Efficient formless Peer to Peer Information in superimpose Networks

[1]Chinmaya Kumar Nayak, [2]Sambit Pattnaik, [3]Swagatika Tripathy, [4]Bimal Prasad Kar
[1]Assistant Professor, [2]Assistant Professor, [3]Assistant Professor, [4]Assistant Professor
[1,2,3,4]Department of Computer Science & Engineering,
[1,2,3,4]Gandhi Institute for Technological Advancement(GITA),Bhubaneswar,Odisha,India

Abstract: Peer-to-peer (P2P) formless networks are in vogue in a massive network. These networks can be built very efficiently without specific rules and, therefore, are considered appropriate for the online environment. But the scanning techniques used in these networks are usually compromised with a large network size. To optimize search queries in unstructured P2P networks, use the connection between network participants. The search protocol was designed to accelerate the search process through self-organized P2P networks in a small world. Both theoretical and experimental analyzes are carried out, which show the effectiveness and effectiveness of this approach. Based on the file exchange model, which shows the power-right property, we present a query construction algorithm, in which any non-structured G (V, E) P2P network, where V is a set of participating counterparts, and E is a set of superposed connections with even elements in V. The equations in G can be combined with each other in a random way. The objective is to restructure G with the following characteristics: 1. High grouping, 2. Low diameter, 3. Progressive.

***Index Terms* - Peer to Peer, Optimizing, DHT, formless.**

I. Introduction

Peer-to-peer (P2P) networks have been widely deployed in the internet [1]. These networks are largely classified into two categories, namely Structured Peer to peer and Unstructured Peer to Peer. The structured P2P networks, constructed based on distributed hash tables (DHT)[2]. Unstructured peer to peer networks based on random search strategies (e.g., flooding). Without imposing any stringent constraints over the network topology. Unstructured P2P networks can be constructed very efficiently and have attracted far more practical use in the internet. They provide many services like file sharing, information retrieval and media streaming. P2P applications are popular because they provide low entry barriers. Prior studies (e.g.,[3]) reveal that P2P services may dominate up to around 20 percent of internet traffic. Object search is the essential building block in several P2P applications (e.g., file sharing, media streaming). Gnutella is the popular P2Psearch protocol, because Gnutella networks are unstructured and the peers participating in the network are connected to one another randomly as shown in Fig.1, and the peers search the objects in the networks through message flooding. The basic idea of this paper is that the statistical patterns over locally shared resources of a peer can be explored to guide the distributed resource discovery process and therefore enhance the overall resource discovery performance in unstructured peer to peer networks.

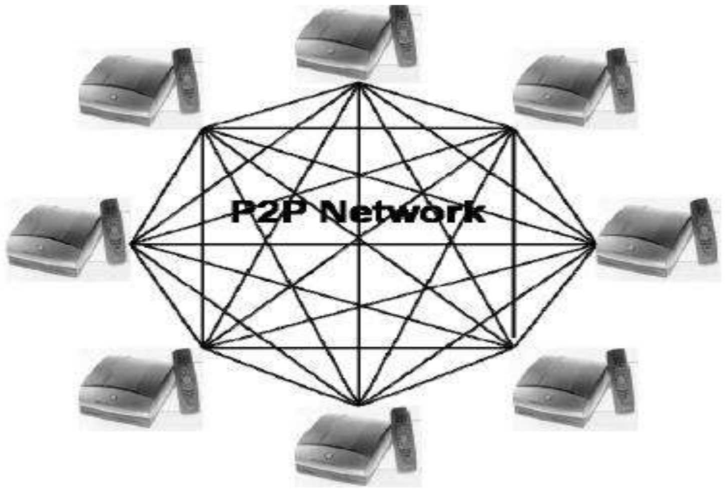


Fig.1.1 Formless Peer to Peer Network

II. Background Information

This section outlines our data model and query language, and describes the construction and properties of SONs. For more detail, we refer the interested reader to [6,18,2,20].

