A Role-Based Trust Model for Peer-to-Peer Communities and Dynamic Coalitions

Mujtaba Khambatti
Microsoft Corporation
mujtabak@microsoft.com

Partha Dasgupta
Arizona State University
partha@asu.edu

Kyung Dong Ryu
Arizona State University
kdryu@asu.edu

Abstract

Although P2P systems are usually used for information exchange between peers, they have either protected peers' anonymity, or required transacting peers to trust each other implicitly. Both these approaches are vulnerable to attacks by malicious peers who could abuse the P2P system to spread viruses, incorrect, or damaging information.

In this paper, we propose an approach for trust management in P2P systems. We introduce an optimistic role-based model for trust amongst peers and show that it is scalable, dynamic, revocable, secure and transitive. Our proposed solution permits asymmetric trust relationships that can be verified by any peer in the system through a simple, low-cost algorithm. This paper introduces a metric known as iComplex that combines a peer's trust value for each of its roles into a single, relative, probabilistic guarantee of trust. Finally, we discuss how our trust model allows peers to revoke relationships with malicious peers, and the non-repudiation of peer relations.

We use simulations to illustrate the trust value distribution amongst peers in the network. Our analysis and experiments demonstrates the low-cost involved to verify and validate trust values. Lastly, we establish the effectiveness of using sum as the aggregation function to combine trust values of a peer.

Keywords: Communities, Dynamic Coalitions, Peer-to-Peer, Trust.

1. Introduction

The emergence of decentralized and dynamic file-sharing applications, such as Napster [1] and Gnutella [2], provided the catalyst that drew a lot of attention to a new breed of distributed systems called peer-to-peer (P2P) systems. Current peer-to-peer systems are often targeted for global information sharing, replicated file storage, and searching by using an end-to-end overlay network. Although these systems usually involve information exchange between peers, they have either protected peers' anonymity [3, 4], or required transacting peers to trust each other implicitly [2].

Both these approaches are vulnerable to attacks by malicious peers who could abuse the P2P system to spread viruses, incorrect, or damaging information. Therefore in order to enable practical information sharing in such decentralized and dynamic systems, a viable trust model needs to be incorporated that will allow peers to have varying amounts of dynamically changeable trust amongst each other. The main challenges that need to be addressed are: how to describe if a peer is trustworthy, what low-cost verification algorithm can be executed by a peer to determine the trust value of some other peer, how are trust values about peers exchanged within the system, how can dishonest peers be punished.

In this paper, we propose an approach for trust management in P2P systems. We introduce a role-based model for trust amongst peers and show that it is scalable, dynamic, revocable, secure and transitive. The trust model assigns role-based trust values to peers proportional to their status in the system. The status of a peer depends on its relationships with other peers. Our proposed solution permits asymmetric trust relationships that can be verified by any peer in the system through a simple, low-cost algorithm. Since trust values are proportional to the status of a peer, it is essential to ensure that relationships between any two peers will be legally binding and have non-repudiation; that is peers cannot falsely deny their relationship with another peer. However it is equally essential that peers have the ability to revoke their relationships with malicious peers to punish them for false or damaging information. Finally, this paper introduces a metric that combines a peer's trust value for each of its roles. The combined trust value is a single, relative, probabilistic guarantee that offers peers with a simple, verifiable trust metric about other peers in the P2P system.

Traditionally, many of these functionalities were implemented through exhaustive policy lists that needed to be created at system design time or through a complex predetermined role framework, such as role-based access control (RBAC) [5]. In contrast, our trust management system is dynamic and requires minimal global knowledge. Further, the decentralized nature of our algorithms makes it suitable for P2P systems.

1.1. Dynamic Coalitions

The research described in this paper is part of a larger project known as *Dynamic Coalitions* [6]. Dynamic Coalitions enables a set of partners to work together while sharing information, resources, and capabilities in a controlled and accountable fashion. The partners themselves are organizations composed of people, departments, computational entities, and agents who perform tasks consistent with the internal rules of their organization.

Coalitions are supported by several innovative techniques such as transitive delegation, cryptographic file systems, capacity sandboxing, reverse sandboxing, and fine-grained access control. These techniques facilitate scalable authentication and revocable authorization of agent computations even when they span resources of different organizations. In addition, they improve overall efficiency by permitting migration of computations to, and a caching of services in, partly trusted environments of another organization.

1.2. Communities of Peers

In an electronically connected world, people use network-addressable[1] computing elements (such as a desktop personal computer, a laptop computer, a personal digital assistant, and so on), which we call *peers.* Peers have comparable roles and responsibilities and are used by their owners to communicate information, share or consume services and resources with other peers whom they know. Every peer belongs to at least one pre-determined group corresponding to the department or organization of its human user. For home users, the domain name of an Internet connection is used to identify the pre-determined group of the peer. Thus the basic construct of P2P systems can be used to implement a practical Dynamic Coalition environment where coalitions are created between peers in different groups.

In our research, we investigate a generalization of the notion of peer group to a multiplicity of groups (possibly overlapping) called *peer communities*. While a group is a physical collection of objects, a community is a set of active members, who are involved in sharing, communicating and promoting a common interest.

Our concept of peer communities is loosely based on the idea of "interest groups", such as Yahoo Groups [7], Usenet Newsgroups, or web communities. The user of a peer in the system claims to have some interests and depending upon the claims of all the peers' users, communities are implicitly formed (made up of peers with the same or similar interests). Note that communities are formed implicitly, i.e. they are self organizing. If a peer in New York declares an interest in wombats, and a peer in China also declares the same interest, then the two peers become part of an implicit, undiscovered community. A peer may belong to many different communities and communities may overlap.

1.3. Canonical Application

The canonical application that we consider for our algorithms is a digital library built out of a collection of peers in which each peer owns a set of books that it is willing to share with other peers (assume these are non-copyrighted works). The subjects of the books owned by a peer form its set of interests. Peers are implicitly grouped into communities based on the common interests they share. Because a peer could own books from a variety of subjects, we can imagine that a peer could be a member of multiple communities.

1.4. Limitations of Our Approach

In terms of limitations, the techniques that we developed can only be applied to specific situations, such as the digital library application, where the set of interests is constrained, well defined and understood by almost all the peer members. Our proposed algorithms would place individual users into peer communities based on the common interests that they share with other peers. A generalized peer-to-peer system, where the set of interests includes the universe of all possible interests, might not contain a single peer that shares common interests with other peers. Therefore no communities would form.

Additionally, in real life, links between peers are not always bidirectional. We make this assumption to simplify the process of network formation.

