An Efficient Superpeer Overlay Construction and Broadcasting Scheme Based on Perfect Difference Graph

Jung-Shian Li, *Member*, *IEEE*, and Chih-Hung Chao, *Member*, *IEEE*

Abstract—Two-layer hierarchy unstructured peer-to-peer (P2P) systems, comprising an upper layer of superpeers and an underlying layer of ordinary peers, are commonly used to improve the performance of large-scale P2P systems. However, the optimal superpeer network design involves several requirements including superpeer degree, network diameter, scalability, load balancing, and flooding performance. A perfect difference graph has desirable properties to satisfy the above design rationale of superpeers 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 superpeer 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 superpeer receives just one lookup query flooding message. The theoretical results show that the proposed system improves existing superpeer 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.

Index Terms—Unstructured peer-to-peer system, superpeer, perfect difference graph, forwarding algorithm.

1 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], [2], [3], [4]. Such networks offer numerous advantages, including a robust wide-area routing architecture, an efficient search capability, anonymity, 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 perform 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 provide 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 problem of high-churn peer populations. However, content lookup is an important issue in unstructured P2P networks 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 among the peers (e.g., Gnutella) or superpeers (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 "superpeers" (KaZaA) or "ultra-peers" (Gnutella) and an underlying layer of ordinary peers. In both systems, the super (or ultra) peers are chosen from among 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 indexing the files shared by all the ordinary peers connected 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 superpeer, which (assuming that it does not possess the relevant information itself) then floods a lookup message to the other superpeers in the network.

Hierarchical P2P systems such as KaZaA and Gnutella have two major advantages compared to pure decentralized systems, namely a reduced discovery time and an improved ability to exploit the inherent heterogeneity of the participating nodes. Therefore, superpeer overlay networks offer the potential for building efficient and scalable file-sharing systems. However, establishing the optimal superpeer network design necessarily involves making various performance trade-offs and raises a number of key questions. For example, how should the superpeers connect with one another? How should a suitable topology be chosen for the superpeer overlay network? How should the network

- *The authors are with the Department of Electrical Engineering, Institute of Computer and Communication Engineering, National Cheng Kung University, EE Building, 1 University Road, Tainan 701, Taiwan. E-mail: {jsli, n2892123}@mail.ncku.edu.tw.*

Manuscript received 12 Sept. 2008; revised 4 Apr. 2009; accepted 2 June 2009; published online 8 June 2009.
Recommended for acceptance by A. Boukerche.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2008-09-0345.
Digital Object Identifier no. 10.1109/TPDS.2009.94.

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 hierarchical superpeer represents a potential point of failure for multiple associated clients [28]?

In an attempt to address some of these questions, this study presents a method for dynamically constructing and maintaining the superpeer overlay topology of a two-layer hierarchical P2P system using a perfect difference graph (PDG) method. In addition, a PDG-based forwarding algorithm is developed to improve the broadcasting efficiency of the P2P system by ensuring that each superpeer receives just one lookup query flooding message.

1.1 Outline of Proposed Superpeer Selection and Broadcasting Scheme

In the superpeer overlay network construction and broadcasting scheme developed in this study, a superpeer table is maintained by a bootstrap server. Any peer joining the P2P network and wishing to become a superpeer 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 2 MB/s, the server either selects the peer as a superpeer, and sends the peer the corresponding forward and backward connections, or registers the peer as a redundant superpeer, and provides the peer with a list of superpeers from which it can use to connect to the system.

In the event that a new superpeer joins the overlay network or an existing superpeer leaves the system, the BS server automatically extends or shrinks the configuration of the superpeer 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 superpeer to the other superpeers in the network in such a way that each superpeer receives just one message.

1.2 Main Contributions

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

1. Superpeer 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 superpeer layer of the P2P system. Specifically, each superpeer receives just one broadcast message in place.
3. Performance evaluation demonstrates that the proposed superpeer hierarchical system outperforms existing superpeer hierarchical unstructured P2P systems in terms of a smaller network diameter, a lower number of lookup query flooding messages, and a lower average delay.

The remainder of the paper is organized as follows: Section 2 presents a brief overview of the literature relating to superpeer and cluster-based P2P systems, while Section 3 introduces the basic principles of the PDG method and broadcasting protocol in the context of superpeer overlay networks. Section 4 describes the proposed PDG-based superpeer overlay construction algorithm to deal with the superpeers' request and maintain the superpeer overlay topology. Section 5 presents a theoretical evaluation of the performance of the PDG-based superpeer topology configuration method and broadcasting scheme. Experimental results on our testbed for the proposed prototype system are shown in Section 6. Section 7 discusses the communication overheads incurred between the BS server and the superpeers in the P2P network and considers the need for multiple BS servers to minimize the impact of single-point failures. Finally, Section 8 presents some brief conclusions.

2 Related Work

In recent years, various hierarchical two-layer unstructured P2P systems have been proposed as a means of scaling up conventional unstructured P2P systems. Such systems of which Gnutella versus 6 [1] and KaZaA [2] are the most widely used comprise superpeers 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 workload and the risk of single-point failures, i.e., the failure or departure of a single superpeer causes all of its children (ordinary peers) to lose their connections to the system until they are reassigned to a new superpeer.

In an attempt to address these issues, Yang and Garcia-Molina [28] proposed several rules of thumb for accomplishing the major trade-offs required in superpeer networks and introduced a k-redundancy concept for improving the system reliability and reducing the workload imposed on the individual superpeers. Watanabe et al. [29] 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 superpeer.

Chawathe et al. [34] improved the performance of unstructured P2P systems by using a dynamic scheme to select appropriate superpeers 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 toward the high-capacity nodes in the system. However, the efficiency 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 and Reeves [30] presented a protocol designated as SUPs for constructing the superpeer overlay topology of scalable unstructured P2P systems using a random graph method. The results showed that SUPs were not only more computationally straightforward than the scheme presented in [34], but also were much compatible with existing system and were likely be adopted. Although the resulting overlay 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.

