An Efficient, Secure and User Privacy-Preserving Search Protocol for Peer-to-Peer Networks

Jaydip Sen

Abstract. A peer-to-peer (P2P) network is a distributed system in which the autonomous peers can leave and join the network at their will and share their resources to perform some functions in a distributed manner. In an unstructured P2P network, there is no centralized administrative entity that controls the operations of the peers, and the resources (i.e., the files) that the peers share are not related to the their topological positions in the network. With the advent of the Internet of Things (IoT), the P2P networks have found increased interest in the research community since the search protocols for these networks can be gainfully utilized in the resource discovery process for the IoT applications. However, there are several challenges in designing an efficient search protocol for the unstructured P2P networks since these networks suffer from problems such as fake content distribution, free riding, whitewashing, poor search scalability, lack of a robust trust model and the absence a of user privacy protection mechanism. Moreover, the peers can join and leave the network frequently, which makes trust management and searching in these networks quite a challenging task. In this chapter, a secure and efficient searching protocol for unstructured P2P networks is proposed that utilizes topology adaptation by constructing an overlay of trusted peers and increases the search efficiency by intelligently exploiting the formation of semantic community structures among the trustworthy peers. It also guarantees that the privacy of the users and data in the network is protected. Extensive simulation results are presented and the performance of the protocol is also compared with those of some of the existing protocols to demonstrate its advantages.

Keywords: P2P network, topology adaptation, trust, reputation, semantic community, malicious peer, user privacy.

Jaydip Sen
Innovation Labs, Tata Consultancy Services Ltd.
Bengal Intelligent Park, Salt Lake Electronic Complex, Kolkata 700091, India
jaydip.sen@acm.org

N. Bessis et al. (Eds.): *Internet of Things & Inter-cooperative Comput. Technol.*, SCI 460, pp. 279–320.
DOI: 10.1007/978-3-642-34952-2_12

1 Introduction

During the past few years, in the area of wireless communications and networking, a novel paradigm named the IoT which was first introduced by Kevin Ashton in the year 1998, has gained increasingly more attention in the academia and industry [45]. By embedding short-range mobile transceivers into a wide array of additional gadgets and everyday items, enabling new forms of communication between people and things, and between things themselves, IoT would add a new dimension to the world of information and communication. Unquestionably, the main strength of the IoT vision is the high impact it will have on several aspects of every-day life and behavior of potential users. From the point of view of a private user, the most obvious effects of the IoT will be visible in both working and domestic fields. In this context, assisted living, smart homes and offices, e-health, enhanced learning are only a few examples of possible application scenarios in which the new paradigm will play a leading role in the near future [5][7]. Similarly, from the perspective of the business users, the most apparent consequences will be equally visible in fields such as automation and industrial manufacturing, logistics, business process management, intelligent transportation of people and goods.

The real power of the IoT lies in the universal connectivity among all devices and objects. However, it calls for interoperability so that the service requestors must know the features offered by the service providers, and it should be possible for the service requestors to understand what the service providers have to offer by semantic modeling. This is a key issue for stepping towards ubiquitous services, where the new or modified services may appear at any time, and towards device networks that are capable of dynamically adapting to the context changes as may be imposed by the application. This calls for a middleware which will interface between the devices and the applications. Since the devices need to communicate with each other, there is a need for a naming and addressing scheme, and a mechanism for search and discovery. Moreover, since each device is mapped to an identity (through naming and addressing), there are serious security and privacy concerns. All these challenges should be tackled by the middleware. One efficient approach for developing the middleware platform for IoT is using a multi-agent system. In a massively distributed system like the IoT, several agent platforms will exist each having a set of agents running and registered with a *directory facilitator* (DF). Problem, however, arises when the agents from different platforms will have to search the remote DFs and interact with the agents located on the remote platforms. In these scenarios, the agents will have to use resource discovery protocols which are similar to the file searching protocols in a purely unstructured P2P network. Hence, efficient searching in unstructured peer-to-peer network has a direct contextual relevance to the resource discovery in IoT applications. We provide below a brief descussion on the P2P networks before presenting the motivation and contribution of this chapter.

The term *P2P systems* encompasses a broad set of distributed applications which allow sharing of computer resources by direct exchange between systems. The goal of a P2P system is to aggregate resources available at the edge of Internet and to share it co-operatively among the users. The file sharing P2P systems have particularly become popular as a new paradigm for information exchange among large number of users in the Internet. These systems are more robust, scalable, fault-tolerant and they offer better availability of resources than the traditional systems based on the client-server model. Depending on the presence of a central server, the P2P systems can be classified as *centralized* or *decentralized* [44]. In the decentralized architecture, both the resource discovery and the resource download happen in a distributed manner. The decentralized P2P architectures may further be classified as *structured* or *unstructured* networks. In structured networks, there are certain restrictions on the placement of the contents and the network topologies. In unstructured P2P networks, however, the placement of the contents is unrelated to the topologies of the networks. The unstructured P2P networks perform better than their structured counterparts in dynamic environments. However, they need efficient search mechanisms and they also suffer from numerous problems such as: possibilities of fake content distribution, free riding (peers who do not share, but consume resources), whitewashing (peers who leave and rejoin the system in order to avoid penalties) and the lack of scalability in searching. The open and anonymous nature of the P2P applications leads to a complete lack of accountability of the contents that a peer may put in the network. The malicious peers often use these networks to carry out content poisoning and to distribute harmful programs such as Trojan Horses and viruses [47]. *Distributed reputation based trust management systems* have been proposed by the researchers to provide protection against the malicious content distribution in a distributed environment [1]. The main drawbacks of these schemes are their high overheads of message exchange and their susceptibility to misrepresentation by the malicious nodes. Guo et al. have proposed *trust-aware adaptive P2P topology* to control the free-riders and the malicious peers [29]. In [16] and [55], a topology adaptation approach is used to minimize the distribution of inauthentic files by the malicious peers in P2P networks. However, these schemes do not work well in the unstructured networks. The unstructured P2P networks also suffer from the poor search scalability problem. The traditional mechanisms such as controlled flooding, random walker and topology evolution all lack scalability. Zhuge et al. have proposed a trust-based a probabilistic search algorithm called *P-walk* to improve the search efficiency and to reduce unnecessary traffic in P2P networks [68]. In P-walk, the neighboring peers assign trust scores to each other. During the routing process, the peers preferentially forward the queries to the highly ranked neighbors. However, the performance of the algorithm in large-scale unstructured networks is questionable. To combat the free-riders, various trust-based incentive mechanisms are presented in [57]. Most of these mechanisms, however, involve large overhead of computations.

To combat the problem of inauthentic downloads as well as to improve search scalability while protecting the privacy of the users, this chapter proposes an *adaptive trust-aware protocol* that is robust and scalable. The proposed protocol increases the search efficiency by suitably exploiting the semantic community structures formed as a result of topology adaptation, since most of the queries are resolved within the semantic communities. Moreover, it effectively combines the functionalities of a robust trust management model and the semantic community formation thereby making the searching process secure and efficient while protecting the privacy of the users. The trust management module uses direct observations by a peer about its neighbors as well as the indirect observations reported by neighbors about the peers in its neighborhood. The direct observations are referred to as the *first-hand information*, while the observations reported by the neighbors of a peer are referred to as the *second-hand information*. The trust management module computes the trust metrics for different peers, and based on the values of the trust metrics, it segregates the honest peers from malicious peers using both first-hand and second-hand information. The semantic community formation allows topology adaptation to form cluster of peers sharing similar contents in the network. The formation of the semantic communities also enables the protocol to form a neighborhood of trust which is utilized to protect user privacy in the network. The work presented in this chapter is an extension of our already published work [49] [50]. The specific contributions made in present work are: (i) the usefulness of the proposed protocol in the context of the IoT has been identified, (ii) an extensive state-of-the-art survey of the existing searching mechanisms for P2P networks has been presented so that the specific contributions of the proposed protocol can be understood clearly, (iii) the impact of the phenomena of node churning and free riders on the proposed protocol are analyzed, and (iv) a detailed comparative analysis of the proposed protocol with two existing protocols is presented.

The rest of the chapter is organized as follows. Section 2 discusses some of the existing search protocols for structured and unstructured P2P networks. Section 3 presents the proposed protocol for secure and privacy-aware searching. Section 4 introduces various metrics to measure the performance of the proposed protocol. In Section 5, we present the performance results of the protocol based on the metrics defined in Section 4. A brief discussion is also made on the comparative analysis of the performance of the proposed protocol with some of the existing similar protocols in the literature. Section 6 concludes the chapter while highlighting some future scope of work.

2 Related Work

In this section, we briefly describe some of the searching protocols for P2P networks existing in the literature. We broadly divide these protocols into three categories: (i) general searching schemes, (ii) secure searching schemes,

and (iii) privacy-preserving searching schemes. The primary objective of the schemes under the general searching category is to enhance the search efficiency - i.e., to reduce the search time, to increase the scalability, and fault-tolerance etc. The secure searching schemes attempt to incorporate security into the searching mechanisms by defending against various possible attacks on the peers and the overall network. The privacy-preserving searching mechanisms protect peer (i.e., the user) privacy while carrying out the searching operation. In the following subsections, we briefly discuss some of the currently existing schemes under each of these three categories of search.

2.1 General Searching Schemes in P2P Networks

De Mello et al. have proposed a searching mechanism that is based on the discovery of trust paths among the peers in a P2P network [20]. Li & Wang proposed a global trust model based on the distance-weighted recommendations to quantify and evaluate the peers in a P2P network [36]. Adamic et al.[4] propose random-walk strategies in power-law networks(refer to Section 3.1), and find that by modifying the walkers to seek out for the peers having high degrees, the search performances in P2P networks can be greatly enhanced. However, such strategies lack scalability and do not perform well in a network having large number of peers.

Condie et al. presented a protocol named *adaptive peer-to-peer technologies* (APT) for the formation of adaptive topologies to reduce spurious file downloads and free riding. The peers connect to those peers from whom they are most likely to download the authentic files [16]. The peers add or remove their neighbors based on *local trust* and *connection trust* which are decided based on the transactions history. The scheme follows a defensive strategy for punishment since the peers follow the same strategy of punishment for both the malicious peers as well as the neighbors through whom they receive the responses from the malicious peers. This punishment strategy is relaxed in the *reciprocal capacity-based adaptive topology protocol* (RC-ATP), wherein a peer connects to others which have higher reciprocal capacities [55]. The *reciprocal capacity* of a peer is defined based on its capacity of providing good files and also on its ability of recommending the source of authentic files for download. While the RC-ATP scheme provides better network connectivity than the APT scheme and it also reduces the cost due to the inauthentic downloads, it has a large overhead due to the topology adaptation. Kamvar et al. proposed an algorithm that reduces the number of downloads of inauthentic files in a file-sharing P2P network [34]. Each peer is assigned a unique global trust value that is computed based on the historical activities of the peer in the network. A distributed and secure method based on *power iteration* is also presented for computing the global trust values of the peers. Based on the global trust values, the malicious peers are identified and isolated from the network. Xiao et al. have proposed an *adaptive connection*

establishment (ACE) protocol that constructs an overlay multicast tree by including each source peer and the neighboring peers within a certain diameter from the source peer [60]. It further optimizes the connecting edges in the overlay graph that are not included in the tree while retaining the scope of the search. The protocol is fully distributed since the peers do not need global knowledge of the whole overlay network while using the search protocol.

In [61], which is known as Gnutella v0.6 system, a two-layer hierarchical structure is deployed. The peers are categorized into two types: the *leaf-peer* and the *ultra-peer*. The leaf-peers have connections with their respective ultra-peers, while the ultra-peers have connections with their own leaf-peers as well as with the other ultra-peers. The leaf-peers can initiate lookup requests, receive lookup responses and respond to requests for which they have exact answers. An ultra-peer forwards the lookup requests to other the ultra-peers or the leaf-peers to which the ultra-peer is connected, if it exactly knows which leaf-peer has answers to the requests. At the ultra-peer level of the hierarchy, a flooding mechanism is used for forwarding the lookup requests.

Hsiao et al. have addressed the *topology mismatch problem* in unstructured P2P networks using a novel topology matching algorithm [30]. In the proposed algorithm, each peer creates and maintains a constant number of overlay connections with other peers in a distributed manner. Tang et al. have proposed an analytical scheme that studied the search performance in a P2P network under time-to-live (TTL)-based search [56]. In [67], a fully distributed protocol named *distributed cycle minimization protocol*(DCMP) has been presented that minimizes duplicate messages by eliminating any possible redundant cycles of the messages. Lin el al. proposed a dynamic search algorithm which combines the strategies of flooding and *random walk* [38].