2.1 Data model and query language

We utilize the Vector Space Model (VSM) to represent documents, queries, peer and network descriptions. In our setting a resource is any piece of information that can be described by a set of keywords, such as a text document which is characterized by its terms, an image, which is often associated to a set of tags, or an mp3 file. We associate a weight to each keyword so as to represent the importance of the keyword as a description for the given resource. A query is a set of keywords, for which the weights are explicitly assigned by the user or implicitly by the system (e.g., through relevance feedback techniques [31]). A resource is characterized by a resource description ri that is a vector containing keywords and weights for these keywords. Similarly, the profile of a peer P, profile(P), is computed using the descriptions r1; : : : ; rm of the resources stored at this peer.

A standard technique to decide which resource description r best matches a given query q, is to utilize a similarity function sim(q; r), which assigns a numerical score to each pair (q; r). The scores corresponding to different resource descriptions are then compared to derive a relevance ranking with respect to query q. A common similarity measure in IR is the cosine of the angle formed by the vector representations of q and r. Notice that, in practice, any function that models the similarity between a resource and a query can be used. For example, in the case of a social network the similarity function could also contain a social component, that also considers the strength of relations among users.

2.2 Semantic overlay networks

In a SON each peer P performs a variable number of random meetings with other peers in the network, during which they exchange their profiles and compute their similarity. Based on the similarity function sim (), a peer P establishes two types of links:

– Short-distance links towards the k most similar peers in the network discovered through the random meeting process. The number of short-distance links k is usually small (e.g., O(logN), where N is the number of peers in the network).

– Long-distance links towards k0 (typically k0 < k) peers chosen at random from the rest of the network.

To maintain short-distance links up-to-date and ensure the clustering property of the SON, a periodic rewiring procedure [6,18,2,20] is executed by all peers, aiming at discovering new more similar peers, or refreshing links that have become outdated due to network dynamics. Long-distance links, usually updated using random walks, are necessary to avoid creating tightly clustered groups of peers that are disconnected from the rest of the network. Query answering in SONs benefits from the fact that peers containing related information are directly linked or at a short-hop distance from each other. Thus, the task of finding a peer that can answer a query q reduces to locating the

appropriate cluster of peers. Once a peer in the appropriate cluster is reached (i.e $sim(profile(P),q)) \geq \beta$) _ b, where b is called broadcast threshold), the query goes through a limited broadcast using short-distance links aiming at reaching all neighbors of P. Due to the SON properties, these peers are able to answer q with high probability.

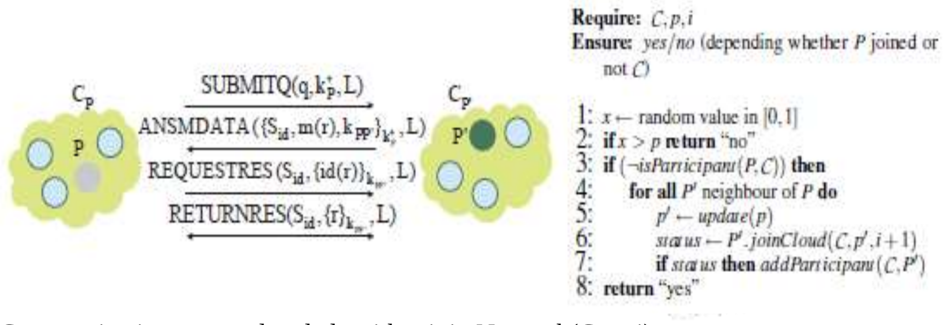
III. The Networks protocols

This section describes in detail the protocols that regulate the interactions between peers and allow them to anonymously share and retrieve resources available in the network.