TABLE 1
Comparison of Vertex Degree and Graph Diameter

Vertex degree	Graph diameter	Example
$O(n)$	1	Complete Graph
$O(\sqrt{n})$	2	Perfect Difference Graph
$\Omega(n\ln n)$	$\Theta(\frac{\ln n}{\ln\ln n})$	Random Graph
$O(\log n)$	log$n$	Binary Tree, hypercube
$O(1)$	n/2	Ring

TABLE 2
Correlation among Number of Vertices, Superpeer 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

Xiao et al. [10] presented a workload model for establishing the optimal size ratio between the superlayer and the leaf layer, and proposed an efficient dynamic layer management (DLM) scheme for superpeer architectures. In the proposed approach, the DLM algorithm automatically selects the peers with larger lifetimes and capacities as superpeers and designates those with shorter lifetimes and capacities as leaf peers. However, the DLM algorithm inevitably incurs a substantial traffic overhead in exchanging information among neighboring peers and a peer adjustment overhead is incurred when a superpeer is demoted to be a leaf peer. Moreover, they did not examine which topology is suitable for superpeers to maximize their benefits.

3 Superpeer Overlay Networks and Broadcasting Protocols

Since superpeers have a fast Internet connection, they can accommodate a high traffic demand. The topology for superpeers can be modeled by a graph with higher degree in which vertices represent individual superpeers while undirected edges stand for connections between superpeers. Since all of the superpeers are regarded as being of equal importance in terms of their ability to route traffic, it is suitable to construct the topology of superpeers into a regular graph, where the degree of each vertex is the same, to easily achieve load balancing. Besides balancing the load within the P2P system, it is also desirable to minimize the diameter of the superpeer overlay topology in order to limit the length of the paths that a lookup query generated by any superpeer must traverse to reach the other superpeers in the network. Finally, the degree of the superpeers in the overlay topology should be such that the P2P system is both practical and scalable.

Table 1 summarizes the vertex degree and graph diameter 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 network, it is impractical for large n and lacks the scalability required to support the network growth. Therefore, it is generally preferable to relax the maximum hop-count parameter to $D = 2$ for practical large-scale systems and to model the network using a PDG.

Each vertex in a PDG has a degree $O(\sqrt{n})$, and thus, the network is significantly more scalable than that modeled 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 superpeer overlay construction scheme presented in this study.

3.1 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 interconnection 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 $(i \pm s_j) (\bmod n)$ for $1 \leq j \leq \delta$, 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 10 PDSs. Fig. 1 presents a PDG graph based on the PDS {1, 3}. Since there are two elements in the 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 $i \pm s_1 (\bmod n)$, where $s_1 = 1$.

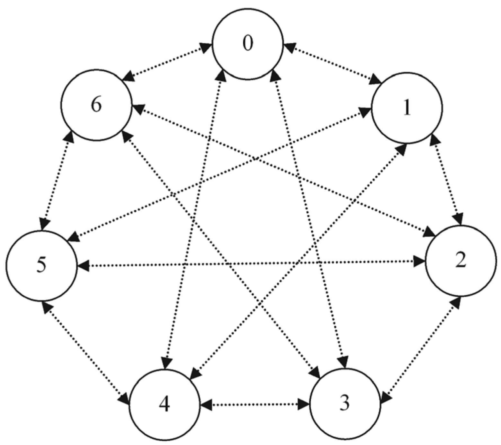

Fig. 1. Perfect difference graph with seven vertices based on perfect difference set {1, 3}.

- Chord edge: the edge connecting nonconsecutive vertices i and $i \pm s_j (\mathrm{mod}\, n)$, $2 \leq j \leq \delta$.
- Forward edges: for vertex i, the forward edges include the chord edge connecting vertices i and $i + s_j (\mathrm{mod}\, n)$ and the ring edge connecting vertices i and $i + s_1 (\mathrm{mod}\, n)$.
- Backward edges: for vertex i, the backward edges include the chord edge connecting vertices i and $i - s_j (\mathrm{mod}\, n)$ and the ring edge connecting vertices i and $i - s_1 (\mathrm{mod}\, n)$.

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.

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 degree 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 $n \cdot \delta = (\delta^2 + \delta + 1) \cdot \delta$.*

Proof. Since the connectivity of the PDG leads to a degree $d = 2\delta$, the total degree of vertices equals

$$\sum_{v \in V(G)} d(v) = n \cdot 2\delta.$$

By Proposition 1, $n \cdot 2\delta$ is equal to $2e$. Therefore, the total number of edges is equivalent to $n \cdot \delta = (\delta^2 + \delta + 1) \cdot \delta$. □

3.2 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 (SNC) [7] to broadcast content lookup queries among all the peers. In SNC, the source peer puts its address and a broadcast sequence number into a broadcast message, and then sends this message to all of its neighbors. Each peer maintains a list of the source addresses and sequence numbers of all the broadcast messages 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.

Superpeer overlay networks are similar to Gnutella in that the superpeers 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 transmission of redundant broadcast messages. As a result, the communication overhead within the network is inevitably higher than that in the ideal scenario in which each superpeer in the P2P network receives just one copy of the broadcast message.

3.3 Broadcasting over Superpeer Perfect Difference Graph Overlay Network

This study develops a PDG-based forwarding algorithm [9] in which the flooding messages are disseminated to all the superpeers 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 $\mathrm{TTL} = 2$ to all of its forward neighbors and sends a flooding message with $\mathrm{TTL} = 1$ to all of its backward neighbors.
- Step 2: If an intermediate vertex receives the message, it duplicates the message to all of its backward neighbors other than the neighbor from which it received the original message.

Fig. 2 presents a schematic illustration of the proposed PDG-based forwarding algorithm for a superpeer overlay network forming a PDG with an order of $\delta = 2$. In this example, it is assumed that superpeer 0 wishes to flood a lookup message to all the other superpeers in the network. In accordance with the two-step procedure described above,

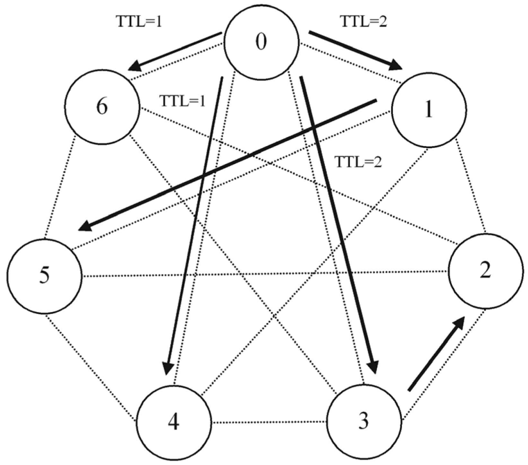

Fig. 2. Illustrative example of PDG-based forwarding algorithm.

superpeer 0 sends a flooding message with $\mathrm{TTL} = 1$ 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, superpeer 0 also sends a flooding message with $\mathrm{TTL} = 2$ 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 $\mathrm{TTL} = 0$ when decrementing the TTL parameter, and therefore, take no further action.