Some of the other problems that are outside the scope of this dissertation are: specifying how static IP addresses can be used as peer identities; providing more than just probabilistic guarantees of trust, therefore the trust paradigm is vulnerable to peers that lie infrequently and with due measure; describing the channel or protocol of communication used by peers; defining the format for the interests of a peer, obtaining the interests from a peer; and fragmentation of the network after targeted denial-of-service attacks on certain peers.

2. Terms and Definitions

We argue that peer communities provide an implicit and natural organization of peers in structures that can be efficiently uncovered or cultivated. In our previous work,

[1] We assume that each peer has a static IP address that serves as the *peer identity*. While this assumption is not universally true, it can be facilitated through various techniques (dynamic DNS, IPv6, or firewall penetrating mechanisms) that are outside the scope of this paper.

we provided a motivation for the study of peer-to-peer communities and illustrated some scenarios to define and discover peer communities [8]. Using simulated models of communities, we have gained an insight to the architecture of randomly created communities. Our algorithm for the discovery of communities allows for the computation of *Link Weights*, a very important value that enables the working of all our subsequent algorithms. Link weights help determine the membership of a peer in a community. They are also used to rank peers in a community for the purposes of information dissemination and trust. For a detailed understanding of our algorithms for community formation, discovery, information dissemination and search, we refer readers to [8, 9, 10].

In this section, we briefly explain some of the terms from our earlier work. First we describe what we mean by interests. We then provide a definition of P2P communities, followed by a thorough treatment of the subject of peer links. Finally at the end of this section we define Link Weights.

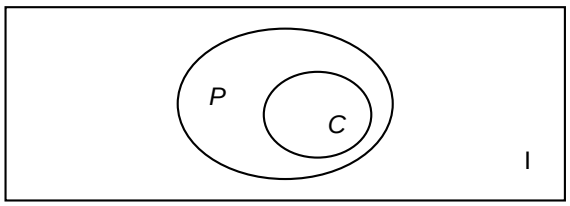


Figure 1. Venn diagram of interest attribute sets for a peer. *I* is the universe of all attributes; *P* is the set of personal attributes; and *C* is the set of claimed attributes.

2.1. Interest Attributes

In our model, interests are represented by attributes, which are used to determine the peer communities in which a particular peer would participate. There are of course privacy and security concerns in using such information, so we divide interests into two classes – *personal* and *claimed*.

The full set of attributes for a peer is called *personal attributes*. However, for privacy and/or security reasons, all these attributes may not be used to determine community membership. A peer may also not want to reveal some of her personal attributes because she might not consider them relevant amongst the peers that she knows. Hence, a peer explicitly makes only a subset of these attributes public, which are called *claimed attributes* (see Fig. 1).

2.2. P2P Communities

Below, we formally define a peer-to-peer community based on the attributes of each peer.

PEER-TO-PEER COMMUNITY: *The non-empty set N of nodes is a peer-to-peer community iff N has a non-null signature.*

SIGNATURE: *Let i be a node and C_i be a set that contains attributes claimed by i. Consider a non-empty set N of nodes. Then the set resulting from the intersection of C_k, for all $k \in N$ is called a signature of the set N.*

With this definition, given any collection of peers, we would be able to tell whether the collection is a peer-to-peer community or not.

2.3. Peer Links

We observe in projects like HITS [11] and Web Communities [12] the concept of self-organized communities that form implicitly based on hypertext links between web pages. The human creators of the web page explicitly place these links typically in order to point towards web pages with similar content. This is one of the factors that Internet search engines have exploited to enhance their search operation.

We draw an analogy from the above research to understand the behavior of peer-to-peer systems. We find that peers also regularly link to other peers, in the form of relationships (being present in their address book), or direct connections (being on the same network), when their human owners share something in common. We assume that these links are bi-directional communication channels that can be established on an as-needed basis. In a social network, this is similar to getting in touch with people you know when you need something. We refer to these end-to-end overlay communication channels as *peer links*.

2.3.1. The Need for Peer Links. Links are not necessary to form and manage peer-to-peer communities. However, they are needed to to feasibly run low-cost algorithms such as community formation and discovery. We therefore introduce the notion of a set of *neighbors*, which are directly (1-hop) linked peers. The neighbors of a peer help when a peer needs to communicate with other peers that are not directly linked to it.

When node *X* is born, it needs to have one or more logical neighbors. If it has three neighbors, *A*, *B*, and *C*, then we say that it has three links, $X \rightarrow A$, $X \rightarrow B$, and $X \rightarrow C$. Unlike overlay networks used by Distributed Hash Table (DHT) based peer-to-peer systems, our link based

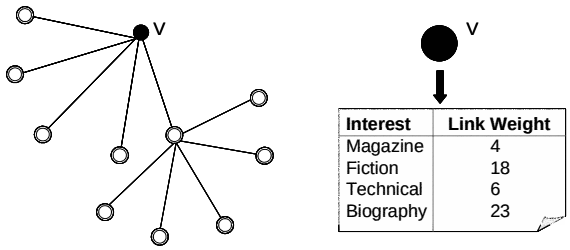


Figure 2. (*left*) Peer *V* compares its claimed interest attributes with all 1-hop and 2-hop neighbors. (*right*) As a result of the comparison, *V* calculates Link Weights for each claimed interest attribute.

network is not contingent to node names, but to user selected neighbors. Like in social networks, the more links a node acquires the more successful it will be in receiving information and searching the peer-to-peer network. It is the responsibility of each peer to acquire as many neighbors as possible.

First, let us explain a case where links are essential. Suppose a node, belonging to domain abc.com claims the attribute "baseball". This node is essentially isolated, unless it *a priori* knows about the other members of the baseball community or the other members of the abc.com community. There is a need for a "seed" to start the community formation and information search needs.

2.3.2. Creating Peer Links. Flooding and querying a central server are two solutions to the isolation problem described above; however, the first is expensive and the second violates the self-configuring tenet of the peer-to-peer structure.

A new node *X* has the following options that solve the isolation problem:

1. Connect to a special bootstrapping node present within each network domain
2. Connect to a peer known to *X* that it knows – a friend / colleague.

For a novice/new node, the first option may be the most appropriate. As *X* ages, it finds other nodes and adds these links to improve search speed and information access. The linkages are bi-directional and similar to friendships in real life, or to http links in the Web. They are directed by humans.

2.4. Link Weights

Peer links are used to compute *Link Weights* at each peer in the network. As mentioned earlier, this value is very important to help determine the membership of a peer in a community and rank peers in a community for the purposes of information dissemination and trust.