3.1 Protocol overview

The key principle behind the anonymity mechanism of Networks is to cloak both the querying peer and the resource provider behind a group of neighboring peers, called network. Peers generate networks at random, without necessarily using them, to minimize the correlation between the events of joining and using a network. Additionally, they nondeterministic ally decide to participate or not in networks created by other peers. Networks are created using the short-distance links of peers and are thus populated by peers in the neighborhood of the network initiator. Communication takes place between networks, and all peers in a network share the same probability of being involved in any communication which has this network as the start- or end-point (also known as k-anonymity [28]). To avoid correlation of roles in the protocol with specific actions, which would compromise anonymity, the protocol is designed so that the observable behavior is the same for all peers, regardless of them being initiators of forwarders of a message.

The proposed cryptographic protocol aims at addressing the problem of censorship at the communication level, where a malicious party aims at filtering out any communication that contains unwanted content (either a query or a resource). The secrecy of the resource is protected by cryptography, making it hard for the attacker to censor the communication based on an inspection of the message content. The design of such a protocol is conceptually challenging since we do not assume previously shared secrets or centralized infrastructures. The protocol is composed of four steps summarized in Figure 3. A querying peer chooses a network it participates in to issue a query. The query follows a random walk in the network to obscure the message initiator, leaves the network from multiple peers to


C_P
P
$C_{P'}$
P'
SUBMITQ(q, k_P^+, L)
ANSMDATA$(\{S_{id}, m(r), k_{PP'}\}_{k_P^+}, L)$
REQUESTRES$(S_{id}, \{id(r)\}_{k_{PP'}}, L)$
RETURNRES$(S_{id}, \{r\}_{k_{PP'}}, L)$

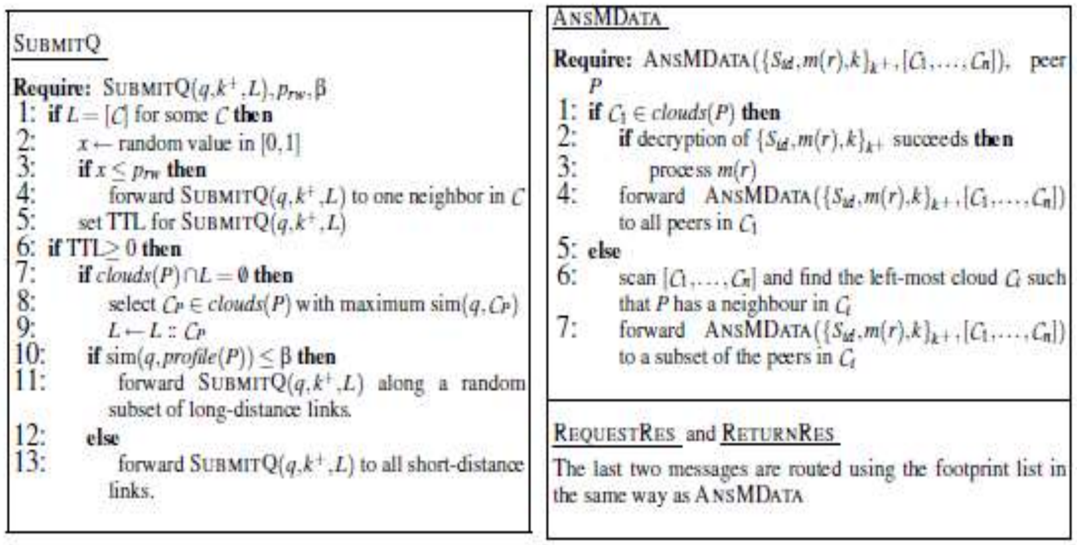
```
Require: $C, p, i$
Ensure: $yes/no$ (depending whether $P$ joined or
    not $C$)

$x \leftarrow$ random value in $[0, 1]$
if $x > p$ return "no"
if $(\neg isParticipant(P, C))$ then
    for all $P'$ neighbour of $P$ do
        $p' \leftarrow update(p)$
        $status \leftarrow P'.joinCloud(C, p', i+1)$
        if $status$ then $addParticipant(C, P')$
return "yes"
```