4 System Architecture and Construction

4.1 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 $i = 1, 2, 3, \ldots, m$, and V^i is a nonempty subset of vertices and includes at least one node, referred to as the superpeer node V_s^i. All the other nodes in V^i apart from the superpeer node are referred to as ordinary peers. The connections between the ordinary peers in V^i and the associated superpeer 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 henceforth as an intraconnection) only to its associated superpeer, i.e., the individual ordinary peers are not connected directly to one another.

Let $G' = (V', E')$ be an undirected graph with $V' \subset V$ and $E' \subset E$, where V is a set of superpeers and E' is a collection of connections between superpeers. Note that these connections are referred to as "interconnections" to distinguish them from the "intraconnections" in E^i between the ordinary nodes and the superpeer nodes. These connections 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 superpeer V_s^i is connected via undirected edges to the other superpeers $V_s^{(i \pm s_j)(\mathrm{mod}\ m)}$ for $1 \leq j \leq \delta$, 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 superpeers in G'.

In the hierarchical unstructured P2P system considered in this study, the ordinary peers communicate the indexes of their shared files to their associated superpeer via the intraconnections between them. If an ordinary peer wishes to search for an object, it issues a lookup request to its associated superpeer via its intraconnection. When the superpeer receives this lookup query, it performs 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 superpeers via its interconnection using the PDG-based forwarding algorithm described in Section 3.3.

4.2 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 BS server, provides new ordinary nodes joining the system with a randomly compiled list of superpeers, accepts or rejects a superpeer request, and maintains the superpeer overlay topology.

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

Any peer joining the P2P network and wishing to become a superpeer must first issue a request to the BS server. The peer should have a fast Internet connection such as an upload speed of 1 MB/s and a download speed of 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 superpeer, and sends the new peer the corresponding forward and backward connections, or registers the peer as a redundant superpeer, and provides the peer with a list of superpeers to connect to the system.

The BS server takes advantage of a superpeer table to control the superpeer overlay topology. Table 3 includes the vertex ID number, the superpeer IP address, the forward and backward connections of each superpeer, and the status of the vertex. Here, the ID number is simply the number of

TABLE 3
Example of Superpeer 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

the superpeer in the perfect difference overlay graph and is mapped to the IP address of the corresponding superpeer. Meanwhile, the forward and backward connection fields represent the IP addresses of the forward and backward neighbors of each superpeer, respectively. Finally, the status field contains a value of "1" if the superpeer 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 superpeer by the BS server.

Fig. 3 illustrates the scenario in which a new peer joins the superpeer overlay network. Note that in this figure, a superpeer overlay topology has already been constructed by the BS server and the superpeers form a perfect difference overlay graph with an order of $\delta = 2$. As indicated in the legend, the superpeers are represented by large open circles, while the ordinary peers are depicted as small black circles. In addition, the interconnections between the superpeers are shown using thick double-arrowheaded lines, while the intraconnections between the ordinary peers and the superpeers are shown using thin solid lines. In Fig. 3, the new joining peer (with an address IP_G) issues a request to become a superpeer. The BS server processes the peer request and then accepts the peer as a superpeer. The BS server adds IP_G in Table 3 and sends IP_G information, such as superpeer status, the corresponding forward connections of IP_A and IP_C, and backward connections of IP_D and IP_F.

The BS server manages the superpeer's request and maintains the overlay topology by a request process algorithm. A peer to join the P2P network as a superpeer issues a joining request, including the request type, its IP, and bandwidth description. A superpeer or redundant peer to leave the system issues a departing request consisted of the request type and the departing peer IP. The following sections discuss the details of the procedures performed at the BS server when superpeers join or leave the network prompting the requirement to extend 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 ℓ—the order of the successor PDS. Note that k and ℓ satisfy the constraint $k < \delta < \ell$. Finally, n is the total number of active and redundant superpeers.

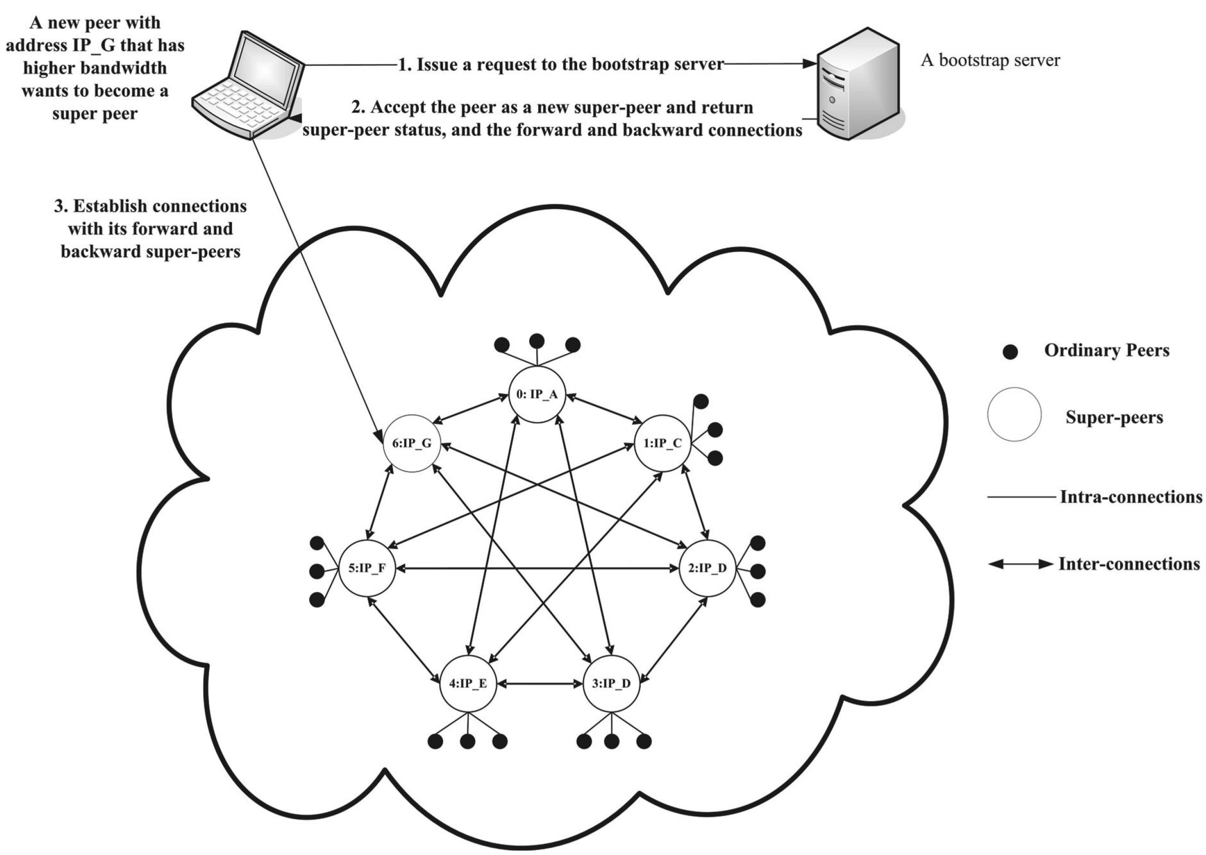

Fig. 3. Schematic illustration showing a new peer joining the superpeer overlay network.