Li et al. have proposed a consistency maintenance scheme for heterogeneous P2P systems with shorter convergence time and light-weight bandwidth consumption by taking into consideration the network locality information and the heterogeneity of peer capacities in the network [37]. Martinez-Yelmo et al. have proposed a two-level hierarchical P2P overlay architecture for inter-connection of different *P2P session initiation protocol* (P2PSIP) clusters [40].

Zhang & Hu have presented a protocol for P2P search with the assistance from a partial indexing service based on the interests of the peers and the data popularity [66]. Yang & Yang proposed a two-level hybrid P2P system to make use of the advantages of both structured and unstructured P2P networks [65]. The upper level of the system is a structured core network which forms the backbone of the hybrid system while the lower level consists of multiple unstructured P2P networks each of which is attached to a super-peer at the upper level. Huang-Fu et al. proposed a hybrid P2P system for mobile devices that utilizes the short message service as the control protocol to identify the address of the called peer [31]. In the proposed scheme, the *mobile station integrated services digital network* (MSISDN) number, i.e., the telephone number, is used as the globally unique identification for each

participating peer. Joung & Lin have proposed a fully decentralized algorithm to build a hybrid P2P system that does not need any human intervention and does not involve any centralized gateway to select the peers or to guide the peers to build a structured overlay [32].

Gkantsidis et al. [26] propose several hybrid search schemes for unstructured P2P networks. The authors have studied the performances of the search strategies in terms of several metrics such as: the number of distinct peers discovered, the number of messages propagated (i.e. communication overhead), and the maximum response time for the search queries. The authors have evaluated the performance of normalized flooding in non-regular P2P networks to show that normalization in flooding effectively tackles the problems caused by non-regularity in the network. It has also been shown that 1-step replication is helpful in search by random walk as well as search by normalized flooding, especially when the network has a small number of supernodes. The authors have utilized the theory of random graph to develop new algorithms based on *edge criticality heuristics* [35] used in the theory of approximate algorithms[58].

2.2 Secure Searching Schemes in P2P Networks

While efficiency of searching has been the major focus in most of the aforementioned schemes, security has also attracted attention of the researchers. Balfe et al. have discussed how the concepts of trusted computing can be applied to secure P2P networks [6]. The authors have argued that the central problem in securing P2P network lie in the fact that these networks do not have any stable verifiable peer identity verification mechanism. This leads to a major conflict between the requirements of anonymity of the users to protect their privacy and an increasing need to provide robust access control, data integrity, confidentiality and accountability services. The authors have shown how the *trusted computing group* (TCG) protocols for *direct anonymous attestation* (DAA) can be used to enforce the use of stable, platform-dependent pseudonyms so that spoofing attacks can be prevented. The proposed scheme also uses the DAA protocol to build entity authentication using pseudonyms for establishing secure communication channels between any given pair of peers.

Dingledine et al. [22] and Douceur [23] discuss various ways in which the spoofing attacks can be launched by malicious peers in a network that does not have a trusted central authority to verify the identities of the peers. The authors have proposed the use of reputation of the peers and the micro-cash schemes to detect such attacks. Sit & Morris [54] present a framework for performing a security analysis in P2P networks. The authors have proposed a taxonomy of attacks at the various layers of the communication protocol stack in the peers. At the network layer, attacks have been identified in the routing table lookup, maintenance, and route discovery process. The possible

attacks on the file storage systems in the peers, and various types of *denial of service* (DoS) attacks on the peers and on the overall network have also been identified.

A large number of studies have been carried out on the reputation and trust management in both the unstructured and the structured P2P networks. The reputation schemes such as EigenTrust [34] and PeerTrust [63] have been proposed to work on top of the structured P2P networks such as CAN [41] and P-Grid [3]. Aberer & Despotovic have proposed a scheme to identify the dishonest peers based on the complaints received from the honest peers in the network [2]. However, since the scheme uses the negative feedbacks only, no distinction can be made between an honest peer and a peer which has not been active for some time or a newly joined peer. The EigenTrust scheme proposed by Kamvar et al. [34] evaluates the trust information provided by the peers based on their trustworthiness. The scheme utilizes a novel normalization process in which the trust ratings of a peer are averaged and normalized. However, the normalization may lead to partial loss of important information on the original distribution and variance of trust function.

In a scheme proposed by Damiani et al. [19], the trustworthiness of file is determined based on a voting mechanism invoked among the participating peers in a P2P system. However, the scheme does not distinguish between the votes of the peers having high reputation values from those of the peer with low reputation. Hence, peers having low reputation values can manipulate the final result of the vote thereby making the scheme unreliable. Xiong & Liu [62] propose a scheme for evaluating trust in the peers in a P2P e-commerce environment. Cha & Kim propose a reputation management scheme based on the unbiased collective intelligence of the nodes in a P2P network for identifying and removing fake multimedia files [12].

2.3 Privacy-Preserving Searching Schemes in P2P Networks

Since the protection of the privacy of the users has become a critical requirement over the years, the researchers have attempted to address this issue in P2P protocol designs. One easy way to preserve the privacy of the users in network communication is to deploy some fixed servers or proxies for this purpose. For example, in the Publius system [59], the identity of a publisher is protected by encrypting the data communicated in the network, and managing the key distribution among k servers by using the mechanism of *threshold cryptography* [21][52]. Some anonymity schemes based on the use of a trusted third party server have been presented in [64]. A scheme called "APES" has been proposed to achieve mutual anonymity in a peer-to-peer file sharing system [46].

One popular approach for preserving peer privacy in a P2P system is to reveal the identity of the previous peer only over an entire multi-hop route from a source to a destination. FreeNet [15][14], Crowds [43], Onion routing [28][42], and the shortcut responding protocol [64] are few examples of this approach. Although very effective for hiding the identity of the peers, this approach has a serious problem in P2P systems. In a P2P system, the logical neighbor of a peer may be far away in terms the physical distance. Multi-hop communications over such long links usually lead to high rate of packet drops, delay, and jitter.

Lu et al. propose a trust-based privacy preservation scheme for P2P data sharing [39]. The proposition is based on selection of a trusted peer as the proxy during the data acquirement. The requester peer sends the request and receives the data through the proxy without revealing its identity. Since the real identity of the requester is never revealed during the communication, the privacy of the requester node is protected. However, in an structured P2P network, the selection and maintenance of the trusted peers for each peer is difficult due to the dynamic nature of the network topology and the autonomy of the peers. Hence, the scheme is difficult to deploy in real-world networks.

In [53], a *peer-to-peer personal privacy protocol* (p^5) has been proposed for protecting the privacy of a sender-receiver pair. In this protocol, the packets from a source are transmitted to all the members of a broadcast group to which the source and the receiver belong, instead of sending the packets to the receiver only. To ensure confidentiality of the message, each packet is encrypted by the sender using the public key of the receiver. In order to maintain the traffic level in the network at a constant level, the peers generate noise packets if they have no real packets to send. The use of noise packets makes it impossible for an eavesdropper to distinguish a data packet form a noise packet. The anonymity is achieved at the cost of the suboptimal utilization of the network bandwidth.

Goel et al. have proposed a peer-to-peer communication system - named *Herbivore* - that can ensure provable anonymity of the peers [27]. The idea behind its design is borrowed from the well-known *dining cryptographer networks* [13]. To make the anonymization protocol scalable, the network is logically partitioned into a large number of small anonymizing cliques. For anonymizing one bit of information, Herbivore has to propagate at least $2(k - 1)$ bits, where k is the size of the clique in which the message is being communicated. Moreover, if a node has to send a packet, for achieving anonymity, all the other peers in the same clique will have to send at least the same amount of data. This results in a high communication overhead in the protocol.

The Motivation of the Proposed Protocol: The protocol presented in this chapter draws its motivation from the APT [16] and RC-ATP [55] protocols. However, there are some significant differences between the protocol presented in this chapter and the APT and the RC-ATP protocol. First, in

the proposed protocol, the links in the original overlays are never deleted in order to avoid network partitioning. Second, in presence of malicious peers, the robustness of the proposed protocol is higher than that of the APT and the RC-ATP protocol. This claim is validated by the simulation results presented in Section 5. Third, as APT and RC-ATP both use flooding to locate resources, they have the typical problem of scalability in searching. The protocol presented in this chapter takes the advantage of semantic communities formation to improve the *quality of service* (QoS) of search by reducing the search time and increasing the rate of authentic file downloads. Fourth, APT and RC-ATP do not employ any robust trust model for ensuring security in searching and for protecting the user identity and data privacy. On the other hand, the central module of the proposed protocol is a robust trust management framework, which is responsible for securing the searching process and protecting the privacy of the peers and their data. Finally, unlike the APT and the RC-ATP protocols, the proposed protocol punishes the malicious peers by blocking all the queries which originate from these peers. This ensures that the malicious peers are not allowed to consume the resources and the services available in the network.

3 The Secure and Privacy-Aware Searching Protocol

This section is divided into three sub-sections. In Section 3.1, various parameters and the network environment of P2P network for which the proposed protocol is designed are discussed. In Section 3.2, the proposed search protocol is presented. Finally, Section 3.3 describes how the user privacy is protected in the proposed searching protocol.

3.1 The Network Environment

To obtain reliable results, the proposed protocol is evaluated on a realistic P2P network model. The factors that are taken into consideration in designing the protocol are discussed below.

(1) Network Topology: The topology of a P2P network plays an important role in the formation of trust among its peers and for efficient operation of a search protocol in the network. Following the work in [16] and [55], in the current proposition, the P2P network has been modeled as a *power law graph*. In a power law network, the degree distribution of the peers follows a *power law distribution*, in which the fraction of peers having degree L is L^{-k} where k is a network dependent constant. In the network environment, a certain percentage of the peers are randomly chosen to act as malicious peers. The malicious peers distribute bogus files in the network. As the protocol executes, the peers adjust topology locally to connect to those peers which have better chance to provide good files in future, and drop the malicious

peers from their neighborhood. The network links are categorized into two types: *connectivity link* and *community link*. The connectivity links are the edges of the original power law network which provide seamless connectivity among the peers. To prevent the network from being partitioned, these links are never deleted. On the other hand, the community links are added probabilistically between the peers who know each other, and have already interacted with each other before. A community link may be deleted when the perceived trustworthiness of a peer falls in the perception of its neighbors. The formal procedure of computing trust of a peer is discussed later in this section. However, informally, it may be said that the value of the trust metric of a peer i as computed by another peer j increases when the peer j has some positive experience while interacting with the peer i (i.e. getting an authentic file from peer the i). A negative experience (i.e. getting a bogus file) leads to decrease in the trust value. A limit is put on the additional number of edges that a peer can acquire to control the bandwidth usage and the query processing overhead in the network. This increase in network load is measured relative to the initial network degree (corresponding to the connectivity edges). Let *final_degree(x)* and *initial_degree(x)* be the initial and the final degree of a node x. The *relative increase in connectivity* (RIC) as computed in (1) is constrained by a parameter called *edge_limit*.

$$RIC(x) = \frac{final\_degree(x)}{initial\_degree(x)} \leq edge\_limit \tag{1}$$

(2) Content Distribution: The dynamics of a P2P network are highly dependent on the volume and the variety of the files that each peer chooses to share. Hence a model reflecting the real-world P2P networks is required. It has been observed that the peers are, in general, interested in a subset of the contents in the P2P network [17]. Also, the peers are often interested only in the files from a few content categories. Some categories of files are more popular than the others. It has been shown that the Gnutella content distribution follows the *zipf distribution* [48]. In the proposed scheme, the files are assigned to the peers at the network initialization phase as follows: The peer i is assigned some content categories C^i and the peer i is given an interest level for each content category $c \in C$. Finally, the peer i is assigned files F according to its content categories and interest levels in those categories. Each distinct file $f_{(c,r)}$ is uniquely identified by the content category c to which it belongs and its popularity ranking r within that category [48].

Accordingly, in the proposed protocol, the content categories and the file popularity within each category are both modeled as *zipf distribution* with $\alpha = 0.8$.

Content distribution model: In the proposed scheme, we assume that there are 32 content categories. It must be noted that the number of content categories can be any positive integer n. However, for the evaluation of the performance of the proposed protocol, we have used 32 content categories.

Table 1 An illustrative content distribution among peers

Peers	Content Categories
P_1	C_1, C_2, C_3
P_2	C_3, C_4, C_6, C_7
P_3	C_2, C_4, C_7, C_8
P_4	C_1, C_2
P_5	C_1, C_5, C_6

Let the content categories be $C = \{c_1, c_2,..., c_{32}\}$. Each content category is characterized by its popularity rank. For example, if $c_1 = 1$, $c_2 = 2$ and $c_3 = 3$, then c_1 is more popular than c_2 and hence it is more replicated than c_2 and so on. Since, the files in the more popular categories are searched and queried more frequently, more number of these files are stored and replicated than the files belonging to the less popular categories. This strategy ensures that more number of files belonging to the popular categories are available, which, in turn, makes the searching process faster and efficient. More details on the way in which the content categories are assigned to the peers and the interest levels of the peers are modeled can be found in [48].

