH DIAMETER

Vertex Degree	Graph Diameter	Example
O(n)	1	Complete – Graph
O(√n)	2	Perfect Difference Graph
Ω(n ln n)	Θ (ln n / 1 n ln n)	Random Graph
O(log n)	log n	Binary Tree, Hypercube
O(l)	n/2	Ring

A. Perfect difference graphs

PDGs [8], based on the mathematical notion of perfect difference sets (PDSs), are undirected graphs of degree $d=2\delta$ (where δ is the number of elements in the PDS) and diameter D=2.

Definition 1: A PDG is an undirected inter-connection graph with $n= \delta^2 + \delta + 1$ vertices, numbered *0* to *n-1*. In the PDG, each vertex *i* is connected via undirected edges to vertices $(i \pm s_j) \pmod n$ for $1 \le j \le \delta$, where s_j is an element of the PDS $\{s_1, s_2, \ldots, s_j\}$ of order δ.

Table 2 illustrates the number of vertices, the order and the number of elements in the first ten PDSs. Figure 1 presents a PDG graph based on the PDS {1, 3}. Since there are two elements in this PDS, the graph has seven vertices. Furthermore, the PDS has a degree of 2δ, and thus each vertex has four undirected edges leading to neighboring vertices. For example, vertex 0 has undirected edges leading to vertices $(0 \pm 1) \bmod 7$ and $(0 \pm 3) \bmod 7$. In other words, vertex 0 has undirected edges to vertices 1, 3, 4 and 6. For convenience, the following terms are adopted when discussing the PDG methodology in the remainder of this paper:

- Ring edge: the edge connecting consecutive vertices *i* and $i \pm s_1 \pmod n$, where $s_1 \pm 1$.
- Chord edge: the edge connecting non-consecutive vertices *i* and $i \pm s_j \pmod n$, $2 \le j \le \delta$.
- Forward edges: for vertex *i*, the forward edges include the chord edge connecting vertices *i* and $i + s_j \pmod n$ and the ring edge connecting vertices *i* and $i + s_1 \pmod n$.
- Backward edges: for vertex *i*, the backward edges include the chord edge connecting vertices *i* and $i - s_j \pmod n$ and the ring edge connecting vertices *i*and $i - s_1 \pmod n$.

For example, in Fig. 1, the forward edges of vertex 0 are the edges connecting vertex 0 to vertices 1 and 3, respectively, while the backward edges are the edges connecting vertex 0 to vertices 4 and 6, respectively.

TABLE 2
CORRELATION BETWEEN NUMBER OF VERTICES, SUPER-PEER ORDER AND PERFECT DIFFERENCE SETS

n	δ	Perfect difference sets
7	2	1,3
13	3	1,3,9
21	4	1,4,14,16
31	5	1,3,8,12,18
57	7	1,3,13,32,36,43,52
73	8	1,3,7,15,31,36,54,63
91	9	1,3,9,27,49,56,61,77,81
133	11	1,3,12,20,34,38,81,88,94,104,109
183	13	1,3,16,23,28,42,76,82,86,119,137,154,175
273	16	1,3,7,15,63,90,116,127,136,181,194,204,233,238,255

Proposition 1. If G=(V,E) is a graph consisting of a set of vertices V and a collection of edges E connecting pairs of vertices in *V,* then $\sum_{v \in V(G)} d(v) = 2e(G)$, where $d(v)$ represents the de-gree of vertex *v* in a graph *G* and $e(G)$ represents the number of edges in *G*.

Proof. Summing the degree of vertices counts each edge twice, since each edge has two ends and contributes to the vertex degree at each endpoint [5].

Lemma 1. The total number of edges in a PDG is equal to $n.\delta = (\delta^2 + \delta + 1).\delta$.

Proof. Since the connectivity of the PDG leads to a degree $d = 2\delta$, the total degree of vertices equals $\sum_{v \in V(G)} d(v) = n.2\delta$. By Proposition 1, $n.2\delta$ is equal to 2*e.* Therefore, the total number of edges is equivalent to $n.\delta = (\delta^2 + \delta + 1).\delta$.

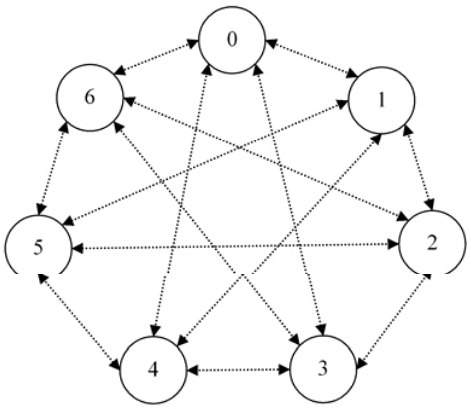

Fig. 1. Perfect difference graph with 7 vertices based on Perfect difference set {1, 3}.