4.3 Extension of Topology to Accommodate New Superpeers

In accordance with the request process algorithm, any peer with a fast Internet connection to enter the P2P network as a superpeer 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 superpeer and assigns the new peer the appropriate forward and backward connections.

When the number of superpeers is larger than the value $(\delta^2 + \delta + 1)$, it represents that all the positions in the PDG are already filled with active superpeers. If the requesting peer is qualified to become a superpeer, the BS server designates the peer as the role of a redundant superpeer, and is allowed to connect to the network by accessing a superpeer with a minimal response time selected from a list randomly compiled by the BS server.

When the number of superpeers and redundant superpeers increases to the threshold value, $1/2[(\delta^2 + \delta + 1) + (\ell^2 + \ell + 1)]$, there are a number of redundant superpeers existing in the system. In order to utilize the bandwidth capability of the redundant superpeers and increase system scalability, the order of the current PDS is enlarged to that of the successor PDS, and the superpeer 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 superpeer table. It then assigns the status of 1 to the new joining peer and all of redundant superpeers. Next, the BS server calculates and updates new forward and backward connections based on the new order δ in the superpeer table for these active superpeers. Finally, the BS server sends the new joining peer information, such as the status, the forward connections, and the backward connections. The BS server also notifies redundant superpeers about the status, the forward connections, and the backward connections, and informs the original active superpeers about the new forward and backward connections.

We illustrate an example to describe the overlay topology extension. In the initial setup phase (i.e., no superpeers have yet been identified), the BS server adopts a low-order PDS (i.e., an order of 2) to construct an initial superpeer overlay network for a maximum of 7 superpeers. Assume that there are 10 new peers wishing to become superpeers. Since the number of new peers exceeds the number of available spaces in the overlay network, the former seven peers are assigned as superpeers, and the remaining peers temporarily designated as redundant peers.

Later, when a new peer wishing to become a superpeer enters the system, it will result in the number of peers, including active superpeers, redundant superpeers, and the new joining peer, exceeding a threshold $10 (= (7 + 13)/2)$. The BS server, according to the request process algorithm, extends the superpeer overlay topology using a PDS with an order of 3, thereby allowing space for a maximum of 13 superpeers. Thus, the redundant superpeers and the new joining peer are assigned as new superpeers and are informed about the IP addresses of their forward and backward connections by the BS server. At this point, 11 active superpeers participate in the new enlarged topology.

4.4 Shrinking of Topology to Accommodate Departure of Existing Superpeers

When a superpeer 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 superpeers to take the place of the departing superpeer in the superpeer topology. After selecting a redundant superpeer, the BS server assigns it the vertex ID, the forward and backward connections of the departure superpeer, and the active status. The BS server then replaces the departure superpeer IP with the selected superpeer IP. Finally, the server informs the redundant peer to be an active superpeer and instructs those active peers infected by the superpeer departure to update their connection address records accordingly.

Algorithm. Request Process in the BS server
Input: Receiving a request(TYPE, IP, BW)
Output: Update the superpeer table in the BS server and return information to the superpeers
Initialize a superpeer 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 forward and backward connections based on the order of δ, and the status of 1 into the superpeer table;
//Accept the new joining peer as a new superpeer;
   Return the status, the forward connections and the backward connections to the new superpeer;
   else if $((\delta^2 + \delta + 1 < n \leq 1/2[(\delta^2 + \delta + 1) + (\ell^2 + \ell + 1)])$ **then**
   Assign the new joining peer a vertex ID, the peer IP, and the status of 0 into the superpeer table;
//Register the new joining peer as a redundant superpeer;
   Return the status and the addresses of some randomly selected superpeers 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 superpeer table;
   Assign the status of 1 to the new joining peer and all of redundant superpeers;
   Calculate and update new forward and backward connections based on the new order δ into the superpeer table for these active superpeers;
//Accept the peer as a new superpeer and extend the
//superpeer overlay topology;
   Return the status, the forward connections and the backward connections to the new superpeer;

Inform the redundant superpeers about the status, the forward connections, and the backward connections;
Inform the original superpeers about the new forward and backward connections;
else
Return a superpeer list containing the addresses of several randomly-selected superpeers 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 superpeer) **then**
if $(n \geq (\delta^2 + \delta + 1))$ **then**
Randomly select a new superpeer from the redundant superpeers;
Assign the vertex ID, the forward and backward connections of the departure superpeer, and the status of 1 to the selected superpeer;
Swap the departure superpeer IP of the superpeer table for the selected superpeer IP;
//Choose a redundant superpeer to become a new superpeer;
Inform the new superpeer about the status, the forward connections, and the backward connections;
Inform those active peers infected by the superpeer departing about the new superpeer 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 superpeer table for those peers infected by the superpeer departing;
Delete the records of the departing superpeer from the superpeer table;
//No redundant superpeers can replace the departing
//superpeer. Therefore, the BS server perform superpeer
//table updating process only;
Inform the messages of connections nonavailable to those active peers infected by the superpeer 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 superpeers;
Calculate and update new forward and backward connections based on the new order δ into the superpeer table for superpeers 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 superpeers with vertex IDs less than $\delta^2 + \delta + 1$ about the new forward and backward connections;
Inform all of superpeers with vertex IDs equal or greater than $\delta^2 + \delta + 1$ about the status and the addresses of some randomly selected superpeers;
else
Delete the records of the departing redundant superpeer from the superpeer table;
//The departure peer is a redundant superpeer;
end while

Having received a departure message from a superpeer wishing to disconnect from the P2P network, the ordinary peers then reconnect to the network by choosing one of these superpeers with the lowest response time in its superpeer list. The ordinary peers also can issue a request to the BS server to get a new superpeer list.