As already discussed earlier, the peers are assumed to be interested in a subset of the total available contents in the network. Accordingly, each peer initially chooses a number of content categories and shares files only in those categories. In the proposed protocol, each peer randomly chooses between three to six content categories. The files belonging to more popular categories are shared more in numbers. Table 1 shows an illustrative content distribution among 5 peers in a network. The category C_1 is more replicated as it is the most popular category. Peer 1 (P_1) shares files of three categories: C_1, C_2, C_3. As explained earlier, P_1 shares maximum number of files in category C_1, followed by category C_2 and so on. On the other hand, Peer 3 (P_3) shares maximum number of files in category C_2 as it is the most popular among the categories of files chosen by it.

(3) Query Initiation Model: The authors in [48] have shown that the peers usually query for the files which are available in the network, and which belong to the content categories of their interests. However, the number of queries a peer issues may vary from peer to peer. Using the *Poisson* distribution this is modeled as follows: If M is the total number of queries issued in a cycle, and N is the number of peers present in the network, query rate $\lambda = M / N$ is the mean of the Poisson process. The expression: $p(\#\ of queries = K) = \frac{e^{-K}\lambda^K}{K!}$ gives the probability that a peer issues K queries in a cycle. The probability that a peer issues a query for the file $f_{c,r}$ depends on the peer's interest level in category c and rank r of the file within that category.

When a peer generates a query, instead of generating a search string, it generates the category and the rank (i.e., popularity) of the file that will

satisfy the query. On receiving the query, each peer checks whether it supports the file category and if so, whether it shares the file.

(4) Trust Management Engine: A trust management engine is designed which helps a peer to compute the trust ratings of the other peers based on the past transactions, as well as on the recommendations of its neighbor. For computing the trust values of the peers, a method similar to the one proposed in [24] is followed. The framework employs a *beta distribution* for reputation representation, updates and integration. The first-hand information and the second-hand information (recommendation from neighbors) are combined to compute the reputation value of a peer. The weight assigned by a peer i to a second-hand information received from a peer k is a function of the reputation of the peer k as maintained in the peer i. For each peer j, a reputation R_{ij} is computed by a neighbor peer i. The reputation is embodied in the *Beta model* which has two parameters: α_{ij} and β_{ij}. α_{ij} represents the number of successful transactions (i.e., the number of authentic file downloads) that the peer i had with the peer j, and β_{ij} represents the number of unsuccessful transactions (i.e., the number of unauthentic file downloads). The reputation of the peer j as maintained by the peer i is computed using (2).

$$R_{ij} = Beta(\alpha_{ij} + 1, \beta_{ij} + 1) \tag{2}$$

The trust metric of a peer is the expected value of its reputation and is given by (3).

$$T_{ij} = E(R_{ij}) = E(Beta(\alpha_{ij} + 1, \beta_{ij} + 1)) = \frac{\alpha_{ij} + 1}{\alpha_{ij} + \beta_{ij} + 2} \tag{3}$$

The second-hand information is presented to the peer i by its neighbor peer k. The peer i receives the reputation R_{kj} of the peer j from the peer k in the form of the two parameters α_{kj} and β_{kj}. After receiving this new information, the peer i combines it with its current assessment R_{ij} to obtain a new reputation R_{ij}^{new} as shown in (4).

$$R_{ij}^{new} = Beta(\alpha_{ij}^{new}, \beta_{ij}^{new}) \tag{4}$$

In (4), the values of α_{ij}^{new} and β_{ij}^{new} are given by (5) and (6) as follows.

$$\alpha_{ij}^{new} = \alpha_{ij} + \frac{2\alpha_{ik}\alpha_{kj}}{(\beta_{ik} + 2)(\alpha_{kj} + \beta_{kj} + 2)(2\alpha_{ik})} \tag{5}$$

$$\beta_{ij}^{new} = \beta_{ij} + \frac{2\alpha_{ik}\beta_{kj}}{(\beta_{ik} + 2)(\alpha_{kj} + \beta_{kj} + 2)(2\alpha_{ik})} \tag{6}$$

To prevent against *bad-mouthing* and *ballot-stuffing attacks* [10][18], the peers assign higher weights to the first-hand observations (i.e. direct observations) made by them, and less weights are given to the evidences provided by the other peers (i.e., the second-hand information). As mentioned earlier in this

section, the second-hand observations received from different peers are also weighted in proportions to the values of their respective reputation metrics. To incorporate these issues while updating the reputation values using the second-hand information, the *Dempster-Shafer theory* [51] and the *belief discounting model* [33] are employed. The use of these models leads to the derivation of the expressions in (5) and (6).

To make the trust management system robust against *sleeper attack* [25], where a peer behaves honestly for a sufficiently long time to acquire a good reputation and then starts misbehaving and exploiting the system, the proposed system assigns more weights to the recent observations for computing the aggregate reputation metrics of a peer. In this approach, the reputation metrics of a peer are periodically decreased by a weight w, using (7) and (8).

$$\alpha_{ij}^{new} = w * \alpha_{ij} \tag{7}$$

$$\beta_{ij}^{new} = w * \beta_{ij} \tag{8}$$

The choice of the weight w in (7) and (8) and the interval at which the reputation updates are made are two *tuneable parameters*. In [11], a technique has been proposed for computing the weight(w) by comparing the reputation evolution in the system with and without the weighting parameters.

As mentioned earlier in this section, the trust value of a peer is computed as the statistical expected value of its reputation. The trust value of a peer lies in the interval [0, 1]. The peer i considers the peer j as trustworthy if $S_{ij} \geq 0.5$, and malicious if $S_{ij} < 0.5$. In the implementation of the proposed protocol, we have used an LRU (least recently used) data structure which is maintained in each peer to keep track of the most recent transactions the peer had with maximum of 32 peers. However, the choice of the number of peers whose transaction history is maintained in each peer is a tuneable parameter, which can be increased or decreased based on the memory and the computing capabilities of the peers.

(5) Identity of the Peers: Each peer generates a 1024 bit public/private RSA key pair. The public key serves as the identity of the peer. The identities are persistent and they enable two peers that have exchanged keys to locate and connect to one another whenever the peers are online. In addition, a *distributed hash table* (DHT) is maintained that lists the transient IP addresses and the port numbers for all the peers and for all the applications running on the peers. The DHT entries for the peer i are signed by the peer i and encrypted using its public key. Each entry is indexed by a 20 byte randomly generated shared secret, which is agreed upon during the first successful connection between the two peers. Each peer's location in the DHT is independent of its identity and is determined by hashing the client's current IP address and the DHT port. This prevents any possible systematic monitoring of the targeted regions of the DHT key space, since the region for which each peer is responsible is determined by the peer's network address and the port.

(6) Node Churning Model: In P2P networks, a large number of peers may join and leave at any time. This activity is termed as *node churning*. To simulate node churning, prior to each *generation* (a set of consecutive searches), a fixed percentage of nodes are chosen randomly as *inactive peers*. These peers neither initiate nor respond to a query in that generation, and they join the system later with their LRU data structure cleared. The clearing of the LRU data structure ensures that these peers do not have any historical information about their past transactions with other peers in the network. Since in a real world network, even in presence of churning, the approximate distribution of content categories and files remain constant, the contents of the peers undergoing churn are exchanged with the peer remaining in the network, so that the content distribution model of the network remains unchanged.
(7) Threat Model: The malicious peers adopt various strategies (threat models) to conceal their behavior so that they can effectively disrupt the activities in the network, and yet go undetected. The proposed protocol considers two threat models. The peers which share good quality files enjoy better topological positions after topology adaptation. In the threat model A, the malicious peers attempt to circumvent this effect by providing good files (occasionally) with a probability - known as *degree of deception*- to lure other peers to form communities with them. In the threat model B, a group of malicious peer joins the system and provides good files until the connectivity of the peers reaches a maximum value - the *edge limit*. The peers then start acting maliciously by spreading fake contents in the network. In Section 5, we will see how effective these strategies are in disrupting the network operations.

3.2 The Proposed Search Protocol

The network learns the trust information through the search process, and updates the trust information and adapts then topology based on the outcome of the search. An ideal search protocol should satisfy several requirements such as: (a) It should have a high search efficiency and search quality - i.e. it must have the ability to download authentic files in a short period of time, (b) it should have a minimal overhead in terms of computation, storage and message passing, (c) It must provide incentives to the peers which share a large number of authentic files, (d) it should be self-policing in the sense that a peer should be able adjust its search strategy based on the local estimate of the network connectivity, and (e) it should be able to protect the privacy of its users. The proposed search protocol has been designed to satisfy each of these requirements.

The proposed protocol works in three steps: (i) search, (ii) trust computing and verification, and (iii) topology adaptation. Each of these steps is discussed in the following.

Search: A *time to live* (TTL) bound search is used. At each peer, the query is forwarded to a subset of its neighbors; the number of neighbors is decided based on the local estimate of connectivity. The *connectivity index* for the peer x is denoted as $Prob_{com}(x)$ and is given by (9).

$$Prob_{comm}(x) = \frac{current\_degree(x) - initial\_degree(x)}{initial\_degree(x)(edge\_limit - 1)} \tag{9}$$

When $Prob_{com}$ for a node is low, the peer has the capacity to accept new community edges for expanding the community structure. Higher the value of $Prob_{com}$, it is less likely that the neighbors will disseminate the queries. As the protocol executes, the connectivity of the good peers increases and finally reaches a maximum value. At this time, the peers focus on directing the queries to appropriate communities which may host the specific file rather than expanding the communities. For example, if peer i can contact at most 10 neighbors and $Prob_{com}$ of i is 0.6, it forwards the query to: 10 x (1 - 0.6) = 4 neighbors only. The search strategy is changed from the initial TTL-limited *breadth first search* (BFS) to a directed *depth first search* (DFS) with the restructuring of the network. The search process operates in two steps: *query initiation* and *query forward*. These steps are described in the following.

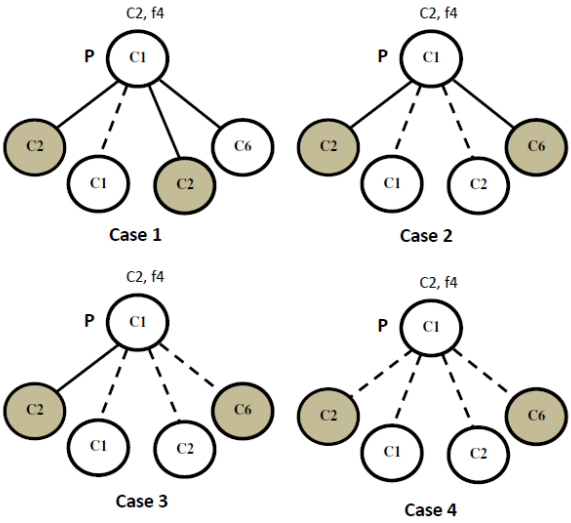

Fig. 1 Neighbor selection by peer P for forwarding the query string (c_2, f_4). The community edges and the connectivity edges are drawn using solid and dotted lines respectively. The peers that receive the query for forwarding are shaded.

(i) Query Initiation: The initiating peer forms a query packet containing the name of the file (c, r) and forwards it to some of its neighbors along with the $Prob_{com}$ and the TTL values. The query is disseminated using the following *neighbor selection rule*. The neighbors are ranked based on both their trustworthiness and their similarities of interest. Preference is given to the trusted neighbors sharing similar contents. Among the trusted neighbors, the community members having their contents matched to the query are preferred. If the number of community links is not adequate enough, the query is forwarded through the connectivity links also. The various cases of neighbor selection are illustrated in Fig. 1. It is assumed that in each case only two neighbors are selected for forwarding a query. When the query (c_2, f_4) reaches the peer P, following four cases may occur. In Case 1, the peer P has sufficient number of community neighbors (two community neighbors) sharing files in the category c_2. Hence, these peers are chosen for forwarding the query. In Case 2, the number of community neighbors sharing the requested category of file is not sufficient enough - only one community neighbor has the file in the category c_2. In this scenario, the community neighbors sharing the c_2 and the c_6 categories of files are preferred over the connectivity neighbor sharing the file category c_2 for forwarding the query. This is because of the fact that the peers forward queries to the community peers which have higher trust values than the connectivity peers. In Case 3, there is only one community neighbor that shares the file category c_2. Hence that neighbor is chosen for the purpose of query forwarding. Among the remaining connectivity neighbors, the most trusted one containing the c_6 category is selected. In Case 4, there are no community neighbors. Assuming that the peer P has the same level of trust for all its neighbors, the neighbor sharing the matching content category c_2 is chosen for forwarding the query. Among the rest of the neighbors, the peer c_6 is chosen randomly (since only two forwarding peers are to be selected).

When a query reaches peer i from peer j, peer i forwards the query further in the network as discussed below.