A. Broadcasting over an unstructured P2P network

In unstructured P2P systems, a broadcasting protocol is required to enable the delivery of messages from a source node to all the other nodes in the network. One of the most common forms of broadcasting protocol is the flooding approach, in which the source node simply sends a copy of its message to each of its neighbors. When the neighbors receive this message, they in turn send copies of the message to all of their neighbors other than the neighbor from which they received the original message.

The flooding approach is commonly used for the search of data objects over unstructured P2P systems. For example, Gnutella uses an application-level forwarding scheme known as sequence-number-controller [7] (abbreviated as SNC) to broadcast content lookup queries amongst all the peers. In SNC, the source peer puts its address and a broadcast sequence number into a broad-cast message, and then sends this message to all of its neighbors. Each peer maintains a list of the source ad-dresses and sequence numbers of all the broadcast mes-sages it has received and forwarded. Thus, when a peer receives a broadcast message, it first checks whether or not the message is already in this list. If the message has already been added to the list, the received message is simply dropped. However, if the message is not included in the list, the peer duplicates it and forwards it to all of its neighbors other than the neighbor from which it received the message. Gnutella also uses a time-to-live (TTL) parameter to limit the total number of hops over which a query message can pass. Thus, whenever a Gnutella client receives and duplicates a query, it decrements the TTL value by one before forwarding the query to its neighbors. In the event that the value of the TTL is reduced to zero, the client simply takes no further action.

Super -peer overlay networks are similar to Gnutella in that the super -peers within the network depend on a flooding-based approach to relay the lookup query messages when searching for data objects. Flooding-based approaches resolve the problem of broadcast storms in the P2P network, but do not entirely eliminate the trans-mission of redundant broadcast messages. As a result, the communication overhead within the network is inevita-bly higher than that in the ideal scenario in which each super-peer in the P2P network receives just one copy of the broadcast message.

B. Broadcasting over super-peer perfect difference graph overlay network

This study develops a PDG-based forwarding algorithm [9] in which the flooding messages are disseminated to all the super-peers in the overlay topology via the forward and backward edges of the graph. The forwarding algorithm can be invoked by any vertex to initiate a broadcast and ensures that each vertex receives just one copy of the flooding message.

Assume that vertex *i* wishes to flood a message to every other vertex in the overlay network. The PDG-based flooding algorithm executes the following two-step procedure:

Step 1: Vertex *i* sends a flooding message with TTL=2 to its entire forward neighbors and sends a flooding message with TTL=1 to all of its backward neighbors.

Step 2: If an intermediate vertex receives the mes-sage, it duplicates the message to all of its backward neighbors other than the neighbor from which it received the original message.

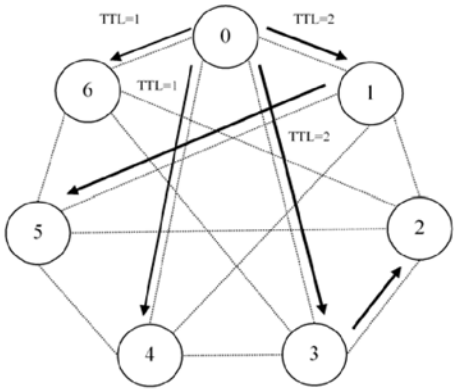

Fig. 2. PDG-based forwarding algorithm

Figure 2 presents a schematic illustration of the pro-posed PDG-based forwarding algorithm for a super-peer overlay network forming a PDG with an order of $\delta \square 2$. In this example, it is assumed that super-peer 0 wishes to flood a lookup message to all the other super-peers in the network. In accordance with the two- step procedure de-scribed above, super-peer 0 sends a flooding message with TTL= 1 along its backward edges to neighbors 4 and 6, respectively. Since the TTL value is reduced to zero following its decrement upon receipt at these nodes, neighbors 4 and 6 take no further action. Meanwhile, super-peer 0 also sends a flooding message with TTL=2 along its forward edges to neighbors 1 and 3, respectively. Following the receipt of these messages, the TTL value is reduced to 1, and thus both neighbors forward a copy of the message along all their backward edges other than the edge on which they received the original message. In other words, neighbor 1 duplicates the message to node 5, while neighbor 3 copies the message to node 2. Nodes 2 and 5 obtain a value of TTL=0 when decrementing the TTL

parameter, and therefore take no further action.

IV. SYSTEM ARCHITECTURE AND CONSTRUCTION

A. System model

The hierarchical unstructured P2P system considered in this study is modeled by an undirected graph $G=(V,E)$ consisting of a set of V vertices and E edges connecting pairs of vertices in V. As described in Section 3, the peers in the P2P network form the vertices of the graph, while the connections between the individual peers are represented by the edges of the graph. Note that hereafter the terms graph and network, node and vertex, and edge and connection, respectively, are used interchangeably with no difference in meaning. An assumption is made that the graph G is divided into several subgraphs $G^i=(V^i,E^i)$, where $i=1,2,3,...,m$. and V^i is a non-empty subset of vertices and includes at least one node, referred to as the super-peer node Vs^i. All of the other nodes in V^i apart from the super-peer node are referred to as ordinary peers. The connections between the ordinary peers in V^i and the as-associated super-peer, V_s^i , are defined by the set of edges E^i . Note that an assumption is made that each ordinary peer is connected by an undirected edge (referred to hence-forth as an intra-connection) only to its associated super-peer, i.e. the individual ordinary peers are not connected directly to one another.