Fig. 3.1. Communication protocol and algorithm join Network(C, p, i).

ensure higher resistance to censorship and is routed towards a region in the network that possibly contains matching resources. A footprint list is used to collect the list of traversed networks and facilitates the routing of the subsequent messages. A responder to the query encrypts the answer with a public key received with the query message and routes it towards the network of the querying peer, as specified in the footprint list. All subsequent messages between the querying peer and the responder will have a network as a destination and, when this network is reached, the message will be broadcasted to reach the intended recipient. Finally, notice that the query message does not contain any session identifier which would connect the query to the subsequent messages. In the last two protocol steps, however, a session identifier is used to avoid costly decryption checks, since the message content is encrypted.

3.2 Network creation

The design of a network creation algorithm should satisfy some fundamental properties, such as randomness, tenability and locality, in order to reveal as little information as possible to potential attackers while maintaining the useful clustering properties of the underlying SON. According to our network creation algorithm, when a peer P generates a new network C, it selects the participants among its neighbors utilizing short-distance links. This guarantees that networks are populated by semantically close peers (network locality). As shown in Section 4.1, this property is crucial to guarantee that network intersections have high cardinality, thus preventing intersection attacks that aim at breaking anonymity. Peers that join a network in turn select other neighbors, and the protocol is executed in a recursive way with decreasing probability to join C. The join Network() algorithm shown in Figure 1 shows this procedure from P's point of view, assuming that P has received a join Network(C; p; i) message that may have been generated either by itself or by any other peer. This message specifies the network C, the probability p to join it, and the step i. With probability 1∈ p, P replies negatively (line 2) to the request, otherwise it accepts to join C (line 8). In this case, if P is not already in C, it triggers a recursive procedure in its neighbours (line 3) by sending a join Network(C; p0; i+1) message to each of them (lines 4-7) with join probability p0 (line 5). Finally, the peers joining C are marked by P as neighbours in C (lines 6-7). The update(p) function used to obtain p0 is the means to control network population and to offer tenability between anonymity and efficiency. For simplicity, in



```
SUBMITQ
Require: SUBMITQ($q,k^+,L$), $p_{rw}$, $\beta$
1: if $L = [C]$ for some $C$ then
2:     $x \leftarrow$ random value in $[0,1]$
3:     if $x \le p_{rw}$ then
4:         forward SUBMITQ($q,k^+,L$) to one neighbor in $C$
5:     set TTL for SUBMITQ($q,k^+,L$)
6: if TTL $\ge 0$ then
7:     if $clouds(P) \cap L = \emptyset$ then
8:         select $C_P \in clouds(P)$ with maximum sim($q, C_P$)
9:         $L \leftarrow L :: C_P$
10:     if sim($q, profile(P)$) $\le \beta$ then
11:         forward SUBMITQ($q,k^+,L$) along a random
            subset of long-distance links.
12:     else
13:         forward SUBMITQ($q,k^+,L$) to all short-distance
            links.
```

```
ANSMDATA
Require: ANSMDATA($\{S_{id}, m(r), k\}_{k+}, [C_1, \ldots, C_n]$), peer
    $P$
1: if $C_1 \in clouds(P)$ then
2:     if decryption of $\{S_{id}, m(r), k\}_{k+}$ succeeds then
3:         process $m(r)$
4:     forward ANSMDATA($\{S_{id}, m(r), k\}_{k+}, [C_1, \ldots, C_n]$)
       to all peers in $C_1$
5: else
6:     scan $[C_1, \ldots, C_n]$ and find the left-most cloud $C_i$ such
       that $P$ has a neighbour in $C_i$
7:     forward ANSMDATA($\{S_{id}, m(r), k\}_{k+}, [C_1, \ldots, C_n]$)
       to a subset of the peers in $C_i$
```

REQUESTRES and RETURNRES

The last two messages are routed using the footprint list in the same way as ANSMDATA

Fig. 3.2. Behavior of peer P in the different protocol steps.

the experiments we consider the same function throughout the network, but in practice each peer may use its own update() function. Notice that peers only know the neighbours certainly belonging to their networks (i.e., the neighbours that have sent or have positively answered to a join Network request). No information can be derived from a negative answer, since a peer already belonging to the network answers negatively with probability 1∈ p. This helps to avoid statistical attacks on network membership.