When the number of superpeers falls below the value $(\delta^2 + \delta + 1)$, there are not redundant superpeers available to replace the departing superpeers in the overlay topology. The BS server simply updates the corresponding forward and backward connections of the superpeer table for those peers infected by the superpeer departing. It then deletes the records of the departing superpeer from the superpeer table.

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

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

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

To illustrate an example to describe the overlay topology shrinking, consider 11 active superpeers existing in a superpeer overlay topology using a PDS with an order of 3. If one superpeer issues a departing request, the BS server, according to the request process algorithm, shrinks the topology since the number of superpeers equals the threshold $10(= (7+13)/2)$. The BS server adopts a low-order PDS (an order of 2) to shrink the current superpeer overlay network, thereby allowing space for maximum 7 superpeers. It assigns new vertex IDs to the remaining superpeers. The superpeers with vertex IDs less than 7 are active superpeers to participate in the reduced topology. Others are assigned as redundant superpeers. At this point, seven active superpeers and three redundant superpeers exist in the system.

5 Performance Evaluation

This section presents the results of a series of theoretical analyses designed to benchmark the performance of the proposed two-layer hierarchical unstructured P2P system against that of a random graph-based hierarchical unstructured P2P system. In performing the analyses, it is assumed that the PDG-based overlay network and the random-based overlay network have an identical number of superpeers, i.e., $\delta^2 + \delta + 1$, where δ is the number of elements in the PDS. The performance of the two schemes is quantified using the following metrics:

1. The degree of the superpeers—i.e., the number of interconnections that must be maintained by each superpeer.
2. The diameter of the topology—i.e., the maximum number of hops in the path between any pair of source-destination superpeers. It represents a maximum delay measured in hop counts across the superpeer overlay network on the lookup path length.
3. The number of messages incurred by the originator superpeer when flooding a lookup message to the other superpeers in the overlay topology represents the efficiency of the broadcasting protocol over the superpeer overlay network.
4. The average delay measured in hop counts—i.e., the average number of hop counts in the paths between the originator superpeer and all the other superpeers in the overlay topology when broadcasting a lookup query message.

Lemma 2. *The degree of the superpeers in the random-based and PDG-based overlay networks is given by* $\lceil \ln(\delta^2 + \delta + 1) \rceil$ *and* 2δ, *respectively.*

Proof. In [6], the authors investigated random graphs. An important property is concerned with graph connectivity. It was shown that provided a minimum of $\lceil (n/2)\ln(n) \rceil$ edges are maintained, a random graph will be connected with high probability. From Proposition 1 in Section 3.1, the average vertex degree in a random graph is given by $\lceil \ln(n) \rceil = \lceil \ln(\delta^2 + \delta + 1) \rceil$. Furthermore, from the definition of a PDG, the vertex degree is 2δ. □

Lemma 3. *The diameters of the superpeers in a random-based and PDG-based overlay network are equal to* $\frac{\Theta(\ln(\delta^2+\delta+1)}{\ln\ln(\delta^2+\delta+1))}$ *and 2, respectively.*

Proof. From [6], [30], if the number of edges is greater than or equal to $\lceil (n/2)\ln(n) \rceil$, the diameter of the random graph satisfies the following constraint:

$$\left\lfloor \frac{\ln n - \ln 2}{\ln \ln n} \right\rfloor + 1 \le \text{diameter} \le \left\lceil \frac{\ln n + 6}{\ln \ln n} \right\rceil + 3.$$

Therefore, the diameter of the random graph is $\Theta(\frac{\ln(\delta^2+\delta+1)}{\ln\ln(\delta^2+\delta+1)})$. Since any two vertices in a PDG are either connected by a connection directly or via a path of length 2 through an intermediate vertex, the diameter of a PDG is equal to 2. □

Lemma 4. *When a superpeer floods a lookup message over a random-based overlay network, the number of messages incurred by the SNC forwarding algorithm is equal to* $(\delta^2 + \delta + 1)(\lceil \ln(\delta^2 + \delta + 1) \rceil) - (\delta^2 + \delta)$.

Proof. Consider a random overlay network with a total of $\delta^2 + \delta + 1$ superpeers. From Lemma 2, the degree of the superpeers is equal to $\lceil \ln(\delta^2 + \delta + 1) \rceil$. When using the SNC forwarding algorithm, the superpeers duplicate the broadcast message and forward it to all their neighbors other than the neighbor from which they received the original message. Since there exist $\delta^2 + \delta$ intermediate superpeers in the overlay network and each superpeer duplicates $\lceil \ln(\delta^2 + \delta + 1) \rceil - 1$ messages to the other superpeers, the total number of messages incurred at the intermediate superpeers is equal to $(\delta^2 + \delta)(\lceil \ln(\delta^2 + \delta + 1) \rceil - 1)$. Moreover, the originator superpeer duplicates $\lceil \ln(\delta^2 + \delta + 1) \rceil$ messages. Thus, the total number of messages incurred by the originator superpeer in a random overlay network is equal to $(\delta^2 + \delta)(\lceil \ln(\delta^2 + \delta + 1) \rceil - 1)$ plus $\lceil \ln(\delta^2 + \delta + 1) \rceil$. □

Lemma 5. *When a superpeer floods a lookup message to the other superpeers in a PDG-based overlay network, the total number of messages incurred by the PDG-based forwarding algorithm is equal to* $\delta^2 + \delta$.

Proof. When a superpeer floods a lookup message to the other superpeers in a PDG-based overlay network, the message is first duplicated to the forward connections with $\mathrm{TTL} = 2$ and to the backward connections with $\mathrm{TTL} = 1$ according to step 1 in the procedure of the PDG forwarding algorithm (see Section 3.3). The number of messages transmitted to the backward neighbors is equal to 2δ, while that transmitted to the forward neighbors is equal to $\delta(\delta - 1)$. Thus, the total number of messages duplicated over the superpeer overlay topology is given by $2\delta + \delta(\delta - 1) = \delta^2 + \delta$. □

In discussing the average delay incurred by the SNC forwarding algorithm in a random-based overlay network, the following notations are introduced: T—the total delay, i.e., the sum of the delays incurred at each superpeer receiving the lookup message; n—the total number of superpeers; d—the degree of the overlay network; and D—the diameter of the overlay network.