Figure 3. The graph above was plotted using a log Y-axis. It shows how the number of peers that can be reached increases with greater depth. The experiment involved a 10,000 node P2P network. (*See Appendix for technique used to form P2P networks*)

Below, we provide a definition for Link Weights followed by an illustrative example in Fig. 2. In subsequent sections we explain how this value is used in our algorithms.

LINK WEIGHTS: *This is the weight calculated for each claimed attribute of a peer V based on the number of links from V that can reach, after at most one indirection, other peers that claim the same attribute.*

The constraint of at most one indirection is necessary to restrict the maximum depth up to which peers will be examined since more than two levels deep resulted in an unacceptably high number of communication messages. See Fig. 3 for the average number of peers that are reachable from a peer.

3. P2P Community Trust Model

We propose an optimistic trust model that provides probabilistic guarantees based on the status / popularity of the peers. Peers have the ability to revoke their relationships with malicious peers and thus cause the trust values of wrong-doers to be reduced. The probabilistic guarantee provides a web-of-trust style estimate based on a peer's past transactions. The accuracy of the guarantee depends on the thoroughness of the peer in discovering and validating the trust values of other peers. Therefore, non-critical transactions need not consume the resources of the P2P system.

In this section we describe our model for trust using P2P Communities. We explain how trust can be assigned and discovered. The following sections discuss how trust can be revoked, and protected against non-repudiation.

3.1. Peer Roles and Involvement

We previously pointed out that P2P communities are implicitly formed, self-organizing structures that depend on the declared (claimed) interests of peers. As a result, peers may belong to more than one, possibly overlapping community. In the case of a constrained application, such as a digital library, community structures will span across departmental or organizational boundaries. For instance, if the digital library were implemented by government departments to share documents and resources, a conceivable community might include peers from both, the Department of Commerce (Maritime Administration) and the Department of Environmental Resources that are concurrently interested in pollution in US ocean water-ways. This is an example of a cross-departmental community. Peers might also be part of intra-departmental communities, such as the community of Maritime Administration, or the community of Transportation within the Department of Commerce.

The different communities within which a peer can participate due to its claimed interest attributes constitute the roles of the peer. Every peer will have at least one role corresponding to its pre-determined group. Link Weights, by definition, indicate the number of peers known directly (1-hop neighbors), or indirectly (2-hop neighbors) to a peer within each of its roles (communities). Below we provide a definition for *involvement*, which, like Link Weights, is associated with each role Ψ of a peer V and is proportional to the number of peers within the neighborhood (1-hop and 2-hop neighbors) of V that are also part of Ψ. We call peers with high values of involvement, *seers* (See [10] for definition).

INVOLVEMENT: *The average of link weights for elements of the intersection set $C_i \cap S$ is directly proportional to the involvement of node i which has the claimed attribute set C_i in a peer-to-peer community with signature S.*

If peer V from Fig. 2 is a member of community of Science Fiction enthusiasts that has a community signature of {“Technical”, “Fiction”}, then the involvement value of V in this community will be directly proportional 12, which is the average of the individual Link Weights for the claimed attributes, “Technical” and “Fiction”.

For the purposes of simplicity, the examples discussed in this paper consider P2P communities each formed due to single shared interest attributes. This means that the signature S of every community will be a single attribute set. Therefore, the intersection set $C_i \cap S$ can only contain one claimed attribute which has an associated Link Weight that is also the Involvement value of the peer in the community S. Nevertheless, our definition for Involvement provides a way to extract values in more complex scenarios where communities of peers share more than just a single interest attribute in common.

3.2. Trust, Links and Link Weights

3.2.1. First Attempt: Trust and Links. We initially associated trust values with peer links due to the following reasons: (1) Peers create and maintain links to other peers whom they know and therefore trust (optimistically); (2) Since links are bi-directional, information provided by peers that have more links might be more trustworthy. The association of trustworthiness of information (authoritativeness) with links is used by Google in its PageRank metric [13]. The PageRank of a web page measures the authoritativeness of its content; (3) Peer links offer a simple, natural trust model that can easily be revoked. If after some transaction, a peer loses trust in its neighbor, it can break (remove) that link, thereby reducing the number of links at its neighbor.

There are strong arguments that can be made against the use of popularity as a measure of trust, quality or reliability. However, there are specific applications in which this mapping would not be unusual. Examples of analogous systems with a similar association between trust and links include: citation graphs in scientific publications, where experts who are well-known and highly regarded by most other authors tend to be highly connected nodes [11, 14, 15]; and eBay points, where the rank of users is proportional to the number of transactions (purchase / sale) that they have completed with other eBay users.

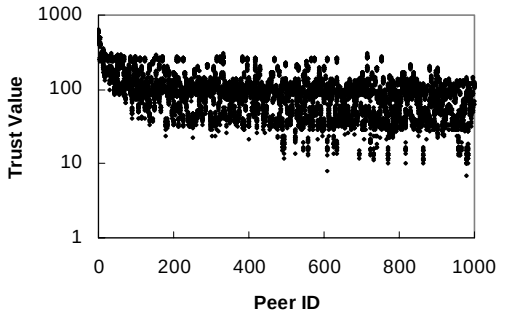
Despite its wide-spread use, the association of trust with peer links does not provide an elegant solution to trust management. Often, peers that are highly linked-to (hubs) make mistakes, provide incorrect information, or assist in spreading damaging information unintentionally. [16] argues that viruses or damaging information from hubs can epidemically spread and persist within a scale-free network, such as P2P network.

We believe that by making a slight modification, links can provide practical and accurate trust guarantees in decentralized systems. The most important detail that has been lacking in previous trust models is the consideration that peers participate in many different communities (roles). Therefore, in the citation graph, although an author of papers in Biology is highly cited, it is conceivable that the author’s explanations of Electrical Engineering concepts are incorrect. Likewise, on eBay, a popular antique seller is not necessarily a trusted expert on electronic equipment.

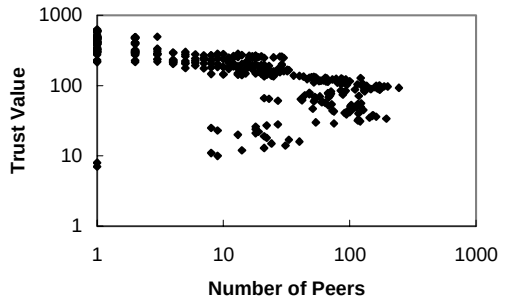
3.2.2. Second Attempt: Trust and Links Weights. It is necessary to consider the roles of a peer when deriving its trust value. We thus propose the use of Link Weights as

an indication of role-based trust. With reference to Fig. 2, peer *V* knows more peers within the community of peers interested in "Biography", than it knows within its other communities. As a result, information provided by *V* and classified as "Biography" is more likely to be accurate than information provided by *V* and classified as "Magazine".