(ii) Query Forwarding: (i) *Check the trust level of the peer j*: The peer i checks the trust rating of the peer j through the *check trust rating* algorithm (explained later in this section). The selection of the peers for further forwarding of the query is done accordingly. (ii) *Check the availability of the file*: If the requested file is found, a response is sent to the peer j. If the TTL value has not expired, the following steps are executed. (iii) *Calculate the number of messages to be sent*: The number of messages to be sent is calculated based on the value of $Prob_{com}$.(iv) *Choose the neighbors*: The neighbors are chosen using the neighbor selection rule. The search process is shown in Fig. 2. It is assumed that from each peer, the query is forwarded to two neighbors. The matching community links are preferred over the connectivity links to dispatch the query. The peer 1 initiates the query and forwards it to two community neighbors 3 and 4. The query reaches the peer 8 via the peer 4. However, the peer 8 knows from its previous transactions with the peer 4 that the peer 4 is malicious. Hence, it blocks the query. The query forwarded

by the peer 5 is also blocked by the peer 10 and the peer 11 as both of them know that the peer 5 is malicious. The query is matched at four peers: 4, 6, 9 and 12. The search process is shown in Fig. 2.

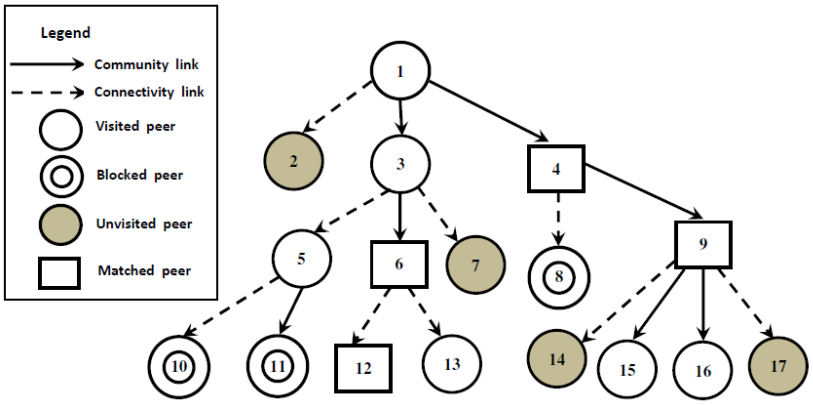

Fig. 2 The breadth first search (BFS) tree for the search initiated by peer 1

Topology Adaptation: The responses are sorted by the initiating peer i based on the reputations of the resource providers, and the peer having the highest reputation is selected as the source for downloading. The requesting peer checks the authenticity of the downloaded file. If the file is found to be fake, the peer i attempts to download the file from other sources until it is able to find the authentic resource or it does have any sources left for searching. The peer then updates the trust ratings and possibly adapts the network topology after a failed or a successful download, to bring the trusted peers closer to its neighborhood, and to drop the malicious peers from its community. The restructuring of the network is controlled by a parameter known as *degree of rewiring* which represents the probability with which a link is formed between a pair of peers. This parameter allows the trust information to propagate through the network. The topology adaptation consists of the following operations: (i) *link deletion*: The peer i deletes the existing community link with the peer j if it detects the peer j as malicious. (ii) *link addition*: The peer i probabilistically forms a community link with the peer j if the resource provided by the peer j is found to be authentic. If $RIC \leq edge_{limit}$, for both the peers i and j, only then an edge can be added, subject to the approval of the resource provider peer j. If the peer j finds that the peer i is malicious (i.e., its trust value is below the threshold), it doesn't approve the link.

Fig. 3 illustrates a topology adaptation on the network topology shown in Fig. 2. In the example shown in Fig. 3, the peer 1 downloads the file from the peer 4 and finds that the file is spurious. It reduces the trust score of the peer 4 and deletes the community link 1-4. It then downloads the file from

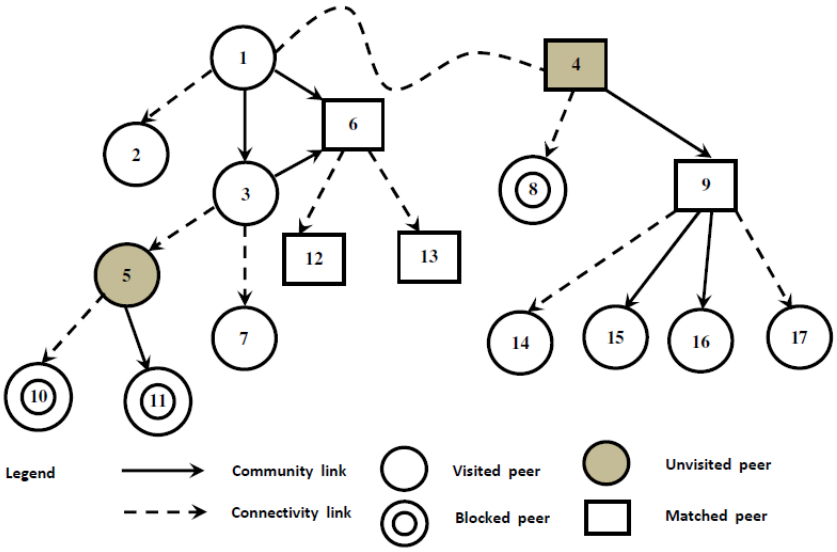

Fig. 3 Topology adaptation based on outcome of the search in Fig. 2. Malicious nodes are shaded in gray color.

the peer 6 and gets an authentic file. The peer 1 now sends a request to the peer 6, and the latter grants the request after checking its trust value. Hence, the community edge 1-6 is added. The malicious peer 4 loses one community link and the peer 6 gains one community edge. However, the network still remains connected by the connectivity edges which are shown in dotted lines in Fig. 3.

It may be noted that the addition of a community link is a more expensive operation than the deletion of a community link. However, if the number of malicious peers in a network is not too high, the link addition operation will be less frequent after the formation of semantic communities and stabilization of the topology adaptation. Hence, except during the initial semantic community formation phase, the overhead of the protocol operation will never be high. This will be discussed in more detail in Section 5.

Checking of the Trust Rating of the Peers: The trust rating of the peers is used at various stages of execution of the protocol to make a decision on the possible source for downloading a file, to stop a query forwarded from a malicious node and to adapt the topology. A *least recently used* (LRU) data structure is used at each peer to keep track of the 32 most recent peers it has interacted with. When no transaction history is available, a peer seeks for the recommendations from its neighbors using a *trust query* message. When the peer i doesn't have the trust score of the peer j in its LRU history, it first seeks for the recommendation about the peer j from all of its community neighbors.

If none of its community neighbors possesses any information about the peer j, then the peer i initiates a *directed DFS search*. The trust computation model has been presented in Section 3.1.

3.3 Privacy-Preservation in Searching

The trust-based searching protocol described above does not guarantee any privacy requirement of the requester (i.e. the initiator of the query). For protecting the privacy of the user, several enhancement of the protocol are proposed. Following cases are identified for privacy preservation.

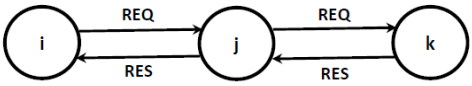

Fig. 4 Identity protection of the requesting peer i from the supplier peer k by use of trusted peer j. *REQ* and *RES* are the request and response message respectively.

(a) Protection of the Identity of the Requesting Peer: In this case, as shown in Fig. 4, instead of sending the request straightway to the supplier peer, the requesting peer asks one of its trusted peers (which may or may not be its neighbor) to look up the data on its behalf. Once the query propagation module successfully identifies the possible supplier of the resource, the trusted peer serves as a proxy to deliver the data to the requester peer. Other peers including the supplier of the resource will not be able to know the real requester. Hence, the requester's privacy is protected. Since the requestor's identity is only known to its trusted peer, the strength of privacy is dependent on the effort required to compromise the trusted peer. As mentioned in Section 3.1, the message communicated by the peers are encrypted by 1024 bit RSA key, which is a provably secure algorithm. Hence, the privacy of the requester peer is protected.

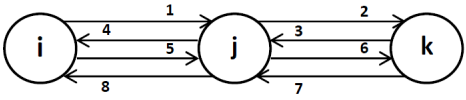

Fig. 5 Protecting data handle using trusted node. Peer i and k are the requester and the supplier peer respectively. Peer j is the trusted peer of the requester peer i.

(b) Protecting the Data Handle: To improve the achieved privacy level, the data handle may not be put in the request at the beginning. When a requester initiates the request, it computes the hash value of the handle and

reveals only a part of the hash result in the request sent to its trusted peer. The steps 1 and 2 in Fig. 5 represent these activities. Each peer receiving the request compares the revealed partial hash to the hash codes of the data handles that it holds. Depending on the length of the revealed part, the receiving peer may find multiple matches. This does not, however, imply that the peer has the requested data. Thus this peer will provide a candidate set, along with a certificate of its public key, to the requester. If the matched set is not empty, the peer constructs a *Bloom filter* [9] based on the left parts of the matched hash codes, and sends it back to the trusted peer. The trusted peer forwards it back to the requester. These are represented by the steps 3 and 4 in Fig. 5. On examining the filters, the requester can eliminate all peers that do not have the required data from the candidate data supplier list. It then encrypts the complete request with the supplier's public key and gets the requested data with the help from its trusted peer. The steps 5, 6, 7 and 8 in Fig. 5 represent these activities. By adjusting the length of the revealed hash code, the requestor can control the number of eliminated peers. The level of privacy is improved manifold since the malicious peers now need to compromise the trusted peer and also break the Bloom filter and the hash function in order to attack the privacy protection scheme.

(c) Hiding the Data Content: Although the privacy-preservation level has been improved during the lookup phase using the previous two schemes, the privacy of the requester will still be compromised if the trusted peer can see the data content when it relays the packets for the requester. To improve the privacy level and prevent eavesdropping, we can encrypt the data handle and the data content. If the identity of the supplier is known to the requester, it can encrypt the request using the supplier's public key. The public key of the requester cannot be used because the certificate will reveal its identity. The problem is solved in the following manner. The requester generates a symmetric key and encrypts it using the supplier's public key. Only the supplier can recover the key and use it to encrypt the data. To prevent the trusted peer of the requester from conducting a man-in-the-middle attack, the trusted peer is required to sign the packet. This provides a non-repudiation evidence, and shows that the packet is not generated by the trusted peer itself. The privacy level has been improved, since now in order launch an attack on the privacy of the requester, a malicious peer needs to break the encryption keys as well.

4 Performance Metrics for the Proposed Protocol

To analyze the performance of the proposed protocol, several metrics are defined. In this section, we provide a detailed discussion on these metrics which are used to evaluate the protocol performance. The performance results of the protocol based on these metrics are presented in Section 5.

(a) Attempt Ratio (AR): A peer keeps on downloading files from various sources based on their trust ratings till it gets the authentic file. AR is the

probability that the authentic file is downloaded in the first attempt. A high value of AR for the honest peers is desirable for a searching scheme to be efficient and scalable.

(b) Effective Attempt Ratio (EAR): It measures the cost of downloading an authentic file by a good peer in comparison to the cost incurred by a malicious peer. If $P(i)$ be the total number of attempts made by the peer i to download an authentic file, EAR is given by (10).

$$EAR = (\frac{1}{M}\sum_{i=1}^{M}\frac{1}{P(i)} - \frac{1}{N}\sum_{j=1}^{N}\frac{1}{P(j)}) \tag{10}$$

In (10), M and N are the number of malicious and good peers issuing queries in a particular generation. For example, EAR = 50 implies that if a good peer needs one attempt to download an authentic file, a malicious peer will need two attempts.

(c) Query Miss Ratio (QMR): Since the formation of semantic communities takes some time, there will be a high rate of query misses in the first few generations of search. However, as the protocol executes, the rate of query miss is expected to fall for the good peers. QMR is defined as the ratio of the number of search failures to the total number of searches in a generation.

(d) Hit per Message (HM): Due to the formation of the semantic communities in the network, the number of messages required to get a hit is expected to fall down as the network topology stabilizes. HM measures the search efficiency achieved by the proposed search protocol and it is defined as the number of query hits per message irrespective of the authenticity of the file being downloaded.