Let $G'=(V',E')$ be an undirected graph with $V' \in V$ and $E' \in E$, where V is a set of super-peers and E' is a collection of connections between super-peers. Note that these connections are referred to as "inter-connections" to distinguish them from the "intra-connections" in E^i between the ordinary nodes and the super-peer nodes. These con-nections may be either in the forward direction or the backward direction (as defined previously in Section 3). The graph G' is a PDG with a PDS of order δ if it satisfies the following condition: the super-peer V_s^i is connected via undirected edges to the other super-peers $V_s^{(i \pm sj)(mod\ m)}$ for $1 \le j \le \delta$, where s_j is an element of the PDS $\{ s_1, s_2 , \ldots, s_j\}$ of order δ and m is the total number of super-peers in G'.

In the hierarchical unstructured P2P system considered in the present study, the ordinary peers communicate the indexes of their shared files to their associated super-peer via the intra-connections between them. If an ordinary peer wishes to search for an object, it issues a lookup re-quest to its associated super-peer via its intra-connection. When the super-peer receives this lookup query, it per-forms an initial search of its own local index to see whether or not it holds the object of interest. If it finds the object, it replies directly to the requestor node; otherwise it floods a lookup query to the other super-peers via its inter-connection using the PDG-based forwarding algo-rithm described in Section 3.3.

B. System construction

In the proposed two-layer hierarchical unstructured P2P system, at least one node exists as an entry point for new nodes wishing to join the network. This node, referred to as a bootstrap (BS) server, provides new ordinary nodes joining the system with a randomly compiled list of su-per-peers, accepts or rejects a super-peer request, and maintains the super-peer overlay topology.

In the case where a new node wishes to joint the net-work as an ordinary peer, it sends a join request to the bootstrap server. Having processed its request, the server sends the node a super-peer list containing the addresses of several randomly-selected super-peers. When the peer receives this list, it selects a super-peer with the minimal response time to connect to the network. Once the new peer connects to the super-peer, it becomes a children peer of the super- peer. When the ordinary peer decides to leave the system, it simply sends a message to that effect to its parent super-peer, which then updates the corre-sponding intra-connection status to show that the node no longer forms part of the network.

TABLE 3
EXAMPLE OF SUPER-PEER TABLE

Vertex ID	Address of super-peers	Forward connections	Backward connections	Status
0	IP_A	IP_B, IP_D	IP_E, IP_G	1
1	IP_B	IP_C, IP_E	IP_F, IP_A	1
2	IP_C	IP_D, IP_F	IP_G, IP_B	1
3	IP_D	IP_E, IP_G	IP_A, IP_C	1
4	IP_E	IP_F, IP_A	IP_B, IP_D	1
5	IP_F	IP_G, IP_B	IP_C, IP_E	1
6	IP_G	IP_A, IP_C	IP_D, IP_F	1

Any peer joining the P2P network and wishing to become a super-peer must first issue a request to the bootstrap (BS) server. The peer should have a fast Internet connection such as an upload speed of 1 MB/s and a download speed of a 2 MB/s. Moreover, the connection cannot be blocked by a firewall to provide connections for ordinary peers by TCP and UDP ports. By verifying the bandwidth, the BS server either selects the peer as a su-per-peer, and sends the new peer the corresponding for-ward and backward connections, or registers the peer as a redundant super-peer, and provides the peer with a list of super-peers to connect to the system.

The BS server takes advantage of a super-peer table to control the super-peer overlay topology. Table 3 includes the vertex ID number, the super-peer IP address, the forward and backward connections of each super-peer, and the status of the vertex. Here, the ID number is simply the number of the super-peer in the perfect difference overlay graph, and is mapped to the IP address of the corresponding super-peer. Meanwhile, the forward and backward connection fields represent the IP addresses of the forward and backward neighbors of each super-peer, respectively. Finally, the status field contains a value of "1" if the super-peer is active (i.e. it forms part of the current perfect difference overlay graph), and has a value of "0" if the peer has been designated as a redundant super-peer by the BS server.

Figure 3 illustrates the scenario in which a new peer joins the super-peer overlay network. Note that in this figure, a super-peer overlay topology has already been constructed by the BS server and the super-peers form a perfect difference overlay graph with an order of $\delta = 2$. As indicated in the legend, the super-peers are represented by large open circles, while the ordinary peers are depicted as small black circles. In addition, the inter-connections between the super-peers are shown using thick double-arrow- headed lines, while the intra-connections between the ordinary peers and the super-peers are shown using thin solid lines. In the example shown in Fig. 3, the new-joining

peer (with an address IP_G) issues a request to become a super-peer. The BS server processes the peer request and then accepts the peer as a super-peer. The BS server adds IP_G in Table 3, and sends IP_G information, such as super-peer status, the corresponding forward connections of IP_A and IP_C, and backward connections of IP_D and IP_F.