Lemma 6. *The average delay incurred by the SNC forwarding algorithm across a random-based overlay network is equal to* $T/T(n-1)$, *where*

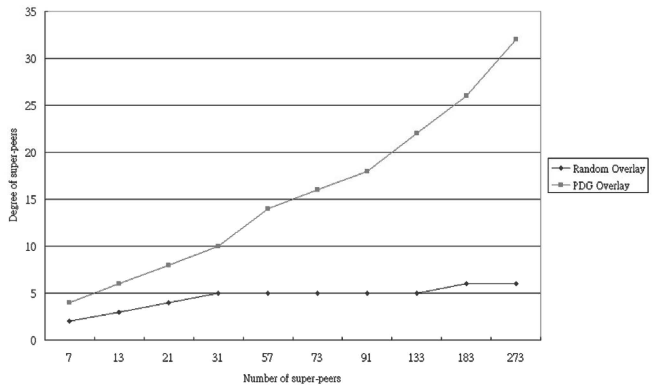

Fig. 4. Comparison of superpeer degree in random-based and PDG-based overlay networks.

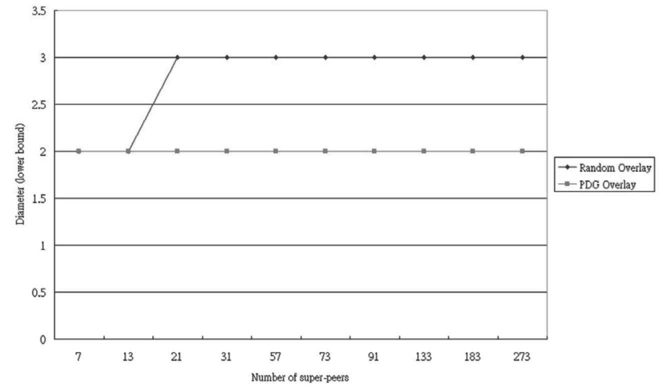

Fig. 5. Comparison of network diameter in random-based and PDG-based overlay networks.

$$T = \{d + 2d(d-1) + \cdots + (D-1)d(d-1)^{D-2} + D[n-1-d-d(d-1)-\cdots-d(d-1)^{D-2}]\}.$$

Proof. Assume that an originator superpeer in a random-based overlay network sends a broadcast message to each of its neighbors using the SNC forwarding algorithm. Upon receipt of the message, the superpeers duplicate it to all of their neighbors other than the neighbor that broadcasts the original message. Since the superpeers in the overlay network have a degree of d, a total of d superpeers of the neighbors of the originator superpeer receive the message with a delay of 1. Furthermore, since each neighbor of the originator superpeer duplicates and forwards the message to its own set of neighbors, a total of $d(d-1)$ superpeers receive the message with a delay of 2, and so on. Ultimately, a total of $[n-1-d-d(d-1)-\cdots-d(d-1)^{D-2}]$ superpeers receive the message with a delay of D. Thus, the total delay incurred by all the superpeers in the random-based overlay network can be generalized as follows:

$$T = \{d + 2d(d-1) + \cdots + (D-1)d(d-1)^{D-2} + D[n-1-d-d(d-1)-\cdots-d(d-1)^{D-2}]\}.$$

The average delay equals $T/(n-1)$. □

Lemma 7. *The average delay caused by the PDG forwarding algorithm across the perfect difference overlay network is equal to* $2\delta/(\delta+1)$.

Proof. When using the PDG-based forwarding algorithm, the originator superpeer sends broadcast messages to its δ backward connections with $\mathrm{TTL} = 1$ and to its δ forward connections with $\mathrm{TTL} = 2$. Therefore, a total of 2δ superpeers receive the message with a delay of 1. In addition, a total of $\delta(\delta-1)$ superpeers receive the message with a delay of 2. Thus, the total delay in the PDG-based overlay network is equal to $2\delta + 2\delta(\delta-1) = 2\delta^2$ and the average delay is equal to $2\delta^2/(\delta^2+\delta) = 2\delta/(\delta+1)$. □

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

Although Lemma 3 presents 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 superpeers. 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 shown in Lemma 5, each superpeer in the PDG-based overlay topology receives just one copy of the broadcast message when the originator superpeer utilizes the PDG-based forwarding algorithm to flood a lookup query. By contrast, Lemma 4 shows that in the random-based overlay network using the SNC forwarding algorithm, the number of broadcast messages received by each superpeer varies as a power law function of δ. Fig. 6 illustrates the variation of the number of broadcast messages with the number of superpeers in the PDG-based and random-based overlay networks, respectively. The results clearly demonstrate that the PDG-based forwarding algorithm generates significantly

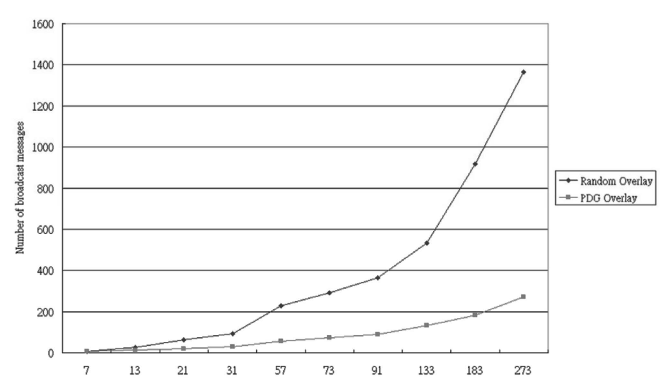

Fig. 6. Comparison of number of flooding lookup messages incurred in random-based and PDG-based overlay networks.

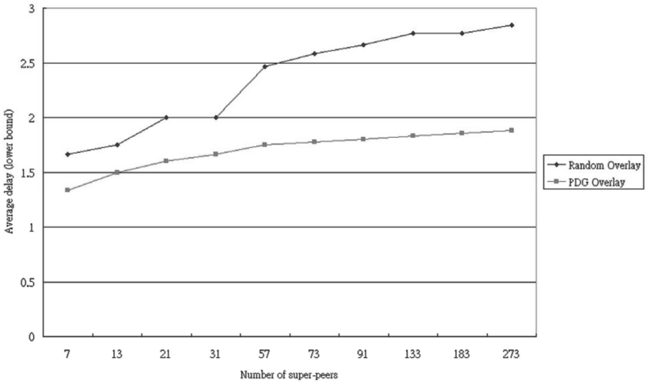

Fig. 7. Comparison of average delay (measured in hop counts) in random-based and PDG-based overlay networks.