(e) Relative Increase in Connectivity (RIC): After a successful download, a requesting peer attempts to establish a community edge with the resource provider, if it is approved by the latter. This ensures that the peers which provide good community services are rewarded by providing them with an increased number of community neighbors. The metric RIC measures the number of community neighbors a peer gains with respect to its connectivity neighbors in the initial network topology. If $D_{init}(i)$ and $D_{final}(i)$ are the initial and the final degrees of the peer i, and N is the number of peers, then RIC for the peer i is computed using (11).

$$RIC(i) = \frac{1}{N}\sum_{i}\frac{D_{final}(i)}{D_{init}(i)} \tag{11}$$

(f) Closeness Centrality (CCen): Since the topology adaptation effectively brings the good peers closer to each other, the length of the shortest path between a pair of good peers decreases. This intrinsic incentive for sharing authentic files is measured by the metric CCen. The peers with higher

CCen values are topologically better positioned. If P_{ij} is the length of the shortest path between the peer i and the peer j through the community edges and if V denotes the set of peers, then CCen for the peer i is given by (12).

$$CCen(i) = \frac{1}{\sum_{j \varepsilon V} P_{ij}} \tag{12}$$

(g) Clustering Coefficient (CC): It gives an indication about how well the network forms cliques. CC plays an important role in the choice of the TTL value in the search protocol. With higher values of CC, lower TTL values can be used in the search operation. If K_i be the number of community neighbors of the peer i, then the CC of the peer i is computed using (13).

$$CC(i) = \frac{2E_i}{K_i(K_i - 1)} \tag{13}$$

In (13), E_i is the actual number of community edges between the K_i neighbors. CC of the network is taken as the average value of all CC(i)s.

(h) Largest Connected Component (LCC): The community edges connect the peers which have similar content interests and have sufficiently high mutual trust among each other. If we focus on the peers which share a particular category of contents, then we can observe that the community edges form a trust-aware overlay. However, it will be highly probable that the trust-aware overlay graph will be a disconnected graph. LCC is the largest connected component of this disconnected overlay graph. In other words, LCC of the network can be taken as a measure of the goodness of the community structure, since it signifies how strongly the peers with similar contents and interests are connected with each other.

(i) Trust Query Propagation Overhead (TQPO): The peers build trust and reputation information by collecting and using both the first-hand and the second-hand information. A trust query message is propagated when the trust information about a peer is not available locally in a peer. A trust query message involves one DFS round without any backtracking. The overhead incurred due to the trust query propagation is measured by the metric called *trust query propagation overhead* (TQPO). TQPO is defined as the total number of distinct DFS search attempts per generation. It may be noted that a trust query may be initiated multiple number of times for a single file search operation - to select a trusted neighbor or to approve a community link.

(j) Topology Adaptation Overhead (TAO): It gives an idea about the overhead due to the topology adaptation and it is measured by the number of community edges that are added or deleted in one cycle of operation of the search protocol. The larger the number of addition and deletion of the community edges, higher will be the associated overhead.

5 Performance Evaluation of the Proposed Protocol

A discrete time simulator written in C is used for simulating the protocol. In the simulation, 6000 peers, 18000 *connectivity edges*, 32 *content categories* are chosen. The values of the *degree of deception* and the *degree of rewiring* are taken as 0.1 and 0.3 respectively. The *edge_limit* value used is 2.5. The TTL values for the BFS and the DFS are taken as 5 and 10 respectively. Since one of the objectives of the simulation is to show higher scalability of the proposed protocol, the number of peers and the number of connectivity edges in the simulated network are chosen to much higher than those used in simulating the APT [16] and the RC-ATP [55] protocols, while the TTL value for BFS is kept constant. The values of the simulation parameters are presented in Table 2.

Table 2 Simulation parameters

Parameters	**Values**
No. of peers	6000
No. of connectivity edges	18000
No. of content categories	32
Degree of deception	0.1
Degree of rewiring	0.3
Edge limit	2.5
TTL for BFS	$5s$
TTL for DFS	$10s$
No. of search per generation	5000
No. of generations per cycle	100

The discrete time simulator simulates the protocol repeatedly on the power law network and outputs all the metrics averaged over the generations. *Barabasi-Albert* generator [8] is used to generate initial power law graph with 6000 nodes and approximately 18000 edges. The number of search per generation is taken as 5000 while the number of generations per cycle of simulation is 100.

To check the robustness of the protocol against attacks from malicious peers, the percentage of malicious peers is gradually increased. Fig. 6 illustrates the cost incurred by each type of peers to download the authentic files. It can be observed from Fig. 6(a) and Fig. 6(b) that with the increase in the percentage of the malicious peers in the network from 10% to 20%, the AR for the malicious nodes increases while the AR for the honest peers falls marginally. Since AR indicates the cost (in terms of number of attempts required for downloading an authentic file) incurred by a peer, it can be concluded that as the percentage of the malicious peers is increased, the cost

incurred by the malicious peers to download the authentic files decreases while that of the good peers increases.

It is also observed from Fig. 7 that the EAR values for the peers decrease as the percentage of the malicious peers in the network is gradually increased from 10% to 60%.

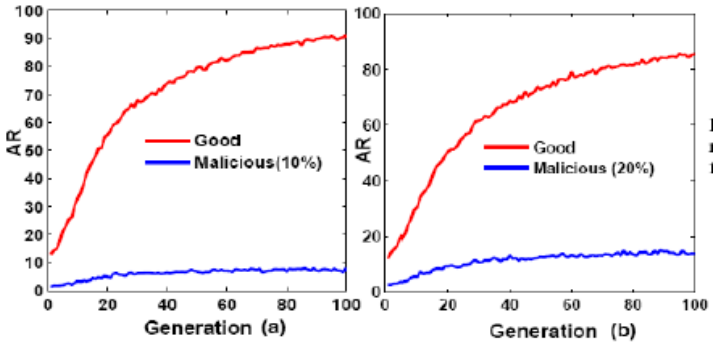

Fig. 6 AR for various percentages of malicious peers in the network. In (a) 10%, in (b) 20% nodes are malicious.

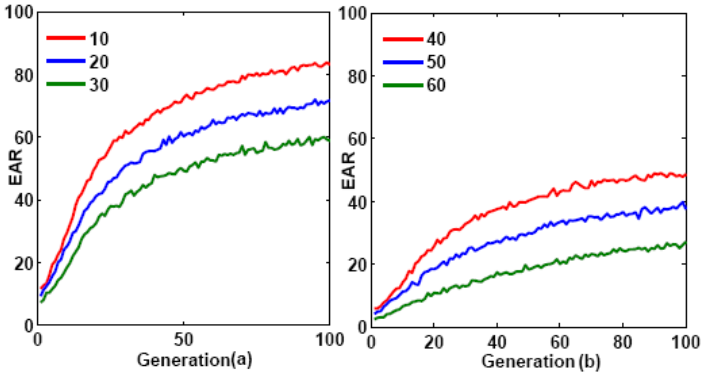

Fig. 7 EAR of honest peers for various percentages of malicious peers in the network. In (a) 10% - 30%, in (b) 40% - 60% peers in the network are malicious.

It is evident from Fig. 7 that when 10% of the peers in the network are malicious, the average EAR is 80; i.e., on the average, if a good peer needs one attempt to download an authentic file, a malicious peer needs 5 attempts. The peers which share high quality files acquire good reputation and earn more community edges and eventually disseminate the query through the

community edges only. As the queries are forwarded via the trusted peers at each hop, the probability of getting the authentic files in the first attempt increases. However, as the queries forwarded by the malicious peers are blocked by the good peers, they need more attempts to download the good files. It may be observed from Fig. 7(b) that when the percentage of the malicious peers in the network is 60%, the value of EAR drops to 30. Hence, as long as the percentage of the malicious peers in the network does not exceed 60%, the good peers have higher probability to get the authentic files in their first attempts as compared to the malicious peers. The results, therefore, indicate that the proposed protocol can withstand attacks by the malicious peers till such peers are less than 60% of the total number of peers in the network.

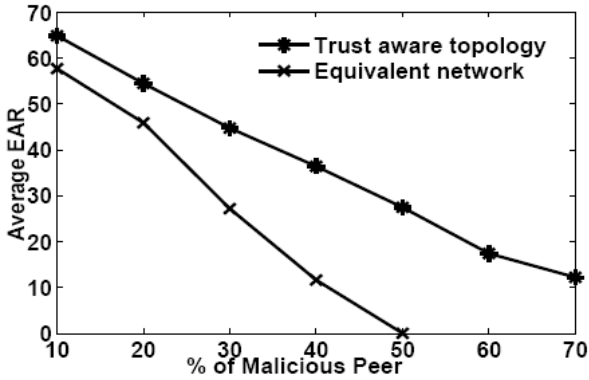

Fig. 8 Avg. EAR for various percentages of malicious peers in the network with and without the trust management module

The performance of the proposed protocol is compared with an equivalent power law network with no trust management framework in place. Since the proposed protocol allows addition of the community edges, therefore, to keep the number of edges in both the networks equal, additional edges are introduced between the similar peers in the equivalent network. Fig. 8 shows the comparison of the average EAR values. In the network without trust and reputation management, the value of EAR drops to zero when 50% or more of the peers in the network are malicious. However, in the network with the proposed protocol in place, even when 60% peers in the network are malicious, the value of EAR is consistently sustained at 20. This clearly demonstrates the robustness of the proposed protocol.

Fig. 9 shows the QMR experienced by both the types of peers for varying percentages of the malicious peers in the network. Initially, the value of QMR is high as no interest-based communities are formed and the searching is essentially a blind (i.e., brute force) one. As the protocol executes further, the peers with similar content interests come closer to each other (in terms

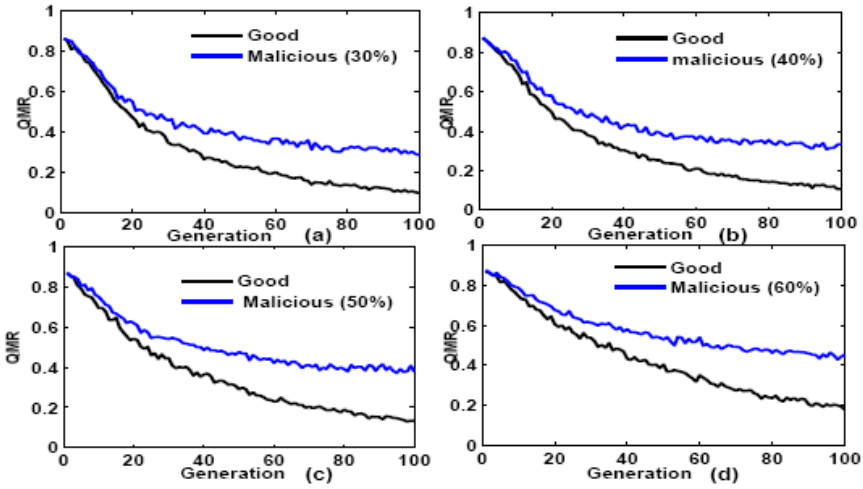

Fig. 9 QMR for various percentages of malicious peers in the network

of number of hops between them), and the queries are forwarded through the community edges. As a result, the value of QMR drops for the good peers. It is observed from Fig. 9 that the steady state value of the QMR for the good peers is less than 0.2, and the value of QMR is independent of the percentage of the malicious peers in the network. This is a significant performance achievement of the proposed protocol. For the malicious peers, the steady state value of QMR is 0.4. The high value of QMR for the malicious peers is due to the fact that the queries from the malicious peers are blocked by the good peers. It is evidently clear from the results that the proposed protocol effectively rewards the peers which share large number of authentic files in the network, which in turn helps in making the searching protocol efficient.

Fig. 10 shows variation of the value of HM for both the types of peers. Although, the value of HM for the good peers reaches a steady state as the topology matures, for the malicious peers, the value of HM fluctuates quite appreciably. The HM for the malicious peers sometimes attains higher values than that of the good peers. Since the queries forwarded by the malicious peers are blocked, HM for these peers are sometimes higher than those of the honest peers. The *hit* here does not mean authentic hit. The authentic hit of the good peers is higher than that of the malicious peers as these peers have higher AR values.

Fig. 11 shows the variation of the RIC for each type of peers under threat model A. It may be observed that the RIC for the good peers increases to 2.4 (constrained by the parameter *edge limit*), whereas for the malicious peers, the RIC does not increase beyond 1.2. With the increase in the percentage of malicious peers, the saturation rate slows down albeit the final value remains

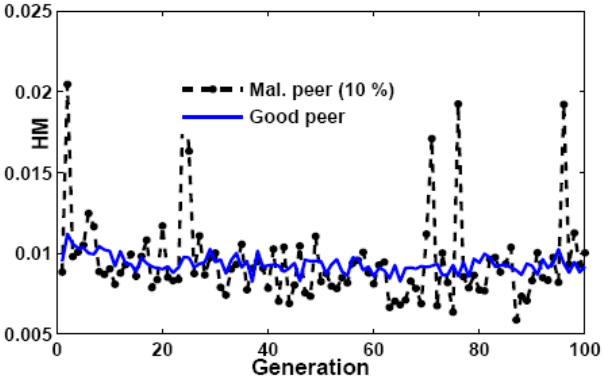

Fig. 10 HM for the malicious and the honest peers in the network. Percentage of malicious peers in the network is 10.