Let us imagine another peer named *W* that has a Link Weight of 10 associated with the interest "Biography." Using our model, *V* would have a trust value of 10 for *W*, but *W* would have a higher trust value of 23 for *V*. The association of Link Weights with trust values allows for asymmetric trust relationships that imitates trust relationships amongst humans in a social network.



(a) Trust Values of all peers (see median band)



(b) Frequency of peers with different trust values

Figure 4. Trust Value Distribution when trust is associated with Link Weights. (1000 peers)

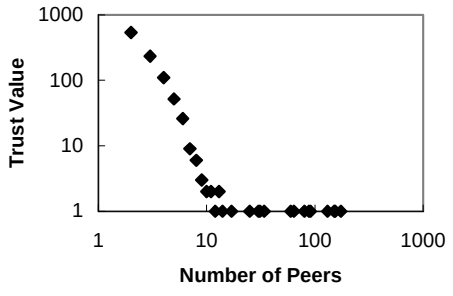


Figure 5. Trust Value Distribution when trust is associated with links. (1000 peers)

3.3. Trust Value Distribution

We plot the distribution of trust values in a simulated P2P network containing 1000 nodes. The graph in Fig. 4(b) shows a non-scale-free distribution of trust values, i.e. a large number of peers had trust values around 100. On the contrary, Fig. 5 shows a power law decay, indicating that almost all peers have low trust values except for a small group of peers that have exceptionally high trust values This is important because it highlights the dissimilarity in the distribution of trust values when it was obtained from links (Fig. 5) and our model (Fig. 4), where trust values are obtained from Link Weights.

In order to correctly model the P2P network, we ensured that the peer link topology was scale-free (See Appendix for technique used to form P2P networks).

Fig. 4(a) illustrates that the majority of peers start out with a median trust value (around the center of a range of values), while a small group of peers have either higher or lower trust values. In the network considered, the average trust value was 100, maximum was 628, minimum was 7, and mode and median were 93. This distribution of trust values is most suitable for an optimistic trust model such as ours because a peer can enter into transactions with other peers whose trust values are median and most likely comparable to its own. Malicious peers will find their trust values dropping unlike in a scale-free distribution where trust values usually cannot be lowered because most peers start out with low trust values. For critical transactions, information can be sought from peers with higher trust values.

3.3. Verification and Validation of Trust Values

In [8] we proposed an *Attribute Escalation* algorithm that uncovered implicit communities and enabled the formation of new communities. The algorithm is an autonomous procedure that is asynchronously executed by each peer. To execute the algorithm, peer *V* sends all peers within its neighborhood (through message forwarding by its neighbors) a list of its claimed attributes. In turn, *V* receives the lists of claimed attributes from its neighborhood peers. Through this simple exchange, we demonstrated how communities were formed. The calculation of Link Weights corresponding to each claimed attribute follows the algorithm.

We propose a simple modification to the Attribute Escalation algorithm that will allow trust values of a peer to be guaranteed. Instead of simply sending out the list of claimed attributes, each peer *V* will construct the

following message *M* and send it individually to each of its neighborhood peers $\{V_1, V_2, \ldots, V_n\}$.

$$M = \left\{IP_{dest}, IP_{source}, \left\langle CA_{source} \right\rangle\right\}, E_{source}(M)$$

where,

M is the constructed message,

IP_{dest} is the destination identity (a neighborhood peer),

IP_{source} is the sender identity (i.e. *V*),

$\left\langle CA_{source} \right\rangle$ is the claimed attributes list of the sender,

$E_{source}(M)$ is the *M*'s signature by the sender's private key.

Every peer is responsible for storing messages received from its neighborhood peers in a publicly accessible blackboard [10]. Blackboards are like websites and the content on a peer's blackboard can be viewed by any peer within the system.

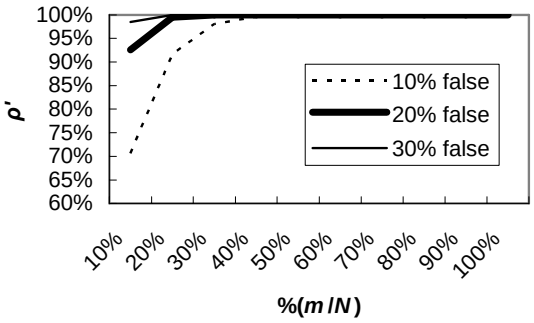


Figure 6. The graph illustrates the relationship between percentage of messages chosen to validate (%*m*/*N*) and probability of uncovering false messages (*ρ'*)

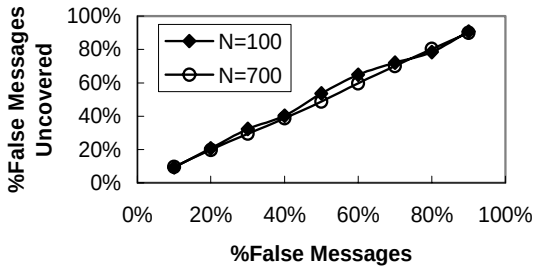


Figure 7. Plot of percentage false messages uncovered (*Y*-axis) as percentage of false messages increased (*X-axis*). %(*m*/*N*) = 10%

Let us return to the example in Fig. 2. When *V* claims a Link Weight (and therefore trust value) of 23 for "Biography," any peer *W* in the P2P system will be able to *verify* this value by visiting *V*'s blackboard and re-calculating the Link Weight from the posted messages. This calculation is a simple counting operation with a complexity of *O(n)* (See [8] for detailed algorithm). Prior to verifying the Link Weight however, *W* might chose to *validate* the signatures of the messages posted on *V*'s blackboard in an attempt to uncover fabricated messages that were used to artificially increase *V*'s Link Weight value. We call these fabricated messages *false messages.*

It might seem intuitive that before entering into a critical transaction with peer *V*, an exhaustive process needs to be employed where every one of *V*'s messages has its signature validated. However, we show that contrary to intuition, peers need only validate a small percentage of messages to uncover one or more false messages (if they exist) with a high degree of probability.

We begin by providing a brief theoretical analysis and back it with results obtained from experiments.

Let *N* be the number of messages on peer *V*'s blackboard. Assume *k* messages are false. Therefore, *N-k* messages are not false. Also assume that peer *W* randomly selects *m* messages to validate. Now the probability that *W* will not discover any false messages is given by:

$$\rho = \frac{N-k}{N} \times \frac{N-k-1}{N-1} \times \frac{N-k-2}{N-2} \times \Lambda \times \frac{N-k-m}{N-m}$$

So the probability that *W* will discover a false message is:

$$\rho' = 1 - \rho$$

Fig. 6 plots the relation between percentage messages verified and ρ'. The curves vary for k =10% of N, k =20% of N, and k =30% of N. N was chosen to be 100. An increasing percentage of m/N forms the X-axis, while ρ' forms the Y-axis. The graph shows that selecting just 10-20% of the messages to validate will uncover false messages with probability of 70-95%. If a peer fabricates more messages, then the validation of messages will quickly uncover false messages.