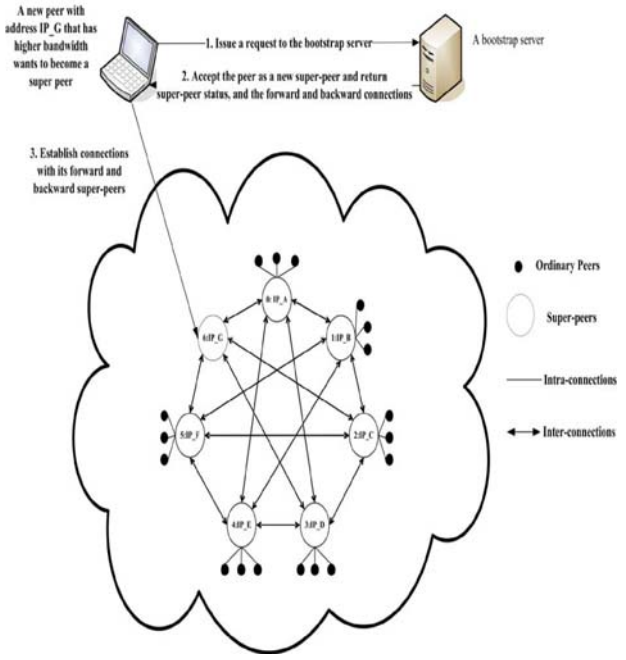

Fig. 3. Schematic illustration showing a new peer joining the super-peer overlay networks

The BS server manages the super-peer's request and maintains the overlay topology by a request process algorithm. A peer to join the P2P network as a super-peer is-sues a joining request, including the request type, its IP and bandwidth description. A super-peer or redundant peer to leave the system issues a departing request com-prised of the request type and the departing peer IP. The following subsections discuss the details of the proce-dures performed at the BS server when super-peers join or leave the network prompting the requirement to ex-tend or shrink the overlay topology, respectively. The discussions adopt the following notations: δ - the order of the current PDS; k - the order of the predecessor PDS; and £ – the order of the successor PDS. Note that k and satisfy the constraint $k < \delta <$ £ . Finally, n is the total num-ber of active and redundant super-peers.

C. Extension of topology to accommodate new super-peers

In accordance with the request process algorithm, any peer with a fast Internet connection to enter the P2P net-work as a super-peer issues a joining request with its bandwidth description and IP to the BS server. After identifying the connectivity quality, the BS server accepts the peer as a super-peer, and assigns the new peer the appropriate forward and backward connections.

When the number of super-peers is larger than the value ($\delta^2 . \delta . 1$) , it represents that all the positions in the PDG are already filled with active super-peers. If the re-questing peer is qualified to become a super-peer, the BS server designates the peer as the role of a fredundant su-per-peer, and is allowed to connect to the network by accessing a super-peer with a minimal response time se-lected from a list

randomly compiled by the BS server.When the number of super peers and redundant super-peers increases to threshold value, that can be given as ½[(δ^2+ δ+1) + ($\ell^2 + \ell$ +1)], there are a number of redundant super-peers existing in the

system. In order to utilize the bandwidth capability of the redundant super-peers and increase system scalability, the order of the cur-rent PDS is enlarged to that of the successor PDS and the super-peer overlay topology is extended accordingly.

Thus, the BS server first assigns the new joining peer a new vertex ID and the peer IP into the super-peer table. It then assigns the status of 1 to the new joining peer and all of redundant super-peers. Next, the BS server calculates and updates new forward and backward connections based on the new order δ in the super-peer table for these active super-peers. Finally, the BS server sends the new joining peer information, such as the status, the for-ward connections and the backward connections. The BS server also notifies redundant super-peers about the status, the forward connections, and the backward connections and informs the original active super-peers about the new forward and backward connections.

We illustrate an example to describe the overlay topology extension. In the initial set-up phase (i.e. no super-peers have yet been identified), the BS server adopts a low-order PDS (i.e. an order of 2) to construct an initial super-peer overlay network for a maximum of 7 super-peers. Assume that there are 10 new peers wishing to be-come super- peers. Since the number of new peers exceeds the number of available spaces in the overlay network, the former 7 peers are assigned as super-peers, and the remaining peers temporarily designated as redundant peers. Later, when a new peer wishing to become a super-peer enters the system, it will result in the number of peers, including active super-peers, redundant super-peers, and the new joining peer, exceeding a threshold 10(= (7+13)/2). The BS server according to the request process algorithm extends the super-peer overlay topology using a PDS with an order of 3, thereby allowing space for a maximum of 13 super-peers. Thus, the redundant super-peers and the new joining peer are assigned as new super-peers and are informed about the IP addresses of their forward and backward connections by BS server. At this point, 11 active super-peers participate in the new enlarged topology.