3.3 Query routing

In this section, we present the routing algorithm used to route a query q from a peer P to a resource provider P0. Figure 2 gives the pseudo code for the query routing procedure followed by any peer P. Notice that the protocol is the same regardless of P being the initiator or the forwarder of the message, in order to avoid breaches in anonymity.
When P wants to issue a query q, it constructs a message msg=SUBMITQ(q;k+ P ;L= [CP]), where k+ P is a public key generated by P especially for this session, and L is the footprint list that will be used to collect the list of networks msg will traverse during the routing. P initializes this list with one of its networks. The collection of networks in the footprint list is performed as follows. A peer Pi that receives msg checks whether one of the networks it participates in is already listed in L. If not, it chooses a network CPi and appends it to L (i.e., L L :: CPi). This information will be exploited by the successive phases of the communication protocol to optimize routing between P and the resource provider.

The routing algorithm for a SUBMITQ message is reported in Figure 2. The algorithm consists of two steps: an intra-network routing, during which the message msg performs a random walk in C, and an inter-network routing, during which msg is delivered to a peer P0 not participating in C. Each peer receiving (or creating) msg, forwards it to a random peer participating in C with probability prw and also to a peer that subsequently enters the inter-network phase. The intra-network routing phase is necessary to avoid revealing the identity of the query initiator to a malicious long-distance neighbor that exploits the existence of a single network C in L. By adding the random walk phase within C, an attacker cannot know whether P is the initiator of msg or simply a forwarder entering into the inter-network phase.

Inter-network routing is based on the fireworks query routing algorithm [17]. A peer P receiving message msg omputes the similarity sim(q;profile(P)) between q and its profile. If $sim(q,profile(P)) \leq \beta,$) _ b, where b is the broadcast threshold, this means that neither P, nor P's neighbours are suitable to answer q. Thus, msg is forwarded to a (small) subset of P's long-distance links. If sim(q;profile(P))>b, the query has reached a neighborhood of peers that are likely to have relevant resources and msg goes through a limited broadcast using short-distance links.

In order to limit the network traffic, each message entering the inter-network phase is associated to a time-to-live (TTL), which is updated at every hop. Message forwarding is stopped when TTL reaches zero. Finally, all peers maintain a message history and use it to discard already processed messages.

IV. ATTACK SCENARIOS

In this section, we introduce the attacks that might in principle break anonymity and censorship-resistance in our framework.

4.1 Attacks on anonymity

Surrounding Attack. Assume that a malicious peer PAdv generates a fake network CAdv and sends a message JoinNetwork(CAdv; p; i) to P, and assume that P accepts to join network CAdv. If i is the last step of the join algorithm, P does not have any other neighbours in CAdv. The anonymity of P is thus compromised since PAdv can monitor all P's activities in CAdv. This attack can be generalized by considering a population of colluding malicious peers trying to surround P. They block all the JoinNetwork messages received from honest peers and instead send P messages of the form JoinNetwork (CAdv; p; i), where I is one of the last steps of the joining algorithm. Notice that the risk of incurring in surrounding not only depends on the topology of the network but also on its physical implementation. In a wireless scenario, for example, it is very unlikely that a malicious peer is able to gain exclusive control of the communication channel of another peer, which is constituted by its surrounding atmosphere. To mitigate the threat of this attack. Intersection Attack. Since peers participate in different networks, a malicious peer might try to "guess" the identity of the querying peer based on the (even partial) information that it has available about the intersection of two networks. Notice that computing network intersections is difficult because network topology is not known in general and a malicious peer only knows its neighbors. Remember that networks are generated using only short-distance links and that networks are thus confined in a small region of the network (locality property).