The theoretical analysis indicates that peers will uncover false messages even when a small, randomly selected set of messages is validated. We also observed this behavior in our experiments. Fig. 7 presents two cases: N = 100 messages, and N = 700 messages, when the percentage of messages randomly selected for validation (%m/N) was set at only 10%. The graph shows that even when only 10% of the messages are false, 10% of those false messages were uncovered by a peer that randomly selected 10% of the original number of messages to validate.

4. Using the Trust Model in Dynamic Coalitions

Dynamic Coalitions are temporarily formed between peers belonging to different communities that each represents a separate organization / department. The trust model we proposed can be used to provide probabilistic trust guarantees to each peer in the coalition.

```
bool CommonCommunities(int);
int[] ListTrusts(int);
bool AskNeighbors(int);
int[] Lower(int[]);
bool Ask2HopNeighbors(int);
bool VerifiedTrusts(int[]);

int[] FindTrust(int PeerID)
begin

   int[] list_Trusts; // trust values

   if (CommonCommunities(PeerID))
   begin-if
      list_Trusts=ListTrusts(PeerID);
   else-if (AskNeighbors(PeerID))
      list_Trusts=ListTrusts(PeerID);
      list_Trusts=Lower(list_Trusts);
   else-if (Ask2HopNeighbors(PeerID))
      list_Trusts=listTrusts(PeerID);
      list_Trusts=Lower(list_Trusts);
      list_Trusts=Lower(list_Trusts);
   else-if
      WARNING "No trust values!";
      return NULL;
   end-if

   if (VerifiedTrusts(list_Trusts))
   begin-if
      return list_Trusts;
   else
      WARNING "Found False Messages";
      return list_Trusts;
   end-if
end-proc
```

Figure 8. Pseudo-code for finding trust values of a peer in a coalition

Fig. 8 lists the algorithm that we use to obtain trust values of a peer in a Dynamic Coalition. Since peers can belong to more than one community, the *FindTrust* method finds all the trust values (Link Weights) of a peer *V*. The method can be invoked by any peer *W* (not necessarily part of the Coalition). Initially, the *CommonCommunities* method checks *V*'s claimed attributes (posted as messages on its blackboard) for any common attributes between *W* and *V*, indicating possibly shared communities. This is a linear search operation with complexity of *O(n)*. If there are no common attributes, *W* asks all its immediate neighbors if any of them share communities with *V*. In the worst case scenario (neighbors need to execute *CommonCommunities*), the operation has $O(n^2)$ complexity. The best case scenario (neighbors already know common communities from past transactions) is an *O(n)* operation. As a final attempt, if still no common attributes exist, *W* asks its 2-hop neighbors the same question (worst case: $O(n^2)$, best case: *O(n)*) before giving up trying to find trust values for *V*. Remember that after the attribute escalation algorithm, a peer knows the identities of all its 2-hop neighbors and therefore does not have to find their identities at this stage. In order to reduce bandwidth utilization and processing time, a peer might decide to forego finding trust values from its 2-hop neighbors. Our experiments revealed that 2-hop neighbors need to be consulted 32% of the time when 10% of randomly selected peers invoked *FindTrust*.

For each attribute found in common with *V*, the corresponding Link Weight is stored in *list_Trusts*. Link Weights provided by 1-hop neighbors will be multiplied by 0.5 and values provided by 2-hop neighbors get multiplied by 0.25. All trust values are validated using the process described in the earlier section.

The list of trust values provides a peer in a coalition with a probabilistic trust guarantee about another peer. Tampered values do not go undetected (due to verification and validation), making these values secure. Additionally, trust values can be transitively obtained from other peers and scaled down depending on the peer providing the values.

As an alternative to our current scaling down process, trust values transitively obtained from another peer could be multiplied by the trust value of that peer. However, we have not yet explained how a peer can have a single trust value. In the next section, we will present our idea for a collective trust value of a peer that can be used for a more realistic scaling process amongst other benefits.

4.1 Aggregating Trust Values into an *iComplex*

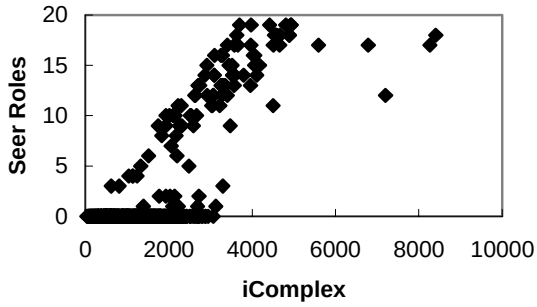
The list of trust values associated with each peer can be used to provide probabilistic guarantees to other peers. However in a practical implementation of P2P communities, a single shared interest attribute will not always be the signature of a community. At the end of section 3.1 we assumed that every community signature contained only a single attribute. Let us see what happens when this simplification assumption were temporarily removed.

Firstly this means that $|S|$ could be greater than one. Imagine a digital library with several communities of peers. Suppose there exists a community (of science-fiction enthusiasts)[2] with signature S_1 = {"Fiction", "Technical"}, and another community (of fiction enthusiasts) with signature S_2 = {"Fiction"}. Finally assume that Peer *V* (from Fig. 2) is a member of both these overlapping communities. Based on the Link Weight values from Fig. 2 and the definition of Involvement (Section 3.1), *V* is more involved in the community of fiction enthusiasts than in the former community. If there

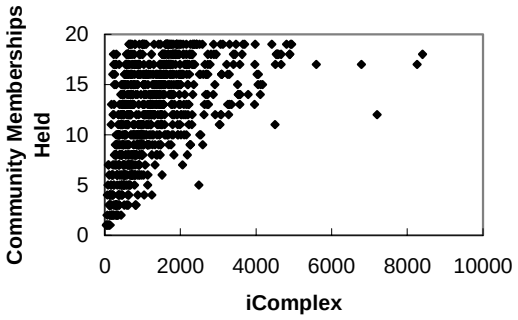
[2] Assigning community names can be done through various consensus and election protocols that are outside the scope of this paper.

exists another peer *W* with Link Weight values, "Fiction"=18 and "Technical"=18 (therefore *W's* Involvement in community S_1 is 18), then information provided by *W* and classified as "science-fiction" is more likely to be accurate than information provided by *V* having the same classification.