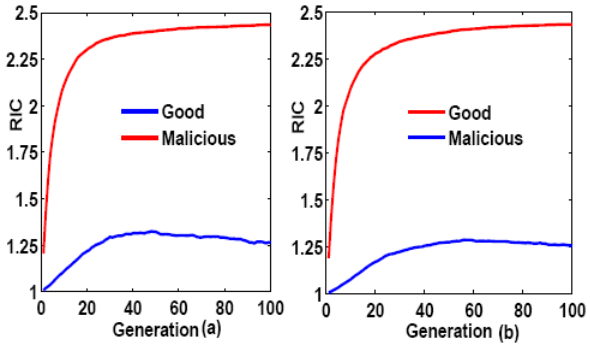

Fig. 11 RIC for various percentages of malicious peers under *threat model A*. In (a) 20% and in (b) 40% peers in the network are malicious.

the same. This shows that the proposed protocol provides better connectivity to the peers which share large number of authentic files. At the same time, the malicious peers are blocked gradually and their community edges are deleted.

Fig. 12 shows the variation of RIC under threat model B. Since in this model, a malicious peer provides fake files after it has achieved a high connectivity and then stops acting maliciously when it has lost sufficient number of community edges, fluctuation in the RIC persists throughout the simulation period.

Fig. 13 presents how the *closeness centrality* (CCen) of the good and the malicious peers varies in the community topology. In computation of CCen, only the community edges have been considered. It may be observed that the steady state value of the CCen for the good peers is around 0.12. However, for the malicious peers, the CCen value is found to lie in between 0.03 to 0.07. This demonstrates that the malicious peers are driven to the fringe of

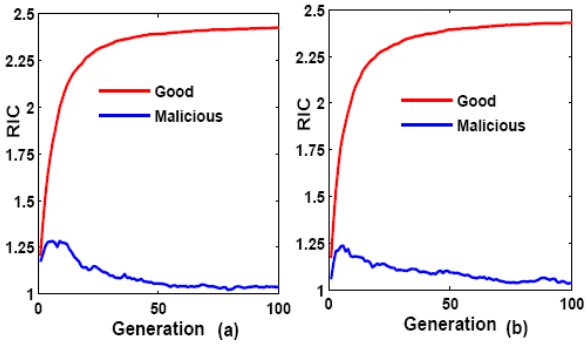

Fig. 12 RIC for various percentages of malicious peers under *threat model B*. In (a) 20% and in (b) 40% peers in the network are malicious.

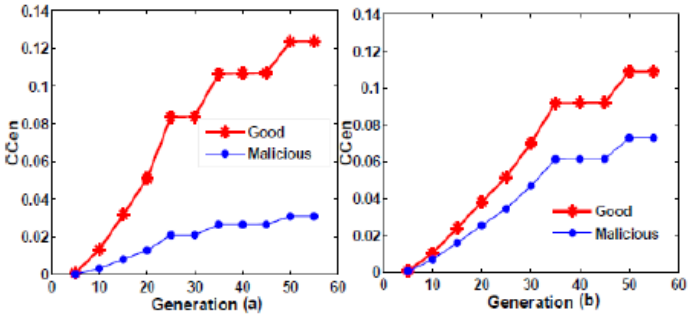

Fig. 13 Closeness centrality for various percentages of malicious peers in the network. In (a) 20% and (b) 40% nodes are malicious.

the network, while the good peers are allowed to form communities among them.

Higher values of CCen also indicate that the good peers have smaller average shortest path length between them. In the simulation, the diameter of the initial network is taken as 5. At the end of a simulation run, if there is no path between a pair of peers using the community edges, then the length of the shortest path between that pair is assumed to be arbitrarily long, say 15 (used in Fig. 14). As shown in Fig. 14, the *average shortest path distance* (ASPD) decreases from the initial value of 15 for both the honest and the malicious nodes. However, the rate and the extent of decrease for the good peers are much higher due to the formation of the semantic communities around them. For the malicious peers, after an initial fall, the value of ASPD increases consistently and finally almost reaches the maximum value of 15. On the other hand, the average value of ASPD for good peers is observed to be around 6. Since the good peers are connected with shorter paths, the query propagations and their responses will also be faster among these peers.

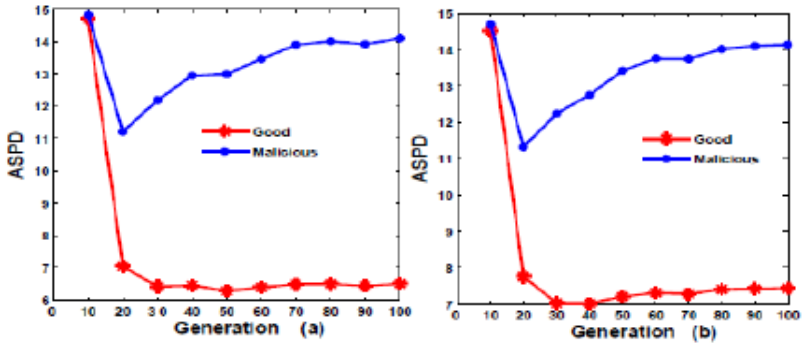

Fig. 14 Avg. shortest path distance vs. generations of search at the step of ten for various percentages of malicious peers. In (a) 30% and in (b) 40% nodes are malicious.

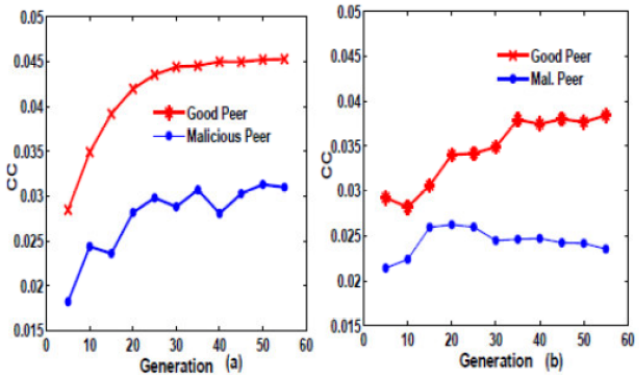

Fig. 15 Clustering coefficient for different percentages of malicious peers in the network. In (a) 20% and in (b) 40% of the peers are malicious.

Fig. 15 shows *clustering coefficient* (CC) for each type of peers. Since the community edges are added based on the download history and the peers having good reputation gain more community edges, the CC is high for the honest peers. This leads to the formation of triangles in the peer communities. To counter this phenomenon, the search strategy adapts itself from the BFS to the DFS to minimize redundant message flows in the network. Since the edges are added based on the download history and similarity of interest, the communities of the peers are formed which are connected to other community by hub of peers having interest in multiple content categories. This leads to lower ASPD for the good peers.

Fig. 16 depicts the size of the *largest connected component* (LCC) for each of the 32 content categories. It may be observed that the average size of the LCC for all content categories remains constant even if the percentage

of the malicious peers in the network increases. This clearly shows that the community formation among the good peers is not adversely affected by the presence of the malicious peers.

The average value of the LCC in the proposed protocol is further compared with that of an equivalent graph for various percentages of the malicious peers and the results are presented in Fig. 17. It may be observed in Fig. 17 that the value of the LCC in the proposed protocol remains almost constant irrespective of the percentage of the malicious peers in the network. However, the average LCC in the equivalent network (without the proposed protocol) falls sharply with the increase in the percentage of the malicious peers. This clearly shows that the proposed protocol is effective in forming the semantic communities for all types of file categories even in the presence of high percentage of the malicious peers.

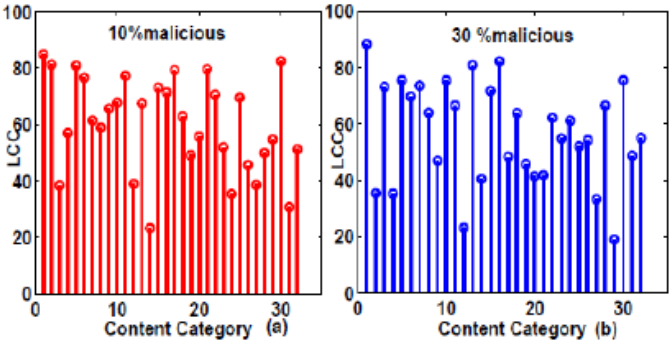

Fig. 16 Largest connected components (LCC) for different content categories

Fig. 18 shows that as the topology of the network matures, the steady state value of the *trust query propagation overhead* (TQPO) becomes quite low. The value of TQPO is less than 10 when 10% of the peers in the network are malicious. Even when the network has 40% of its peers malicious, TQPO gradually decreases and reaches a value of 20 in 100 generations. Hence, the trust propagation module has little impact on the system overhead, since the trust information is efficiently distributed in the trust-aware overlay topology.

The overhead due to the topology adaptation in the proposed protocol is also investigated. As mentioned in Section 4, the overhead due to the topology adaptation is measured by the metric TAO, which is defined as the number of community edges added or deleted in a generation. Fig. 19 shows the variation of TAO for different percentages of the malicious peers. It is observed that the value of TAO starts falling from an initial high value and oscillates with small amplitudes. This is due to the fact that initially the edge capacities of the peers are not saturated and they acquire the community edges

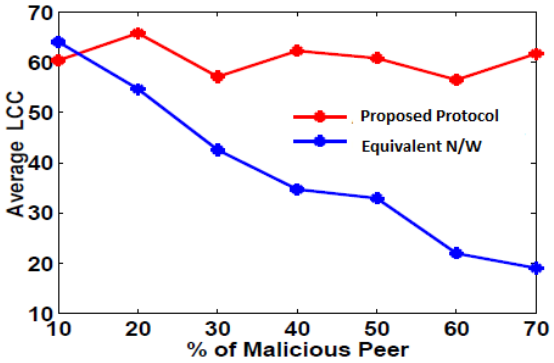

Fig. 17 Avg. LCC for different percentages of malicious peers in the network with the proposed protocol and without the protocol in an equivalent network

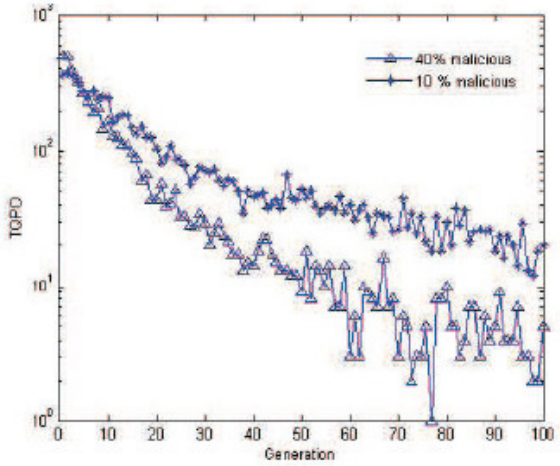

Fig. 18 Overhead of trust query propagation for 10% and 20% malicious peers in the network

rapidly. As the protocol executes further, the good peers acquire relatively stable neighborhood resulting in a sharp decrease in the value of TAO. In the subsequent generations, the value of TAO fluctuates slightly since the good peers delete the existing edges with the malicious peers as soon as the malicious peers are detected, and the new community edges with the fellow good peers are added. With the increase in percentage of the malicious peers, the fluctuation in the values of TAO also increases as more number of peers get added and deleted in the network. However, in all cases, the value of TAO falls sharply and attains a very low value once the community topology of

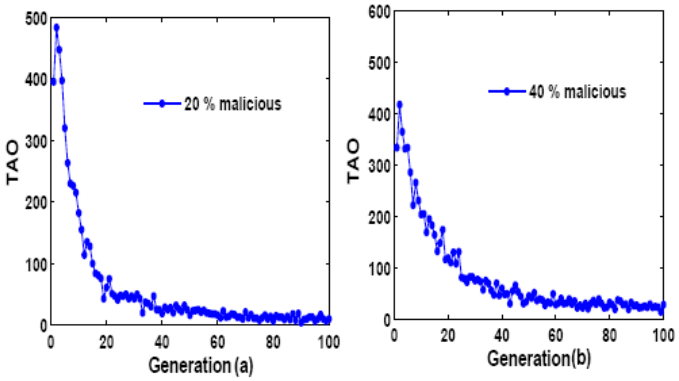

Fig. 19 Overhead due to topology adaptation under the presence of various percentages of malicious peers. In (a) 20% and in (b) 40% of the peers in the network are malicious.

Table 3 Avg. LCC values for different percentages of node churning

% of Node Churn	Avg. LCC
10	58
20	46
30	40
40	35
50	28

the peers becomes stable. This shows that the proposed protocol introduces a very small overhead in computation for topology adaptation process.