4.2 Attacks on censorship resistance

Blocking Attack. The blocking attack aims at blocking (instead of monitoring) all SUBMITQ messages containing undesired queries and, tracking the network C of the querying peer, at subsequently blocking all ANSMDATA messages directed to C. Note that our query routing mechanism guarantees that SUBMITQ is replicated and routed through different paths: this redundancy helps to overcome the blocking attack, since it requires the attackers to be located either in all the paths followed by SUBMITQ message or in one of the paths followed by SUBMITQ message and all the paths followed by ANSMDATA message.

In the following we present a theoretical characterization of the resistance to blocking attacks. This is formalized as the probability that a communication which an adversary is willing to censor will not be blocked.

The probabilities that all SUBMITQ and all ANSMDATA messages are blocked is given by:

$$\Pr\{block(\text{SUBMITQ})\} = [\Pr\{A_1\}]^{P1}$$
$$\Pr\{block(\text{ANSMDATA})\} = \Pr\{A_1\} \times [\Pr\{A_2\}]^{P2}$$

where p1 (resp. p2) is the number of distinct paths followed by the first (resp. second) message and A1 (resp. A2) denotes the event that at least one attacker resides in one of the paths of SUBMITQ (resp. ANSMDATA). For implicitly, we have assumed these events to be independent from each other and from the specific path followed by
the messages. If the attackers are randomly distributed in the network, the probability associated to events A1 and A2 is:

$$\Pr\{A_1\} = \Pr\{A_2\} = 1 - \binom{N-k}{m} / \binom{N}{m}$$

Where N is the number of peers in the network, k is the number of adversaries, and m is the average number of peers in a path connecting P to P0 (where P and P0 are not counted). The fraction in Equation 3 represents the number of safe (i.e., not including any attacker) paths divided by the total number of paths.

Finally, the degree of censorship resistance can be computed as

$1 - \Pr\{block(\textsc{SubmitQ}) \vee block(\textsc{AnsMData})\}$ which can be obtained as the combination of the two non-independent events. Man-in-the-middle Attack. In the man-in-the-middle attack (MITM), the attacker intercepts the SUBMITQ(q;k+ P ; [CP; : : :]) message sent by P and replaces it by a freshly generated message SUBMITQ(q;k+ Adv; [CAdv]), where k+ Adv and CAdv are the adversary's public key and network, respectively. The adversary then runs two sessions of the protocol, one with the querying peer P and one with provider R, pretending to be the query responder and initiator respectively. This allows the attacker to filter and possibly interrupt the communication between P and R. Note that the attackers need to be located in all the paths followed by SUBMITQ, and hence the MITM is just a special case of the blocking attack.

V. SIMULATION REULTS

The hop counts of routing the successful queries, despite the unsuccessful ones, are shown in Fig. 4 for GES, SocioNet. As shown in Fig. 4, our proposal performs very well as most queries can be forwarded to their destination in no more than 10 hops (~ ln N), validating our analytical results. In contrast , some queries in GES and SocioNet may take more than 25 and 40 hops , respectively. Notably, GES performs better than SocioNet in terms of hop count of routing a query message. SocioNet performs poorly, as the resultant overlay quality may trap in a local optimum, where as GES depends on random walks to discover similar peers that may help bypass a local optimum. However, GES does not guarantee exploiting the most similar neighbours for any participating peer. Overall both GES and SocioNet do not perform well, that is, peers cannot efficiently exploit the similarity of participating peers in structuring the overlay geometry and effectively take advantage of peers similarity in routing the queries. We implement the search protocol based on blind flooding (i.e., Gnutella) over GES and SocioNet. Notably, in this study, the TTL values for a query message equal to 3 and 7 are investigated . With a TTL of 7, a query message reaches all peers in the system in this study.