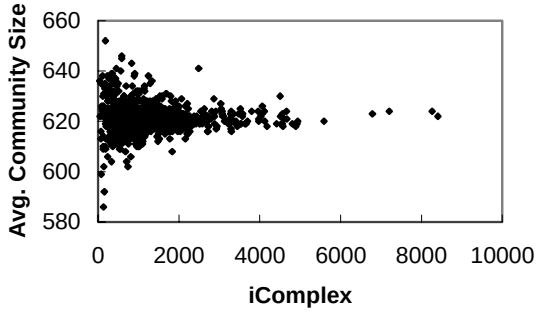
Involvement values, which are associated with each



(a) iComplex Vs. Number of communities in which peer is a seer



(b) iComplex Vs. Community memberships held



(c) iComplex Vs. Average Community Size

Figure 8. The above graphs show the behavior of iComplex when calculated as a sum of all trust values.

community within which a peer is a member, play an important role in determining accurate role-based trust values of a peer. We now propose aggregating all involvement values corresponding to a peer into an *iComplex*. An iComplex value is calculated by each peer individually and stored on their own blackboards. Since iComplex values are, in essence, aggregations of trust values, the verification and validation process described earlier will still apply. As a system design, all peers need to agree upon the aggregation function to calculate their iComplex values. Examples include but are not restricted to: sum of all trust values, or average of all relative trust values. A relative trust value $rt_1 = t_1$ divided by the approximate size of community $S_1=\{C_1\}$, where t_1 is a trust value for attribute C_1, and the approximate size of the community is obtained through Distributed Discovery [9, 10].

In Fig. 8 we show the behavior when sum is the aggregate function used to compute iComplex values. The first graph illustrates that higher iComplex values implies that the peers are seers (highly-involved) in more communities and therefore these seers have higher trust values than other peers in each of their roles. The next graph shows the relationship between iComplex and number of memberships held by a peer. This graph is significant because it demonstrates the effectiveness of an iComplex value aggregated using a simple sum function. The graph shows that peers cannot increase their iComplex values by simply joining many communities. In fact peers that are members of many communities are most likely to have low iComplex values (notice the clustering of points close to the Y-axis). The final graph shows that peers cannot obtain high iComplex values by joining very large communities. The peers with the highest iComplex values were members of average sized communities.

Therefore, iComplex values calculated by adding all trust values of a peer can provide a reliable, collective trust value. Moreover, peers will not be able to synthetically increase their iComplex value by simply joining more communities or joining larger communities. A higher iComplex usually indicates that the peer is a seer of many communities and therefore trusted by the peers of those communities.

4.2. Using iComplex for Information Assurance

The community-based organization of peers enables a more efficient searching mechanism [10] that works by targeting one or more communities, irrespective of the current membership of the searching peer. Any peer that needs to search the P2P network constructs a three-part search query containing: (1) the identity of the peer creating the query, (2) the actual query for an item, and (3) a list of meta-information that describe the item.

In an interest-based P2P network, such as our digital library, a peer might use interest attributes as meta-information to a query. For instance, if the query is for

"books about Vampires," the list of meta-information might include attributes such as, "Twentieth century," "Bram Stoker," and "European authors." Responses to a query are received asynchronously by the searching peer. A peer will respond to a query if it either owns the requested books or can provide information about the peer that owns the requested books.

The response to a query contains: (1) the identity of the responding peer, (2) the query for which the response is being sent, and (3) a list of Link Weight values corresponding to as many meta-information attributes that match the claimed attributes of the responding peer.

An incremental change to the format of the responses can be made by requiring the responding peer to send its iComplex value as well. This provides the querying peer with information on the probabilistic trust values of the responding peers. The iComplex values received will allow a querying peer to rank responses based on the probabilistic trust values of the responding peers.

4.3. Attacks and Threat Assessment

Without the iComplex value, malicious peers could misinform a querying peer about the peer that owns a particular book / resource. A misinformed querying peer will then obtain incorrect / damaging data from the peer identified by malicious peers. In a business-world implementation of digital libraries, malicious peers might dishonestly divert traffic away from certain other peers.

Since iComplex values are ultimately calculated from Link Weights which are dependent on the number of peers in one's neighborhood that share a certain attribute, one way of fraudulently increasing the iComplex value would be to create dummy neighbors with real peer identities and interests. This is a difficult problem to solve. There have been a few attempts to solve this by: using reputation-based systems or making it difficult to create a new peer identity [17] (by computationally expensive key generation, or associating it with a government issued identity number, such as social security number, voter identification number, and so on).

Finally, because our trust model provides probabilistic guarantees, a peer with a high iComplex value can still provide (with low probability of doing so) incorrect / damaging information as a result of a query. We therefore propose a revocation mechanism (described in the next section) as a means to punish wrong-doers.

5. Revocation and Non-Repudiation of Trust

In this section we discuss how our trust model allows peers to revoke relationships with malicious peers, and the non-repudiation of peer relations. Malicious peers are not only peers that provide incorrect/damaging information, but also are peers that use unfair methods to lower the trust values of their neighbors.

5.1. Revocation

We propose a distributed revocation mechanism, where each peer maintains its own revocation list. Therefore a disgruntled peer *W* that has been affected by previous transactions with a malicious peer *V* can simply maintain this information in a revocation list posted on its blackboard.

The validation procedure described earlier involves validating the signatures of the messages posted on a peer *V*'s blackboard in an attempt to uncover false messages. In order to allow for revocation of these messages, we propose an additional action that peers entering into a transaction with *V* can execute after the validation procedure. We call this action *Revocation Check*. The action entails: (1) randomly selecting a few validated messages from *V*'s blackboard; (2) determining the peers that authored those messages; and (3) visiting the blackboards of the message authors to check for possible revocations. If the Revocation Check finds that the message authors have placed *V* in their revocation lists, then those messages are called *revoked messages*.

Section 3.3 explains the relationship between the number of messages validated and the number of false messages uncovered. We therefore know that if 10% of the messages of a peer *V* were selected for Revocation Check and 10% of *V*'s neighbors had placed *V* in their revocation lists, then the Revocation Check will find that 10% of the selected messages are revoked messages.

As a result, if *V* has maintained a good record over a large number of transactions, except for a few incorrect/damaging transactions, then its trust value will remain high. Also, a malicious neighbor of *V* would not be able to independently bring down the *V*'s trust value.

Finally, the Revocation Check procedure can ascertain if a malicious neighbor *W* of *V* has unfairly revoked its relationship with *V*. This means that *W* continues to account *V*'s signed messages to calculate its trust values and iComplex value even after placing *V* in its revocation list.