D. Shrinking of topology to accommodate departure of existing super-peers

When a super-peer leaves the P2P system, it transmits a departure message to both the BS server and all of its child ordinary peers. In accordance with the request process algorithm, the BS server randomly selects one of the redundant super- peers to take the place of the departing super-peer in the super-peer topology. After selecting a redundant super-peer, the BS server assigns it the vertex ID, the forward and backward connections of the departure super-peer, and the active status. The BS server then replaces the departure super-peer IP with the selected

super-peer IP. Finally, the server informs the redundant peer to be an active super-peer and instructs those active peers infected by the super- peer departure to update their connection address records accordingly.

Having received a departure message from a super-peer wishing to disconnect from the P2P network, the ordinary peers then re-connect to the network by choosing one of these super-peers with the lowest response ($\delta^2+\delta+1$), there are not redundant super-peers available to replace the departing super-peers in the overlay topology. The BS server simply update the corresponding forward and backward connections of the super-peer table for those peers infected by the super-peer departing. It then deletes the records of the departing super-peer from the super-peer table.

Since the current super-peer overlay network is an incomplete PDG, and thus some of the super-peers lose their forward or backward connections. As a result, some of the super-peers may fail to receive the TTL=2 messages broadcasted by the other super-peers in the overlay network. To overcome the effect, when the number of super-peers decreases to the threshold value, $\frac{1}{2}[(k^2 + k +1) + (\delta^2+\delta+1)]$, the order of the current PDS is reduced to that of the predecessor PDS, and the super-peer overlay topology is shrunk accordingly.

Thus, the BS server first assign new vertex IDs to the remaining super-peers. The super-peers with vertex IDs less than $\delta^2+\delta+1$ are active super-peers to participate in the reduced topology. Others are designated as redundant super-peers. The BS server then calculates and updates new forward and backward connections based on the new order δ in the super-peers and sets the status of those redundant super-peers equal to 0. Finally, the BS server notifies active super-peers about the forward and backward connections and informs redundant super-peers about the status and the addresses of some randomly selected super-peers.

To prevent abnormal super-peer departure, super-peers periodically send each other hello messages to maintain the status of their forward and backward connections. If one super-peer that sends a specific super-peer a hello message can not receive a response message after a time out, the message originator discriminates that the super-peer is failure. It then issues a departure request with the faulty super-peer IP to the BS server. When the BS server receives the request, it will follow the request process algorithm to update connections of those super-peers infected by the faulty super-peer. By the same token, the parent super-peer of the redundant super-peer can detect whether the redundant super-peers fail, the parent super-peer is responsible for sending a departure message with the redundant super-peers IP to the BS server against abnormal redundant super-peer leaving.

Algorithm: Request Process in the BS server

Input: Receiving a request(TYPE, IP, BW)

Output: Update the super-peer table in the BS server and return information to the super-peers

Initialize a super-peer table;
$n, k \leftarrow 0$;
$\delta \leftarrow 2$;
$\ell \leftarrow 3$;
while (a request(TYPE, IP, BW)) **do**
if (TYPE==1) **then** //TYPE=1 represents a joining request;
if (examine BW) **then**
$n \leftarrow n+1$;
if ($n \leq (\delta^2 + \delta +1)$) **then**
Assign the new joining peer a vertex ID, the peer IP, and for-ward and backward connections based on the order of δ , and the status of 1 into the super-peer table;
//Accept the new joining peer as a new super-peer;
Return the status, the forward connections and the backward
connections to the new super-peer;
else if (($\delta^2+\delta+1 \leq n \leq 1/2[(\delta^2+\delta+1)(\ell^2+\ell+1)]$)
then
Assign the new joining peer a vertex ID, the peer IP, and the status of 0 into the super-peer table;
//Register the new joining peer as a redundant super-peer;
Return the status and the addresses of some randomly selected super-peers to the new peer to enable it to connect to the
P2P system;
else if ($n > 1/2[(\delta^2+\delta+1)(\ell^2+\ell+1)]$)
then
$k \leftarrow \delta$;
$\delta \leftarrow \ell$;
$\ell \leftarrow$ a new successor order of a larger PDS;
Assign the new joining peer a vertex ID and the peer IP into the super-peer table;
Assign the status of 1 to the new joining peer and all of redundant super-peers;
Calculate and update new forward and backward connec-tions based on the new order δ into the super-peer table for these active super-peers;
//Accept the peer as a new super-peer and extend the super-
//peer overlay topology;
Return the status, the forward connections and the back-ward connections to the new super-peer;
Inform the redundant super-peers about the status, the for-ward connections, and the backward connections;
Inform the original super-peers about the new forward and backward connections;
else
Return a super-peer list containing the addresses of several randomly-selected super-peers to enable the new joining peer to connect to the P2P system;
//The BW of the peer doesn't meet the system requirement;
//It is just an ordinary peer;
if (TYPE==0) **then**
//TYPE=0 represents a departure request;
$n \leftarrow n-1$;
if (examine IP whether the peer is a super-peer) **then**
if ($n \geq (\delta^2+\delta+1)$) **then**
Randomly select a new super-peer from the redundant super-peers;
Assign the vertex ID, the forward and backward connections of the departure super-peer, and the status of 1

to the selected super-peer;