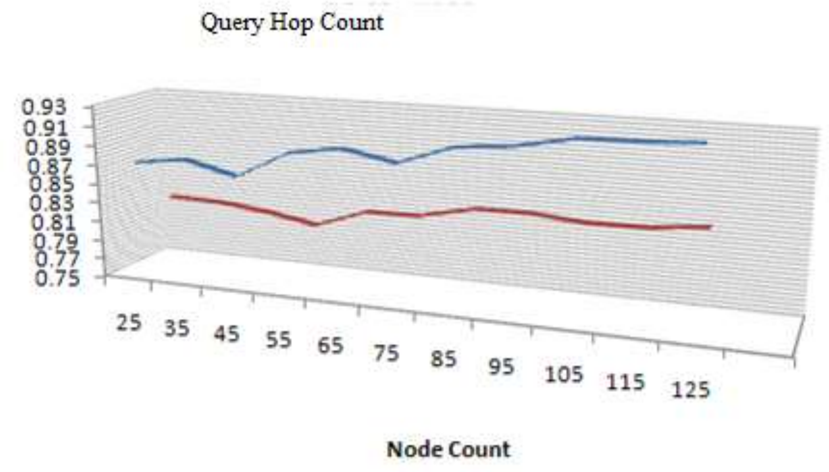


Fig. 4.1 Query hop count

VI. CONCLUSIONS

In this study, we have presented an unstructured P2P network with rigorous performance guarantees to enhance search efficiency and effectiveness. In constant probability, a querying peer takes O(lnc N) hops (where c is a small constant) to reach the destination node capable of resolving the query, where as the query message can progressively and effectively exploit the similarity of the peers. The query can be successfully resolved in an approximate probability of 100 percent. Notably, the theoretical analysis further reveals that the competitive decentralized solutions (e.g., those in [11], [12], [13], [14]) do not perform well as the hop count of routing a query message in such networks, considering the exploitation of the similarity of participating peers, is in the polynomial of system size N. The Search protocol we have suggested in this paper, which takes the advantage of similarity of peers exploited by overlay network can considerably reduce the search traffic. Peers participating in a Peer-to-Peer network are often heterogeneous in terms of their network bandwidth, storage space, and computational capability.

REFERENCES

[1]. S. Androutsellis-Theotokis and D. Spinellis, "A Survey of Peer-to-Peer Content Distribution Technologies," ACM Computing Surveys, vol. 36, no. 4, pp. 335-371, 2 004.

[2]. A. Rowstron and P.Druschel, "Pastry: Scalable, Distributed Object Location and Routing Routing for Large Scale Peer-to-Peer Systems," Proc IFIP/ACM Int'1 Conf.Distributed Systems Platforms (middleware), 2001.

[3]. S. Sen and J. Wang, "Analyzing Peer-to-Peer Traffic Across Large Networks," IEEE/ACM Trans. Networking, vol. 12, no. 2, pp. 219-232, Apr. 2004.

[4]. X. Li and J. Wu, "Searching Techniques in Peer-to-Peer Networks," Handbook of Theoretical and Algorithmic Aspects of Ad Hoc, Sensor, and Peer-to-Peer Networks, pp. 613-642, Auerbach, 2006.

[5]. Y. Zhu and Y. Hu, "Semantic Search in Peer-to-Peer Systems," Handbook of Theoretical and Algorithmic Aspects of Ad Hoc, Sensor, and Peer-to-Peer Networks, pp. 643-664, Auerbach, 2006.

[6]. C. Tang, Z. Xu, and S. Dwarkadas, "Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks," Proc. ACM SIGCOMM '03, pp. 175-186, Aug. 2003.

[7]. M. Li, W.-C. Lee, A. Sivasubramaniam, and J. Zhao, "SSW: A Small World-Based Overlay for Peer-to-Peer Search," IEEE Trans. Parallel and Distributed Systems, vol. 19, no. 6, pp. 735-749, June 2008.