We have also studied how effectively the proposed protocol distinguishes the free riders from the malicious peers. As mentioned in Section 1, the free riders are those peers who do not share any resources with the other peers but they enjoy the resources of the other peers in the network. In the simulation, we have modeled the free riders as those peers who share up to 10 files in the content distribution model. The percentage of the free riders in the network is taken as 40. Since the free riders do not share any files, these peers will not be able to form any semantic communities. Accordingly, the RIC for these peers should be as low as those of the malicious peers. On the other hand, since unlike the malicious peers, the free riders do not distribute any spurious files, their presence does not cause much adverse impact on the network services. Hence, these peers are not penalized as much as the malicious peers. Accordingly, the AR values for the free riders should be higher than those of the malicious peers. The results presented in Fig. 20 validate this hypothesis.

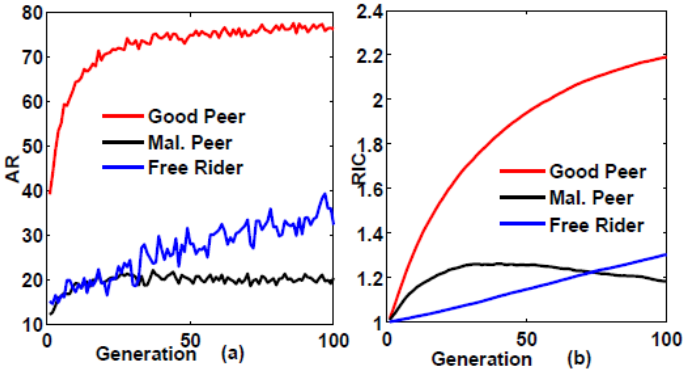

Fig. 20 AR and RIC values for the good peers, the free riders and the malicious peers for various percentages of malicious peers in the network. The percentage of free riders is taken as 40.

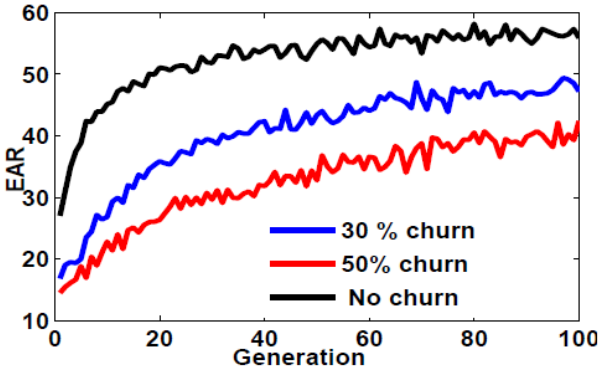

Fig. 21 Effect of node churning on EAR for various percentages of node churning with the proposed protocol in operation

Finally, we have studied the effect of *node churn* in the performance of the proposed protocol. Since, node churn is a natural phenomenon in a P2P network, it is essential that the search protocols should be efficient in the event of an occurrence of a large degree of node churn. Node churn causes disruption in the semantic communities of the peers. Hence, it leads to increase in the value of the QMR and decrease in the value of the EAR. Table 3 presents the average size of the LCC for various levels of node churn in the network, when the proposed protocol is under operation. Fig. 21 shows the effect of node churn on the EAR for various percentages of node churning. It may be observed that the performance of the proposed protocol degrades gracefully with the increase in the percentage of nodes being churned.

The performance of the proposed search protocol is summarized in Fig. 22.

Metrics	Values							
AR	Percentage of malicious peers							
	Peer type	10%			20%			
	Honest peers	90			80			
	Malicious peers	10			10			
	Free riders	35			35			
EAR	Percentage of malicious peers							
	10%	20%	30%	40%	50%	60%		
	85	70	60	50	40	25		
QMR	Percentages of malicious peers							
	Peer type	30%	40%	50%	60%			
	Honest peers	0.10	0.10	0.15	0.20			
	Malicious peers	0.30	0.35	0.40	0.45			
HM	Percentage of malicious peers : 10%							
	Honest peers – 0.01			Malicious peers - 0.02 (max), 0.00275 (min)				
RIC	Threat model A			Threat model B				
	Percentage of malicious peers			Percentage of malicious peers				
	Peer types	20%	40%	Peer types	20%	40%		
	Honest peers	2.4	2.4	Honest peers	2.4	2.4		
	Malicious peers	1.2	1.25	Malicious peers	1.0	1.0		
CCen	Percentage of malicious peers							
	Peer type	20%		40%				
	Honest peers	0.12		0.11				
	Malicious peers	0.03		0.07				
CC	Percentage of malicious peers							
	Peer type	20%		40%				
	Honest peers	0.045		0.039				
	Malicious peers	0.030		0.022				
ASPD	Percentage of malicious peers							
	Peer type	30%		40%				
	Honest peers	6		7				
	Malicious peers	Infinity (14 in simulation)		Infinity (14 in simulation)				
LCC	Percentage of malicious peers							
	Protocols	10%	20%	30%	40%	50%	60%	70%
	Proposed protocol	60	68	56	62	59	55	61
	Equivalent network	67	55	40	35	32	23	20
TQPO	Percentage of malicious peers							
	10%			40%				
	10			20				
TAO	Percentage of malicious peers							
	20			40				
	Max: 490 Min: 0			Max: 420 Min: 20				
Node Churning	Percentage of node churn in the network							
		10	20	30	40	50		
	Avg. LCC	58	46	40	35	28		
	EAR	55	-	48	-	40		
Trust	The trust computation is based on beta-distribution of reputation which is computationally very efficienct.							
Privacy	DHT entries are encrypted /decrypted using 1024-bit RSA key pairs. The identity of the peers and privacy of the data contents both can be protected. The overhead of computing and message overhead depends on how easily a trusted peer can be selected.							

Fig. 22 Summary of the performance metrics of the proposed search protocol

Comparisons with Existing Protocols: In the following, we provide a brief comparative analysis of the proposed protocol with two similar protocols existing in the literature. In [34], a method named *eigen trust* has been proposed to minimize the impact of the malicious peers on the performance of a P2P system. In this scheme, the global trust value for each peer is computed by calculating the left principal eigen vector of a matrix of normalized local trust values. Since the trust and reputation computations are robust, the mechanism is able to sustain a high value of the AR (i.e. the fraction of authentic file downloads) for the good peers even when the percentage of the malicious peers is as high as 80. In contrast, the proposed protocol in this chapter can support a high value of AR for the good nodes as long as the percentage of the malicious peers in the network does not exceed 60. However, the scheme based on eigen trust is computationally intensive, and it is susceptible to produce unreliable results in the event of any Byzantine failures of some of the peers. On the other hand, the proposed protocol in this chapter has a light-weight trust management module that is robust yet efficient in identifying the free riders and Byzantine peers while improving on the QoS of searching.

In the APT protocol [16], as the topology stabilizes, all the paths from the good peers to the malicious peers are blocked, and the characteristic path lengths of these two types (good and malicious) of peers are distinctly different - while the good peers have shorter path lengths between them, the malicious peers are driven to the fringe of the network. However, in the proposed protocol in this chapter, the good peers and the malicious peers still remain connected through the connectivity edges since these edges are not deleted during the protocol operation. The presence of the connectivity edges prevents any possibility of network partitioning, which makes the protocol more robust and fault-tolerant. Moreover, the scalability of the proposed protocol is higher than that of the APT protocol, since it uses a light-weight trust engine. More importantly, the APT protocol does not have any mechanism to protect the privacy of the peers. The proposed protocol provides a very robust and reliable mechanism for protecting the privacy of the peers and their data. This makes it more suitable for deployment in the real-world P2P networks.

A comparative analysis of three protocols - the APT protocol [16], the RC-ATP protocol [55], and the proposed protocol in this chapter- is presented in Fig. 23. It can be observed that the proposed protocol outperforms the other two protocols in terms of its higher scalability, robustness against network partitioning, and its ability protect privacy of the peers and the messages communicated in the network.

Metrics		APT Protocol	RC-ATP Protocol	Proposed Protocol
Quality of search	AR	High	High	High
	EAR	High	High	Very high
Search efficiency	HM	Very Low (honest peers) Low (malicious peers)	Very low (honest peers) Low (malicious peers)	Very low (honest peers) Low (malicious peers)
	QMR	Very low (honest peers) High (malicious peers)	Very low (honest peers) High (malicious peers)	Very low (honest peers) High (malicious peers)
Topology adaptation	RIC	High (honest peers) Low (malicious peers)	High (honest peers) Low (malicious peers)	Very high (honest peers) Low (malicious peers)
	CCen	Very high (honest peers) Very low (malicious peers)	Very high (honest peers) Very low (malicious peers)	Very high (honest peers) Very low (malicious peers)
	CC	High (honest peers) Low (malicious peers)	High (honest peers) Low (malicious peers)	High (honest peers) Low (malicious peers)
	ASPD	Very low (honest peers) Very high (malicious peers)	Very low (honest peers) Very high (malicious peers)	Very low (honest peers) Very high (malicious peers)
	LCC	High	High	High
Trust management		Simple and vulnerable to attacks	Simple and vulnerable to attacks	Robust and resistant to various attacks such as: ballot stuffing attack, bad-mouthing attack etc
Node churn		Node churning is not considered	High EAR and LCC maintained in the event of node churning	High EAR and LCC maintained in the event of node churning
Free riders		Free riders are punished	Free riders are punished	Free riders are punished
Incentive to the good peers		Good peers are provided incentives by semantic community formation and topology adaptation	Good peers are provided incentives by semantic community formation and topology adaptation	Good peers are provided incentives by semantic community formation and topology adaptation
Privacy of the peers and data		No privacy protection	No privacy protection	Peer and data privacy are protected.
Scalability		Not scalable	Scalable	Highly scalable
Robustness		Not resistant to network partitioning and Byzantine failure of peers	Not resistant to network partitioning and Byzantine failure of peers	Robust against network partitioning and Byzantine failure of peers

Fig. 23 A comparative analysis of three protocols - APT, RC-ATP and the proposed protocol

6 Conclusion

In many IoT applications, resource discovery protocols are required which need to perform efficiently in a distributed and large-scale environment. An efficient and secure search protocol for unstructured P2P networks will be an ideal candidate for this purpose. Hence, the P2P architectures and their protocols are finding increasing relevance and adoption in IoT middleware

design. In this chapter, we have presented a search protocol for unstructured P2P networks that solves several problems e.g., inauthentic downloads, poor search scalability, combating free riders, and protecting the user and the data privacy. The protocol exploits the topology adaptation done by the peers and uses a robust trust management mechanism to isolate the malicious peers while providing topologically advantageous positions to the good peers. Due to the topology adaptation, the good peers are able to form semantic communities which enables them get faster and authentic responses to their queries. On the other hand, the malicious peers are driven to the fringes of the network so that the queries from these peers have longer paths to travel to receive responses. In some situations, the queries from the malicious peers are blocked so that these peers do not receive any response to their queries at all. A large number of metrics are defined for evaluating the performance of the proposed protocol, and the protocol is simulated in a power-law network. The simulation results have demonstrated that the protocol is robust even in presence of a large percentage of malicious peers in the network. A detailed comparative analysis of the performance of the protocol is made with two existing similar protocols so that the advantages of the proposed protocol can be clearly understood. As a future plan of work, we intend to carry out an analysis of the message overhead of the privacy module under different network topologies and for different selection strategies of the trusted peers.