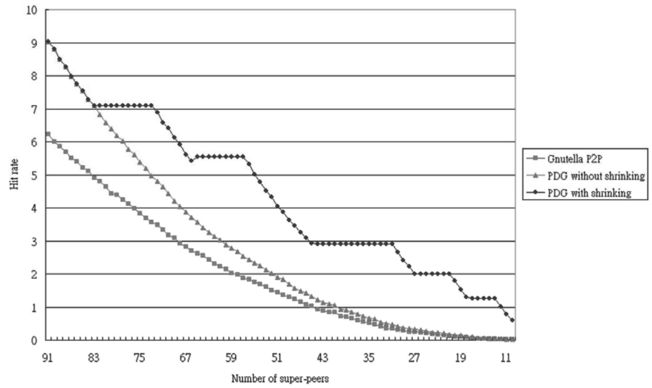

Fig. 8. Comparison of hit rate in Gnutella P2P and PDG-based overlay networks.

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 superpeer topology increases.

Fig. 7 illustrates the variation of the average flooding delay in the random-based and PDG-based overlay networks as a function of the number of superpeers. 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 superpeer overlay network increases. Specifically, it is observed that as the number of superpeers increases toward 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 superpeer is far lower than the total number of superpeers 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 superpeer overlay topology in terms of a reduced number of broadcast messages and a lower average hop-count delay. Moreover, the diameter of the PDG-based overlay network is smaller than that of the random-based superpeer topology. However, the results have shown that these performance improvements are obtained at the expense of a greater superpeer maintenance cost.

6 Implementation

To evaluate the file transfer performance of the proposed two-layer hierarchical unstructured P2P system using a PDG, we implemented a prototype superpeer and BS server incorporating the request process algorithm presented in Section 4 on our testbed [36]. 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 superpeer 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 superpeers. 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 superpeers equals the shrinking threshold value (e.g., 10). In the beginning of each round, each superpeer 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 $\mathrm{TTL} = 2$, respectively. When each round terminates on the condition that each search request is serviced, one randomly selected superpeer leaves the system and the other active superpeers then enter next round to issue new lookup queries.

Three thousand files with 1,000 types are uniformly allocated to the 91 nodes. The content popularity of file types follows a Zipf-like distribution, where the relative probability of a lookup request for the ith most popular file is proportional to $1/i^{\alpha}$, where α typically takes on some value less than unity. The lookup query distribution indicates how many requests are made for each file (e.g., popular files get many more request than unpopular ones). The query distribution also follows the Zipf-like distribution. In the experiment, let α be 0.82 and each superpeer issues 500 lookup queries per round.

The experimental scenario focuses on observing dynamic departure effects and tracks changes in the PDG-overlay topology to evaluate the behavior of the topology shrinking process. Fig. 8 illustrates the variation of hit rate with the number of superpeers in the Gnutella P2P, PDG-based without shrinking, and PDG-based with the proposed shrinking process overlay, respectively. The shrinking process is fast and the results clearly demonstrate that the hit rate of the PDG-based overlay with the proposed shrinking process is significantly higher than those of the Gnutella P2P and PDG-based overlay without shrinking.

In the first round, since each overlay topology is a complete and connected graph for these overlay topologies, the hit rate achieves the highest value. Moreover, since the lookup queries on the PDG-based overlay can be efficiently flooded to each superpeer, 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.

Since superpeers leaving the system will result in the remaining superpeers losing their neighboring connections whether in the Gnutella P2P or PDG-based overlay, some lookup queries cannot be forwarded to other superpeers. The hit rate will decrease with the number of superpeers decreasing. However, the BS server in the proposed PDG-based overlay will adapt the superpeer overlay topology by the proposed shrinking process when the number of superpeers decreases to the threshold value described in Section 4.4. For the experimental results, the BS server will shrink the overlay topology when the number of superpeers equals 82, 65, 44, 26, 17, and 10. At these points, the PDG-based overlay is a complete perfect difference graph. The lookup queries will be efficiently forwarded to each superpeer. Later, when an active superpeer leaves the system, the BS server will assign another redundant superpeer replacing the departing superpeer to keep the completeness of the PDG-based overlay topology. Therefore, the hit rate can be kept in a high level despite superpeer departure.

7 Discussion

To prevent abnormal superpeer departure, the system generates communication overheads due to superpeers periodically sending each other hello messages to maintain the status of their forward and backward connections. In addition, the system incurs an inevitable communication cost during the process of expanding or shrinking the overlay topology to accommodate new or departing superpeers, respectively. For example, when expanding the current PDG based on a PDS with an order of δ to a PDG based on a PDS with an order of ℓ, the bootstrap server must inform all the superpeers within the new overlay topology of their respective statuses and forward and backward connections. Thus, from Lemma 1, a communication overhead of $O(\ell^3)$ is incurred. Similarly, the action of shrinking the overlay network to a PDG based on the predecessor PDS with an order k incurs a total communication overhead of $O(k^3)$.

In the superpeer overlay topology construction process proposed in this study, the entry of a new superpeer or ordinary peer to the network is controlled by a single bootstrap server at the point of entry. Thus, the system is prone to single-point failures. This problem can be resolved by utilizing multiple bootstrap servers to process the registration requests of the new joining peers. However, the task of developing the synchronization mechanism required to ensure the consistency of the superpeer table maintained by the individual bootstrap servers falls outside the scope of the present study.

8 Conclusion

This paper has presented an efficient technique for constructing and maintaining the superpeer overlay topology of a two-layer hierarchical unstructured P2P system using a 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 superpeer overlay topology based on a perfect difference graph has been benchmarked against a superpeer overlay topology based on a random graph using the SNC forwarding algorithm. The theoretical results have shown 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.

Acknowledgments

This work was supported in part by the National Science Council under the Grants NSC 96-2219-E-006-009 and NSC 97-2221-E-006-092. The authors are also grateful to the anonymous reviewers for their useful comments and suggestions.