[8]. I. Stoica, R. Morris, D. Liben-Nowell, D.R. Karger, M.F. Kaashoek, F.Dabek, and H.Balakrishnan,"Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications," IEEE/ACM Trans. Networking, vol. 11,no. 1,pp. 17-21,Feb.2003.

[9]. Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, "Making Gnutella-like P2P Systems Scalable," Proc. AC SIGCOMM '03,pp.407-418,Aug.2003.

[10]. S. Schmid and R. Wattenhofer, "Structuring Unstructured Peer-to- Peer Networks," Proc. Int'l Conf. High Performance Computing (HiPC'07), pp.432-442,Dec.2007.

[11]. Y. Zhu and Y.Hu, "Enhancing Search Performance on Gnutella- Like P2P Systems,"IEEE Trans. Parallel and Distributed Systems, vol. 17, no.12, pp. 1482-1495, Dec. 2006

[12]. M. Freedman, E. Sit, J. Cates, and R. Morris. Introducing Tarzan, a Peer-to-Peer Anonym zing Network Layer. In Proceedings of the International Workshop on Peer-To-Peer Systems (IPTPS), 2002.

[13]. D. Goldschlag, M. Reed, and P. Syverson. Onion Routing. Communications of the ACM (CACM), 1999.

[14]. J. Han, Y. Liu, L. Xiao, and L. Ni. A Mutual Anonymous Peer-to-peer Protocol Design. In Proceedings of the IEEE International Symposium on Parallel and Distributed Processing (IPDPS), 2005.

[15]. W. Hersh, C. Buckley, T. J. Leone, and D. Hickam. OHSUMED: An interactive retrieval evaluation and new large test collection for research. In Proceedings of the Annual International ACM SIGIR Conference, 1994.

[16]. I. King, C. H. Ng, and K. C. Sia. Distributed content-based visual information retrieval system on peer-to-peer networks. ACM Transactions on Information Systems, 2002.

[17]. A. Loser, M. Wolpers, W. Siberski, and W. Nejdl. Semantic Overlay Clusters within Super- Peer Networks. In Proceedings of the International Workshop on Databases, Information Systems and Peer-to-Peer Computing (DBISP2P), 2003.

[18]. A. R. M.Waldman and L. Cranor. Publius: A Robust, Tamper-Evident, Censorship-Resistant, Web Publishing System. In Proceedings of the USENIX Security Symposium, 2000.

[19]. P. Raftopoulou and E. Petrakis. iCluster: a Self-Organising Overlay Network for P2P Information Retrieval. In ECIR, 2008.

[20]. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content- Addressable Network. In Proceedings of the ACM Special Interest Group on Data Communications (SIGCOMM), 2001.

[21]. M. Reiter and A. Rubin. Crowds: Anonymity for web transactions. ACM Transactions on Information and System Security (TISSEC), 1998.

[22]. M. Rennhard and B. Plattner. Introducing MorphMix: Peer-to-Peer Based Anonymous Internet Usage with Collusion Detection. In Proceedings of the International Workshop on Privacy in the Electronic Society (WPES), 2002.

[23]. C. Schmitz. Self-Organization of a Small World by Topic. In Proceedings of the International Workshop on Peer-to-Peer Knowledge Management (P2PKM), 2004.

[24]. R. Sherwood, B. Bhattacharjee, and A. Srinivasan. P5: A Protocol for Scalable Anonymous Communication. In IEEE Security and Privacy, 2002.

[25]. C. Shields and B. Levine. A Protocol for Anonymous Communication over the Internet. In Proceedings of the ACM Conference on Computer and Communications Security (CCS), 2000.

[26]. A. Singh, T. Ngan, P. Druschel, and D. S. Wallach. Eclipse attacks on overlay networks: Threats and defenses. In Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), pages 1–12, 2006.