Swap the departure super-peer IP of the super-peer table for the selected super-peer IP;

//Choose a redundant super-peer to become a new super-peer;

Inform the new super-peer about the status, the forward connections, and the backward connections;

Inform those active peers infected by the super-peer departing about the new super-peer IP;

else if ($1/2[(k^2+k+1)(\delta^2+\delta+1)] < n < (\delta^2+\delta+1)$) **then**

Update the corresponding forward and backward connections of the super-peer table for those peers infected by the super-peer departing;

Delete the records of the departing super-peer from the super-peer table;

//No redundant super-peers can replace the departing super-peer. Therefore, the BS server perform super-peer table updating process only;

Inform the messages of connections non-available to those active peers infected by the super-peer departing;

else if (($n \leq 1/2[(k^2+k+1)(\delta^2+\delta+1)]$) **then**

$\ell \leftarrow \delta$;

$\delta \leftarrow k$;

$k \leftarrow$ a new predecessor order of a smaller PDS;

Assign new vertex IDs to the remaining super-peers;

Calculate and update new forward and backward connections based on the new order δ into the super-peer table for super-peers with vertex IDs less than $\delta^2+\delta+1$;

Set the status of all vertex IDs equal and greater than $\delta^2+\delta+1$ to 0;

Inform all of super-peers with vertex IDs less than $\delta^2+\delta+1$ about the new forward and backward connection;

Inform all of super-peers with vertex IDs equal or greater than $\delta^2+\delta+1$ about the status and the addresses of some randomly selected super-peers;

else

Delete the records of the departing redundant super-peer from the super-peer table;

//The departure peer is a redundant super- peer;

end while

Figure 4 illustrates the variation of the super-peer degree with the number of super-peers in a random-based overlay network and a PDG-based overlay network, respectively. In general, the super-peer degree provides an indication of the cost incurred in maintaining the connections of the super-peers in the overlay topology. Thus, Fig.4 shows that the maintenance costs of the super-peers in the proposed PDG-based network are higher than those of the super-peers in the random-based overlay network.

Although, formulations for the diameters of a random-based overlay network and a PDG-based overlay network, respectively, the diameter of the random overlay network cannot be precisely determined by the number of super-peers. Therefore, Fig. 5 compares the lower bound of the random-based overlay network diameter with the diameter of the PDG-based network. Although the random-based overlay network diameter represents the best-case scenario for this particular type of network, it can be seen that the diameter of the PDG-based overlay network is significantly smaller at all values of n equal to or greater than 13.

As each super-peer in the PDG-based overlay topology receives just one copy of the broadcast message when the originator super-peer utilizes the PDG-based forwarding algorithm to flood a lookup query. By contrast, in the random-based overlay network using the SNC forwarding algorithm, the number of broadcast messages received by each super-peer varies as a power law function of δ . Figure 6 illustrates the variation of the number of broadcast messages with the number of super-peers in the PDG-based and random-based overlay networks, respectively. The results clearly demonstrate that the PDG-based forwarding algorithm generates significantly fewer messages than the SNC forwarding algorithm. Furthermore, it is evident that the relative advantage of the PDG-based forwarding scheme increases as the scale of the super-peer topology increases.

Figure 7 illustrates the variation of the average flooding delay in the random-based and PDG-based overlay net- as a function of the number of super-peers. Note that for simplicity, the results presented for the random- based network are based on the lower bound of the network diameter. The results clearly show that the average flooding delay incurred by the PDG-based forwarding algorithm is significantly lower than that of the SNC forwarding algorithm. Again, the performance improvement of the PDG-based forwarding scheme becomes increasingly apparent as the scale of the super-peer overlay network increases. Specifically, it is observed that as the number of super-peers increases towards infinity, the average delay converges to a value close to that of the network diameter, i.e. 2, since under these conditions, the number of neighboring nodes of the originator super-peer is far lower than the total number of super-peers in the overlay network.

The results presented above confirm that the PDG-based forwarding algorithm proposed in this study outperforms the SNC forwarding algorithm used in a conventional random-based super-peer overlay topology in terms of a reduced number of broadcast messages and a lower average hop-count delay.

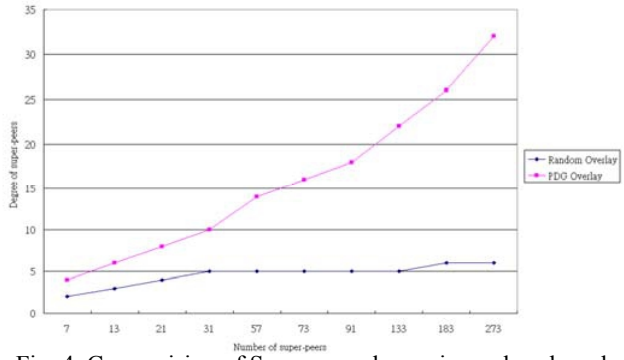

Fig. 4. Comparision of Super-peer degree in random-based and PDG-based overlay networks