5.2. Non-Repudiation

We have shown how peers can revoke their relationships with malicious peers to punish them for false or damaging information. However, since trust values are derived from peer links, it is essential to ensure that peers cannot falsely deny their relationship with another peer.

A malicious neighbor *W* of *V* cannot lie about the fact that it is a neighbor of *V*. This is because the signed message (section 3.3) addressed to *V* and created by *W*

will be publicly accessible from *V*'s blackboard. Therefore trust values calculated as a result of peer links have non-repudiation.

6. Related Work

A considerable amount of research has focused on the analysis of link structures in collections of objects. Through these analyses, researchers had hoped to discover a process that could be implemented to effectively identify and discover specific patterns in the collection. Early attempts to analyze the collective properties of interacting agents have been found in social networks [18], where link structures like cliques, centroids and diameters were studied. The field of citation analysis [19] and bibliometrics [20] seek to identify patterns in collections of literature documents by using citation links.

Patterns in several complex systems have been found to be self-organizing [21], often because of the autonomous creation of links by participating nodes (with some influence of a partial system view). Previously, such types of systems had been described by Erdös and Rényi [22] who modeled complex systems as random networks and studied their properties. Watts and Strogatz [23] later studied the properties of large regularly connected graphs of nodes that contain a few random long-distance edges between nodes. They modeled this structure and demonstrated that the path-length between any two nodes of the graph is in fact surprisingly small. As a result, they called these semi-random structures small-world networks. More recently, Strogatz [24] and Amaral et al. [25] observed that many networks demonstrated topological properties that were different from the predictions made by random network theory. Specifically in these networks, called "scale-free networks", the degree distribution of participating nodes was found to decay as a power law.

Perhaps one of the earliest formalizations of trust in computing systems was done by Marsh [26]. He attempted to integrate the various facets of trust from the disciplines of economics, psychology, philosophy and sociology. Rahman and Hailes [27] proposed a trust model based on the work done by Marsh but specifically for online virtual communities where every agent maintained a large data structure representing a version of global knowledge about the entire network. Gil and Ratnakar [28] describe a feedback mechanism that assigns credibility and reliability values to sources based on averages of feedback received from individual users.

More along the lines of trust and social networks, Golbeck, Hendler and Parsia [29] presented an approach to integrate social network analysis and the semantic web. Yu and Singh [30] introduced a referral graph comprising agents as weighted nodes and referrals as weighted edges between participating agents. The graph topology can be changed over time, for instance after bad experiences agents can change their list of neighbors and also propagate information about the "bad" agent within the network. Yolum and Singh [31] propose a similar approach that enables the study of the emergence of authorities in self-organizing referral networks. Pujol et al. [32] associate reputation of an agent with its degree in a social network graph. Similar to PageRank in Google, an agent gets a high reputation if it is pointed to by other agents that also have high reputation. Aberer and Despotovic [33] analyze earlier transactions of agents and derive from that the reputation of an agent. Reputation provides a value that indicates the probability that the agent will cheat. They also presented a design for trust management using their proprietary decentralized storage method.

Our work introduces a novel role-based trust model and discusses its use within dynamic coalitions of peers. We associate trust values with Link Weights instead of links and finally propose an aggregation of different trust values of a peer into a single probabilistic trust value. Our algorithms are completely decentralized and the trust values are secure and can be thoroughly validated and verified without a high communication overhead.

7. Conclusion

Without a viable trust model, information sharing in P2P systems will be susceptible to the spreading of viruses, and incorrect or damaging information. In this paper, we consider a P2P system containing self-organizing, overlapping, interest-based communities that can be uncovered using decentralized techniques. We relate peer communities to Dynamic Coalitions where coalitions are created between peers in different communities. The communities within which a peer can participate due to its claimed interests constitute the roles of the peer. Every peer will have at least one role corresponding to its pre-determined group. Each claimed interest of a peer is associated with a Link Weight that indicates the number of neighboring peers (1-hop or 2-hop) sharing the same interest. The computation of Link Weights has been described in an earlier work and involves simple message exchanges amongst peers in a process known as *Attribute Escalation*. In this paper, we discussed how these messages can be protected against tampering and counterfeiting.

Our proposed trust model is optimistic in that the majority of peers start with a median trust value, while a small group of peers have either higher or lower trust values. We proposed the use of *Link Weights* as an indication of role-based trust and provided the definition of *Involvement* to extract values in complex scenarios

where communities of peers shared more than just a single interest in common. Trust values are a probabilistic guarantee similar to web-of-trust style estimates and is based on a peer's past transactions. We proposed a simple modification to the Attribute Escalation algorithm that allowed trust values of a peer to be verified and validated by any other peer in the network. The theoretical analysis of the validation process indicated that selecting just 10-20% of messages to validate uncovered false messages with probability of 70-95%. If a peer fabricated more messages, then the validation of messages quickly uncovered false messages. Similar results were also obtained experimentally.

The trust model we proposed can also provide probabilistic trust guarantees to each peer in the coalition. We provided an algorithm to obtain trust values of a peer in a Dynamic Coalition transitively. Involvement values, which are associated with each community within which a peer is a member, play an important role in determining accurate role-based trust values of a peer. We proposed aggregating all involvement values corresponding to a peer into an *iComplex*. An *iComplex* value is calculated by each peer individually and can be guaranteed by using the verification and validation process. We demonstrated experimentally that *iComplex* values calculated by adding all trust values of a peer can provide a reliable, collective trust value. We discussed the use of *iComplex* values in information assurance and also delved into the subject of attacks and known threats of our trust model.

Finally we explained how our trust model allows peers to revoke relationships with malicious peers, and the non-repudiation of peer relations. Malicious peers are not only peers that provide incorrect/damaging information, but also are peers that use unfair methods to lower the trust values of their neighbors.

8. References

[1] Napster. http://www.napster.com

[2] Gnutella. http://www.gnutelliums.com

[3] I. Clark, O. Sandberg, B. Wiley, and T. Hong, "Freenet: A distributed anonymous information storage and retrieval system", *Proc. of the Work. on Design Issues in Anonymity and Unobservability*, Berkeley, CA, 2000.

[4] R. Dingledine, M. Freedman, and D. Molnar, "The freehaven project: Distributed anonymous storage service", *Proc. of the Work. on Design Issues in Anonymity and Unobservability*, 2000.

[5] D.F. Ferraiolo and D.R. Kuhn, "Role Based Access Control", *15th National Computer Security Conference*, 1992.

[6] P. Dasgupta, V. Karamcheti, and Z. Kedem, "Efficient and secure information sharing in distributed, collaborative environments", *Proc. of 3rd Intl. Work. on Communication-based Systems*, April 2000.