References

[1] *Gnutella—A Protocol for Revolution,* http://rfc-gnutella.sourceforge.net.com/, 2008.
[2] *KaZaA,* http://www.kazaa.com/, 2009.
[3] *Overnet/edonkey2000,* http://www.edonkey2000.com/, 2000.
[4] *Bittorrent,* http://bitconjurer.org/BitTorrent/, 2003.
[5] D.B. West, *Introduction to Graph Theory.* Prentice-Hall, Inc., 1996.
[6] B. Bollobás, *Random Graphs.* Academic Press, 1985.
[7] J.F. Kurose and K.W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet,* third ed. Addison Wesley.
[8] B. Parhami and M. Rakov, "Perfect Difference Networks and Related Interconnection Structures for Parallel and Distributed Systems," *IEEE Trans. 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. Parallel and Distributed Systems,* vol. 16, no. 8, pp. 725-736, Aug. 2005.
[10] L. Xiao, Z. Zhuang, and Y. Liu, "Dynamic Layer Management in Superpeer Architectures," *IEEE Trans. Parallel and Distributed Systems,* vol. 16, no. 11, pp. 1078-1091, Nov. 2005.
[11] Y. Dalal and R. Metcalfe, "Reverse Path Forwarding of Broadcast Packets," *Comm. ACM,* vol. 21, no. 12, pp. 1040-1048, Dec. 1978.
[12] I. Stoica et al., "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications," *IEEE/ACM Trans. Networking,* vol. 11, no. 1, pp. 17-32, Feb. 2003.
[13] R.G. Gallager, P.A. Humblet, and P.M. Spira, "A Distributed Algorithm for Minimum Weight-Spanning Trees," *ACM Trans. Programming Languages and Systems,* pp. 66-77, Jan. 1983.
[14] F.C. Gartner, "A Survey of Self-Stabilizing Spanning-Tree Construction Algorithms," Technical Report IC/2003/38, Swiss Fed. Inst. of Technology, School of Computer and Comm. Sciences, June 2003.
[15] J. Yan, Y. Yang, and G.K. Raikundalia, "A SwinDeW p2p-Based Decentralized Workflow Management System," *IEEE Trans. Systems, Man and Cybernetics, Part A,* vol. 36, no. 5, pp. 922-935, Sept. 2006.
[16] L.D. Baumert, "Cyclic Difference Sets," *Lecture Notes in Math.,* vol. 182, Springer-Verlag, 1971.
[17] T.P. Kirkman, "On the Perfect r-Partitions of $r^2 + r + 1$," *Trans. Historical Soc. Lancashire and Cheshire,* vol. 9, pp. 127-142, 1857.

[18] R.K. Guy, *Unsolved Problems in Number Theory*, second ed., pp. 118-121, Springer-Verlag, 1994.
[19] A. Rowstron and P. Druschel, "Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems," *Proc. Conf. Middleware*, 2001.
[20] B.Y. Zhao et al., "Tapestry: A Resilient Global-Scale Overlay for Service Deployment," *IEEE J. Selected Areas Comm.*, 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 Comm. Surveys and Tutorials*, vol. 7, pp. 72-93, 2005.
[22] C. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and Replication in Unstructured Peer-to-Peer Networks," *Proc. Int'l Conf. Supercomputing (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, "A Local Search Mechanism for Peer-to-Peer Networks," *Proc. Conf. Information and Knowledge Management (CIKM)*, 2002.
[25] D. Tsoumakos and N. Roussopoulos, "Adaptive Probabilistic Search for Peer-to-Peer Networks," *Proc. Third IEEE Int'l Conf. P2P Computing*, 2003.
[26] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti, "A Local Search Mechanism for Peer-to-Peer Networks," *Proc. Conf. Information and Knowledge Management (CIKM)*, 2002.
[27] J. Chu, K. Labonte, and B. Levine, "Availability and Locality Measurements of Peer-to-Peer File Systems," *Proc. SPIE*, 2002.
[28] B. Yang and H. Garcia-Molina, "Designing a Super-Peer Network," *Proc. 19th Int'l Conf. Data Eng.*, Mar. 2003.
[29] K. Watanabe, N. Hayashibara, and M. Takizawa, "Performance Analysis of the Superpeer-Based Two-Layer P2P Overlay Network with the CBF Strategy," *Proc. 27th Int'l Conf. Distributed Computing Systems Workshops (ICDCSW '07)*, pp. 32-32, June 2007.
[30] Y.J. Pyun and D.S. Reeves, "Constructing a Balanced, (log(N)/loglog(N))-Diameter Super-Peer Topology for Scalable P2P Systems," *Proc. Fourth Int'l Conf. Peer-to-Peer Computing*, pp. 210-218, Aug. 2004.
[31] A. Montresor, "A Robust Protocol for Building Superpeer Overlay Topologies," *Proc. Fourth Int'l Conf. Peer-to-Peer Computing*, pp. 25-27, Aug. 2004.
[32] A.T. Mizrak, Y. Cheng, V. Kumar, and S. Savage, "Structured Superpeers: Leveraging Heterogeneity to Provide Constant-Time Lookup," *Proc. Third IEEE Workshop Internet Applications (WIAPP '03)*, pp. 104-111, June 2003.
[33] L. Jenn-Wei, Y. Ming-Feng, H. Chin-Yu, and L. Chu-Ti, "Reliable Hierarchical Peer-to-Peer File Sharing Systems," *Proc. IEEE Region 10 Conf. (TENCON '06)*, pp. 1-4, Nov. 2006.
[34] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, "Making Gnutella-Like P2P Systems Scalable," *Proc. ACM SIGCOMM*, 2003.
[35] S. Han and S. Park, "A Dynamic Layer Management Scheme for a Superpeer Ring with a Loosely Consistent DHT," *Proc. IEEE Pacific Rim Conf. Comm., Computers, and Signal Processing*, pp. 383-386, Aug. 2007.
[36] "Testbed@TWISC—Providing Integrated Access to a Wide Range of Experimental Environments: From Simulated to Emulated to Wide-Area Network Testbeds," http://testbed.ncku.edu.tw, 2009.

Jung-Shian Li is a professor in the Department of Electrical Engineering, Institute of Computer and Communication Engineering, National Cheng Kung University, Taiwan. He graduated from the National Taiwan University, Taiwan, with BS and MS degrees in electrical engineering. He received the PhD degree from the Technical University of Berlin, Germany. He teaches communication courses and his research interests include wired and wireless network protocol design, network security, and network management. He is currently involved in funded research projects dealing with network security testbed, common criteria, intrusion prevention system, router active queue management, VoIP security, and P2P architectures. He is the division director of computer and network center, NCKU. He serves on the editorial boards of the *International Journal of Communication Systems* and is guest editor for wireless medium access control in *IEEE Wireless Communication* magazine. He was a founding program chair of the International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE) 2005 and the International Workshop on Forensics for Future Generation Communication Environments 2007 (F2GC-07). He is a member of the IEEE.

Chih-Hung Chao received the BS degree from the Department of Electronic Engineering at the Feng Chia University, Taiwan, and the MS degree from the Department of Electrical Engineering, National Cheng Kung University, Taiwan. He received the PhD degree from the Department of Electrical Engineering, Institute of Computer and Communication Engineering, National Cheng Kung University. His research interests include wireless and mobile network protocols, peer-to-peer computing, and communications. He is a member of the IEEE.