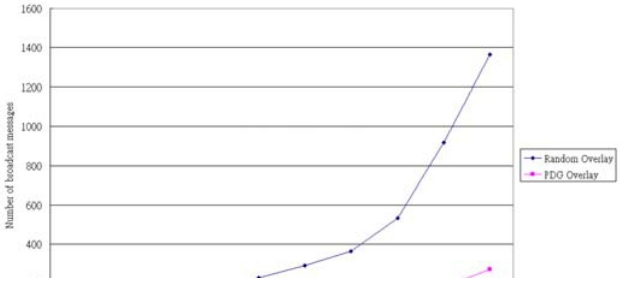

Fig. 6. Comparison of number of flooding lookup messages incurred in random-based and PDG-based overlay networks

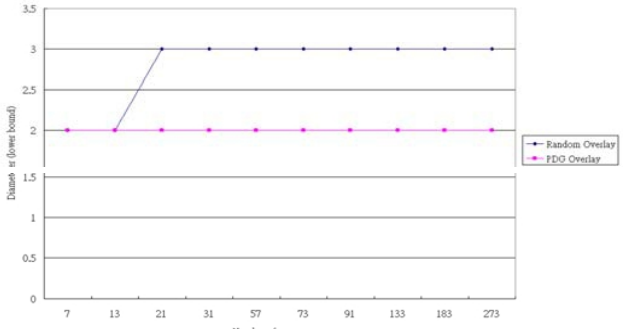

Fig. 5. Comparison of network diameter in random-based and PDG-based overlay networks

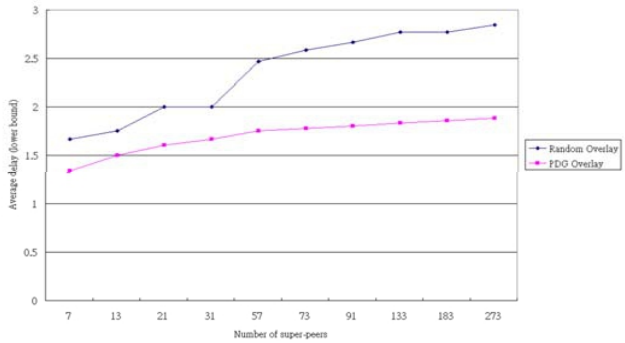

Fig. 7. Comparison of average delay (measured in hop counts) in random-based and PDG-based overlay networks

6 IMPLEMENTATION

To evaluate the file transfer performance of the proposed two-layer hierarchical unstructured P2P system using a perfect difference graph (PDG), we implemented a proto- type super-peer and BS server incorporating the request process algorithm presented in Section 4 on our tested . This work presents a series of experimental results to benchmark the performance of the proposed two-layer hierarchical unstructured P2P system against that of a Gnutella hierarchical unstructured P2P system.

The initial super-peer overlay topology is constructed by 91 nodes on the testbed with a bandwidth capacity 100 Megabits/sec to form a Gnutella P2P and a PDG-based overlay. The PDG-based overlay topology makes use of PDS with an order of 9 described in Section 3.1, thereby allowing space for a maximum of 91 super-peers. The system performance of the two schemes is quantified by hit rate. The hit rate is defined as the total number of discoveries over the total number of queries. A lookup query can result in multiple discoveries, which are copies of the same files stored at distinct nodes.

We allow the system to run several rounds on condition that the number of super-peers equals the shrinking threshold value (e.g. 10). In the beginning of each round, each super-peer issues lookup queries to search files not stored in its local space. Lookup queries are flooded by the PDG-based forwarding algorithm in the PDG-based overlay topology and are forwarded by SNC forwarding algorithm in the Gnutella P2P overlay topology with TTL=2, respectively. When each round terminates on the condition that each search request is serviced, one randomly selected super-peer leaves the system and the other active super-peers then enter next round to issue new lookup queries.

In the first round, since each overlay topology is a complete and connected graph for these overlay topologies, the hit rate achieves a highest value. Moreover, since the lookup queries on the PDG-based overlay can be efficiently flooded to each super-peer, the total number of discoveries is more than that on the Gnutella P2P overlay.Therefore, the hit rate of the PDG-based overlay is better than the Gnutella P2P overlay.