References

1. Abdul-Rahman, A., Hailes, S.: A Distributed Trust Model. In: Proceedings of the Workshop on New Security Paradigms (NPW 1997), Langdale, Cumbria, United Kingdom, pp. 48–60 (1997)
2. Aberer, K., Despotovic, Z.: Managing Trust in a Peer-2-Peer Information System. In: Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM 2001), Atlanta, Georgia, USA, pp. 310–317 (2001)
3. Aberer, K.: P-Grid: A Self-Organizing Access Structure for P2P Information Systems. In: Batini, C., Giunchiglia, F., Giorgini, P., Mecella, M. (eds.) CoopIS 2001. LNCS, vol. 2172, pp. 179–194. Springer, Heidelberg (2001)
4. Adamic, L.A., Lukose, R.M., Puniyani, A.R., Huberman, B.A.: Search in Poer Law Networks. Physics Review E64, 46135–46143 (2001)
5. Atzori, L., Lera, A., Morabito, G.: The Internet of Things: A Survey. Computer Networks 54(15), 2787–2805 (2010)
6. Balfe, S., Lakhani, A.D., Paterson, K.G.: Trusted Computing: Providing Security for Peer-to-Peer Networks. In: Proceedings of the 5th IEEE International Conference on Peer-to-Peer Computing (P2P 2005), Konstanz, Germany, pp. 117–124 (2005)
7. Bandyopadhyay, D., Sen, J.: Internet of Things: Applications and Challenges in Technology and Standardization. Wireless Personal Communications, Special Issue on Distributed and Secure Cloud Clustering (DISC) 58(1), 49–69 (2011)
8. Barabasi, A.L., Albert, R.: Emergence of Scaling in Random Networks. Science 286, 509–512 (1999)

9. Bloom, B.: Space-Time Trade-Offs in Hash Coding with Allowable Errors. Communications of the ACM 13(7), 422–426 (1970)
10. Buchegger, S., Boudec, J.Y.L.: The Effect of Rumor Spreading in Reputation Systems for Mobile Ad-hoc Networks. In: Proceedings of Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt 2003), Sophia-Antipolis, France, pp. 131–140 (2003)
11. Buchegger, S., Boudec, J.Y.L.: Coping with False Accusations in Misbehavior Reputation Systems for Mobile Ad-hoc Networks. EPFL Technical Report No: IC/2003/50 (2003)
12. Cha, B.R., Kim, J.G.: Handling Fake Multimedia Contents Threat with Collective Intelligence in P2P File Sharing Environments. In: Proceedings of the International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC 2010), Fukuoka, Japan, pp. 258–263 (2010)
13. Chaum, D.: The Dining Cryptographers Problem: Uncontrolled Sender and Recipient Untraceability. Journal of Cryptology 1(1), 65–75 (1998)
14. Clarke, I., Miller, S., Hong, T., Sandberg, O., Wiley, B.: Protecting Free Expression Online with FreeNet. IEEE Internet Computing 6(1), 40–49 (2002)
15. Clarke, I., Sandberg, O., Wiley, B., Hong, T.W.: Freenet: A Distributed Anonymous Information Storage and Retrieval System. In: Federrath, H. (ed.) Anonymity 2000. LNCS, vol. 2009, p. 46–66. Springer, Heidelberg (2001)
16. Condie, T., Kamvar, S.D., Garcia-Molina, H.: Adaptive Peer-to-Peer Topologies. In: Proceedings of the 4th International Conference on Peer-to-Peer Computing (P2P 2004), Zurich, Switzerland, pp. 53–62 (2004)
17. Crespo, A., Garcia-Molina, H.: Semantic Overlay Networks for P2P Systems. In: Moro, G., Bergamaschi, S., Aberer, K. (eds.) AP2PC 2004. LNCS (LNAI), vol. 3601, pp. 1–13. Springer, Heidelberg (2005)
18. Dellarocas, C.: Immunizing Online Reputation Reporting Systems against Unfair Ratings and Discriminatory Behavior. In: Proceedings of the 2nd ACM Conference on Electronic Commerce (EC 2000), Minneapolis, MN, USA, pp. 150–157 (2000)
19. Damiani, E., di Vimecati, D.C., Paraboschi, S., Samarati, P., Violante, F.: Reputation-Based Approach for Choosing Reliable Resources in Peer-to-Peer Networks. In: Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002), Washington DC, USA, pp. 207–216 (2002)
20. de Mello, E.R., van Moorsel, A., da Silva Fraga, J.: Evaluation of P2P Search Algorithms for Discovering Trust Paths. In: Wolter, K. (ed.) EPEW 2007. LNCS, vol. 4748, pp. 112–124. Springer, Heidelberg (2007)
21. Desmed, Y.G.: Threshold Cryptography. European Transactions on Telecommunications 5(4), 449–457 (1994)
22. Dingledine, R., Freedman, M.J., Molnar, D.: Accountability Measures for Peer-to-Peer Systems. In: Peer-to-Peer: Harnessing the Power of Disruptive Technologies, ch. 16. O'Reilly and Associates (2000)
23. Douceur, J.R.: The Sybil Attack. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 251–260. Springer, Heidelberg (2002)
24. Ganeriwal, S., Srivastava, M.B.: Reputation-Based Framework for High Integrity Sensor Networks. In: Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN 2004), Washington DC, USA, pp. 66–77 (2004)

25. Ganeriwal, S., Balzano, L.K., Srivastava, M.B.: Reputation-Based Framework for High Integrity Sensor Networks. ACM Transactions on Sensor Networks (TOSN) 4(3), Article No. 15 (2008)
26. Gkantsidis, C., Mihail, M., Saberi, A.: Hybrid Search Schemes for Unstructured Peer-to-Peer Networks. In: IEEE INFOCOM (2005)
27. Goel, S., Robson, M., Pole, M., Sirer, E.: Herbivore: A Scalable and Efficient Protocol for Anonymous Communication. Cornell University, CIS Technical Report TR2003-1890 (2003)
28. Goldschlag, D., Reed, M., Syverson, P.: Onion Routing for Anonymous and Private Internet Connections. Communications of the ACM 42(2), 39–41 (1999)
29. Guo, L., Yang, S., Guo, L., Shen, K., Lu, W.: Trust-Aware Adaptive P2P Overlay Topology Based on Super-Peer-Partition. In: Proceedings of the 6th International Conference on Grid and Cooperative Computing (GCC 2007), Urumchi, Xinjiang, China, pp. 117–124 (2007)
30. Hsiao, H.C., Liao, H., Huang, C.C.: Resolving the Topology Mismatch Problem in Unstructured Peer-to-Peer Networks. IEEE Transactions on Parallel and Distributed Systems 20(11), 1668–1681 (2009)
31. Huang-Fu, C.C., Lin, Y.B., Rao, H.: IP2P: A Peer-to-Peer System for Mobile Devices. IEEE Wireless Communications 16(2), 30–36 (2009)
32. Joung, Y.J., Lin, Z.W.: On the Self-Organization of a Hybrid Peer-to-Peer System. Journal of Network and Computer Applications 33(2), 183–202 (2010)
33. Jsang, A.: A Logic for Uncertain Probabilities. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9(3), 279–311 (2001)
34. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The Eigen Trust Algorithm for Reputation Management in P2P Networks. In: Proceedings of the 12th International Conference on World Wide Web (WWW 2003), Budapest, Hungary, pp. 640–651 (2003)
35. Leighotn, F.T., Rao, S.: An Approximate Max-Flow Min-Cut Theorem for Uniform Multicommodity Flow Problem with Applications to Approximate Algorithms. In: Proceedings of the 29th IEEE Symposium on Foundations of Computer Science (FOCS 1988), pp. 422–431 (1988)
36. Li, X., Wang, J.: A Global Trust Model of P2P Network Based on Distance-Weighted Recommendation. In: Proceedings of IEEE International Conference of Networking, Architecture, and Storage (NAS 2009), Zhang Jia Jie, Hunan, China, pp. 281–284 (2009)
37. Li, Z., Xie, G., Li, Z.: Efficient and Scalable Consistency Maintenance for Heterogeneous Peer-to-Peer Systems. IEEE Transactions on Parallel and Distributed Systems 19(12), 1695–1708 (2008)
38. Lin, T., Lin, P., Wang, H., Chen, C.: Dynamic Search Algorithm in Unstructured Peer-to-Peer Networks. IEEE Transactions on Parallel and Distributed Systems 20(5), 654–666 (2009)
39. Lu, Y., Wang, W., Xu, D., Bhargava, B.: Trust-Based Privacy Preservation for Peer-to-Peer Data Sharing. IEEE Transaction on Systems Man and Cybernetics (Special issues based on best papers in Secure Knowledge Management Conference) 36(3), 498–502 (2006)
40. Martinez-Yelmo, I., Bikfalvi, A., Cuevas, R., Guerrero, C., Garcia, J.: H-P2PSIP: Interconnection of P2PSIP Domains for Global Multimedia Services Based on a Hierarchical DHT Overlay Network. Computer Networks 53(4), 556–568 (2009)

41. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A Scalable Content Addressable Network. In: Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2001), San Diego, California, USA, pp. 161–172 (2001)
42. Reed, M., Syverson, P., Goldschlag, D.: Anonymous Connections and Onion Routing. IEEE Journal on Selected Areas in Communications. Special Issue on Copyright and Privacy Protection 16(4), 482–494 (1998)
43. Reiter, M.K., Rubin, A.D.: Crowds: Anonymity for Web Transactions. ACM Transactions on Information and Systems Security 1(1), 66–92 (1998)
44. Risson, J., Moors, T.: Survey of Research Towards Robust Peer-to-Peer Networks. Computer Networks 50(7), 3485–3521 (2006)
45. Santucci, G.: From Internet of Data to Internet of Things. In: Proceedings of the 4th International Conference on Future of Internet Technology, Seoul, Korea (2009)
46. Scarlata, V., Levine, B., Shields, C.: Responder Anonymity and Anonymous Peer-to-Peer File Sharing. In: Proceedings of IEEE International Conference on Network Protocols (ICNP 2001), Riverside, CA, USA, p. 272 (2001)
47. Schafer, J., Malinks, K., Hanacek, P.: Peer-to-Peer Networks Security. In: Proceedings of the 3rd International Conference on Internet Monitoring and Protection (ICIMP 2008), Bucharest, Romania, pp. 74–79 (2008)
48. Schlosser, M.T., Condie, T.E., Kamvar, S.D., Kamvar, A.D.: Simulating a P2P File-Sharing Network. In: Proceedings of the 1st Workshop on Semantics in P2P and Grid Computing, Budapest, Hungary (2002)
49. Sen, J.: A Secure and Efficient Searching Scheme for Trusted Nodes in a Peer-to-Peer Network. In: Herrero, Á., Corchado, E. (eds.) CISIS 2011. LNCS, vol. 6694, pp. 100–108. Springer, Heidelberg (2011)
50. Sen, J.: Secure and User-Privacy Preserving Searching in Peer-to-Peer Networks. International Journal of Communication Networks and Information Security (IJCNIS) 4(1), 29–40 (2012)
51. Shafer, G.: A Mathematical Theory of Evidence. Princeton University (1976)
52. Shamir, A.: How to Share a Secret. Communications of the ACM 22(11), 612–613 (1979)
53. Sherwood, R., Bhattacharjee, B., Srinivasan, A.: P5: A Protocol for Scalable Anonymous Communication. In: Proceedings of IEEE Symposium on Security and Privacy, Oakland, CA, pp. 53–65 (2002)
54. Sit, E., Morris, R.: Security Considerations for Peer-to-Peer Distributed Hash Tables. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 261–269. Springer, Heidelberg (2002)
55. Tain, H., Zou, S., Wang, W., Cheng, S.: Constructing Efficient Peer-to-Peer Overlay Topologies by Adaptive Connection Establishment. Computer Communication 29(17), 3567–3579 (2006)
56. Tang, X., Xu, J., Lee, W.C.: Analysis of TTL-Based Consistency in Unstructured Peer-to-Peer Networks. IEEE Transactions on Parallel and Distributed Systems 9(12), 1683–1694 (2008)
57. Tang, Y., Wang, H., Dou, W.: Trust Based Incentive in P2P Network. In: Proceedings of the IEEE International Conference on E-Commerce Technology for Dynamic E-Business, Beijing, China, pp. 302–305 (2004)
58. Vazirani, V.V.: Approximation Algorithms. Springer, Berlin (2001)
59. Waldman, M., Rubin, A.D., Cranor, L.F.: Publius: A Robust, Tamper-Evident, Censorship-Resistant, Web Publishing System. In: Proceedings of the 9th USENIX Security Symposium, Denver, Colorado, USA, pp. 59–72 (2000)

60. Xiao, L., Liu, Y., Lionel, M.N.: Improving Unstructured Peer-to-Peer Systems by Adaptive Connection Establishment. IEEE Transaction on Computers 54(9), 1091–1103 (2005)
61. Xie, C., Chen, G., Vandenberg, A., Pan, Y.: Analysis of Hybrid P2P Overlay Network Topology. Computer Communications 31(2), 190–200 (2008)
62. Xiong, L., Liu, L.: A Reputation-Based Trust Model for Peer-to-Peer E-Commerce Communities. In: Proceedings of the 4th IEEE/ACM Conference on E-Commerce (CEC 2003), Newport Beach, California, USA, pp. 228–229 (2003)
63. Xioreng, L., Liu, L.: Peer-Trust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities. IEEE Transactions on Knowledge and Data Engineering 16(7), 843–857 (2004)
64. Xiao, L., Xu, Z., Zhang, X.: Low-Cost and Reliable Mutual Anonymity Protocols in Peer-to-Peer Networks. IEEE Transactions on Parallel and Distributed Systems 14(9), 829–840 (2003)
65. Yang, M., Yang, Y.: An Efficient Hybrid Peer-to-Peer System for Distributed Data Sharing. IEEE Transactions on Computers 59(9), 1158–1171 (2010)
66. Zhang, R., Hu, Y.C.: Assisted Peer-to-Peer Search with Partial Indexing. IEEE Transactions on Parallel and Distributed Systems 18(8), 1146–1158 (2007)
67. Zhu, Z., Kalnis, P., Bakiras, S.: DCMP: A Distributed Cycle Minimization Protocol for Peer-to-Peer Networks. IEEE Transactions on Parallel and Distributed Systems 19(3), 363–377 (2008)
68. Zhuge, H., Chen, X., Sun, X.: Preferential Walk: Towards Efficient and Scalable Search in Unstructured Peer-to-Peer Networks. In: Proceedings of the 14th International Conference on World Wide Web (WWW 2005), Chiba, Japan, pp. 882–883 (2005)