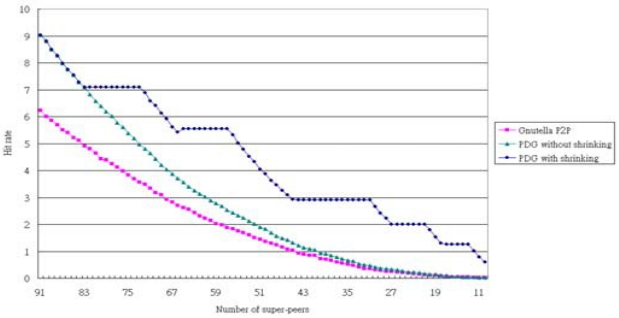

Fig. 8. Comparison of hit rate in Gnutella P2P and PDG-based overlay networks

8 CONCLUSION

This paper has presented an efficiency technique for constructing and maintaining the super-peer overlay topology of a two layer hierarchical P2P system using a perfect difference graph (PDG) – based method. In addition, a PDG-based forwarding algorithm is proposed for enhancing the efficiency of the lookup process. The performance of the proposed super-peer overlay topology based on a perfect difference graph has been benchmarked against a super-peer overlay topology based on a random graph using SNC forwarding algorithm. The theoretical results have grown that the PDG-based construction scheme and the forwarding algorithm yield a lower network diameter, a reduced number of lookup flooding messages, and a lower average hop-count delay. Through experimental results on our testbed, the proposed PDG-based two-layer hierarchy overlay is an efficient P2P solution in the dynamic network environment.

REFERENCES

[1] Gnutella - A protocol for Revolution,http://rfc-gnutella.sourceforge.net.com/.

[2] KaZaA available at http://www.kazaa.com/.

[3] Overnet/edonkey2000, available at http://www. edon- key2000.com/, 2000.

[4] Bittorrent, available at http://bitconjurer.org/BitTorrent/, 2003. [5] Douglas B. West, "Introduction to Graph Theory", 1996 by Prentice-Hall, Inc.

[6] B. Bollobás, "Random Graphs", London: Academic Press, 1985.

[7] James F. Kurose and Keith W. Ross, "Computer Networking: A Top-down Approach Featuring the Internet", third edition, Addison Wesley.

[8] B. Parhami and M. Rakov, "Perfect Difference Networks and Related Interconnection Structures for Parallel and Distributed Systems", IEEE Trans. On Parallel and Distributed Systems, vol.16, no. 8,pp.714-724, Aug. 2005.

[9] B. Parhami and M. Rakov, "Performance, Algorithmic, and Robustness Attributes of Perfect Difference Networks", IEEE Trans. on Parallel and Distributed Systems, vol. 16, no. 8, pp.725-736, 2005.

[10] Li Xiao, Zhenyun Zhuang, and Yunhao Liu, "Dynamic layer management in superpeer architectures", IEEE Trans. on Paral lel and Distributed Systems, vol. 16, issue, 11, pp. 1078 1091, Nov. 2005.

[11] Y. Dalal, R. Metcalfe, "Reverse Path Forwarding of BroadcastPackets", Communications of the ACM, Vol.21, No12, pp. 1040-1048, Dec. 1978.

[12] I. Stoica, and R. Morris et al., "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications", IEEE/ACM Trans. on Net., vol. 11, no. 1, pp. 17–32, 2003.

[13] R. G. Gallager, P. A. Humblet, and P. M. Spira, "A Distributed Algorithm for Minimum Weight-Spanning Trees", ACM Trans. on Programming Languages and Systems, January 1983, pp.66-77.

[14] F. C. Gartner, "A Survey of Self-Stabilizing Spanning-Tree Con- struction Algorithms", Computer and Communication Sciences, June 10, 2003.

[15] J. Yan, Y. Yang, and G. K. Raikundalia, "A SwinDeW p2p-based Decentralized Workflow Management System", IEEE Trans. on Systems, Man and Cybernetics, Part A, Volume 36, Issue 5, pp.922 – 935, Sept. 2006.

[16] Leonard D. Baumert, "Cyclic Difference Sets", Lecture Notes in Mathematics. Springer-Verlag, volume 182, 1971.

[17] T.P. Kirkman, "On the Perfect r-Partitions of r^2+r+1". Trans.Historical Soc. of Lancashire and Cheshire, vol. 9, pp. 127-142,1857.

[18] Guy and R. K., "Unsolved Problems in Number Theory", 2nd New York: Springer-Verlag, pp. 118-121, 1994.

[19] A. Rowstron and P. Druschel, "Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-to-peer Sys- tems", Proc. Middleware, 2001.

[20] B. Y. Zhao et al., "Tapestry: A Resilient Global-Scale Overlay for Service Deployment", IEEE JSAC, vol. 22, no. 1, pp. 41-53, Jan.2004.

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

[22] C. Lv, P. Cao, E.Cohen, K. Li, and S. Shenker., "Search and replication in unstructured peer-to-peer networks", In ICS, 2002.

[23] S. Ratnasamy et al., "A Scalable Content Addressable Network",Proc. ACM SIGCOMM, pp. 161-72, 2001.

[24] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti, "ALocal Search Mechanism for Peer-to-Peer Networks", In CIKM, 2002.

[25] D. Tsoumakos and N. Roussopoulos. "Adaptive Probabilistic Search for Peer-to-Peer Networks", 3rd IEEE Intl Conference on P2P Computing, 2003.

[26] V. Kalogeraki, D. Gunopulos, and D.Zeinalipour-Yazti, "A Local Search Mechanism for Peer-to-Peer Networks", In CIKM, 2002.

[27] J. Chu, K. Labonte, and B. Levine, "Availability and Locality Measurements of Peer-to-Peer File Systems", In SPIE, 2002.