[7] Yahoo Groups. http://groups.yahoo.com

[8] M. Khambatti, K. Ryu, P. Dasgupta, "Efficient Discovery of Implicitly formed Peer-to-Peer Communities", *Int'l Jour. of Parallel and Distributed Systems and Networks*, vol. 5, no 4. 2002, pp. 155-164.

[9] M.S. Khambatti, K.D. Ryu and P. Dasgupta, "Push-Pull Gossiping for Information Sharing in Peer-to-Peer Communities", *Int'l Conf. on Parallel and Distributed Processing Techniques and Applications*, Las Vegas, NV, June 2003.

[10] M. Khambatti, K. Ryu and P. Dasgupta, "Structuring Peer-to-Peer Networks using Interest-Based Communities", *Int'l Work. On Databases, Information Systems and Peer-to-Peer Computing*, Berlin, Germany, September 2003.

[11] J. Kleinberg, "Authoritative sources in a hyper-linked environment", *Proc. of the 9th Annual AVM-SIAM Symp. on Discrete Algorithms*, 1998.

[12] G.W. Flake, S. Lawrence, and C.L. Giles, "Efficient Identification of Web Communities," *Proc. 6th Int'l Conf. Knowledge Discovery and Data Mining*, ACM Press, New York, 2000.

[13] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine", *Computer Networks and ISDN Systems*, 30(1–7), 1998, pp. 107–117.

[14] L. Adamic and E. Adar, "Friends and neighbors on the web", (unpublished), 2000.

[15] L. Page, S. Brin, R. Motwani and T. Winograd, "The PageRank citation ranking: Bringing order to the Web", *Stanford Digital Libraries Working Paper*, 1998.

[16] R. Pastor-Satorras and A. Vespignani, "Epidemic Spreading in Scale-Free Networks", *Phy. Rev. Letters*, vol. 86, no. 14, April 2001, pp. 3200-3203.

[17] T. Murphy VII and A.K. Manjhi, "Anonymous Identity and Trust for Peer-to-Peer Networks", (unpublished), 2002.

[18] J. Scott, *Social network analysis: a handbook*, SAGE Publications, 1991.

[19] E. Garfield, *Citation Indexing: Its Theory and Application in Science*, Wiley, New York, 1979.

[20] H.D. White and K.W. McCain, "Bibliometrics", *Ann. Rev. Information Science and Technology*, Elsevier, 1989, pp. 119-186.

[21] R. Axelrod and M.D. Cohen, "Harnessing Complexity: Organizational Implications of a Scientific Frontier", *Basic Books*, New York, NY, 2000.

[22] P. Erdös and A. Rényi, "On the Strentgh of Connectedness of a Random Graph", *Acta Math. Acad. Sci.*, Hungary, 12, 1961, pp. 261-267.
[23] D. Watts and S. Strogatz, "Collective Dynamics of "Small-World" Networks", *Canadian Jour. Math*, vol. 8, no. 3, 1956, pp. 399-404.
[24] S.H. Strogatz. "Exploring complex networks," *Nature*, London, 410, 268, 2001.
[25] L.A.N. Amaral, A. Scala, M. Barthélémy, and H.E. Stanley, "Classes of small-world networks," *Proc. Natl. Acad. Sci. U.S.A. 97*, 2000, pp. 11149-11152.
[26] S.Marsh, "Formalising Trust as a Computational Concept", *Ph.D. Thesis*, University of Stirling, 1994.
[27] A. Abdul-Rahman and S. Hailes, "Supporting Trust in Virtual Communities", *Proc. of the 33rd Hawaii Int'l Conf. on System Sciences*, 2000.
[28] Y. Gil and V. Ratnakar, "Trusting Information Sources One Citizen at a Time", *Proc. of the 1st Int'l Semantic Web Conf.*, Sardinia, Italy, June 2002.
[29] J. Golbeck, B. Parsia, and J. Hendler, "Trust networks on the semantic web", *Proc. of Cooperative Intelligent Agents 2003*, Helsinki, Finland, August 2003.
[30] B. Yu and M.P. Singh, "A Social Mechanism of Reputation Management in Electronic Communities", *Proc. of the 4th Int'l Work. on Cooperative Information Agents*, M. Klusch, L. Kerschberg (Eds.), *Lecture Notes in Computer Science*, vol. 1860, Springer, 2000.
[31] P. Yolum and M.P. Singh, "Emergent properties of referral systems", *Proc. of the 2nd Int'l Joint Conf. on Autonomous Agents and MultiAgent Systems*, ACM Press, July 2003.
[32] J.M. Pujol, R. Sangüesa, and J. Delgado, "Extracting reputation in multi agent systems by means of social network topology", *Proc. of the 1st Int'l Joint Conf. on Autonomous Agents and Multiagent Systems*, pp. 467–474, ACM Press, July 2002.
[33] K. Aberer and Z. Despotovic, "Managing trust in a peer-2-peer information system", *Proc. of the 10th Int'l Conf. on Information and Knowledge Management*, H. Paques and L. Liu and D. Grossman (Eds.), ACM Press, 2001, pp. 310-317
[34] T. Bu and D. Towsley, "On distinguishing between internet power law topology generators," *Proc. of INFOCOM*, 2002.
[35] R. Albert, H. Jeong, and A. Barabasi, "Diameter of world-wide web," *Nature*, vol. 410, Sept. 1999, pp. 130-131.
[36] M. Granovetter. "Strength of weak ties," *American Jour. of Sociology*, vol. 78, 1973, pp. 1360-1380.
[37] H. Jeong, Z. Néda and A.-L. Barabási, "Measuring preferential attachment for evolving networks", *Euro. Phys. Lett. 61*, 567, 2003.
[38] M. E. J. Newman. "Clustering and preferential attachment in growing networks," *Phys. Rev. E 64*, 025102, 2001.
[39] A. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, 1999, pp. 509-512.
[40] M. Steyvers and J. B. Tenenbaum, "The large-scale structure of semantic networks: statistical analyses and a model of semantic growth" (submitted to *25th Annual Meeting of the Cognitive Science Society*)

Appendix

In this section, we describe the properties of the resulting P2P network, such as small-world networks, and scale free networks. Finally, we propose a new technique for generating a P2P network for our simulations.

Small-World Networks

Watts and Strogatz [23] have described a special type of semi-random network, called "Small-World Networks," which contain a few randomly re-wired long-distance edges within a regularly connected network of nodes. These networks have low characteristic path lengths (as in random networks) and high clustering coefficients (as in regular networks). Subsequently, a number of papers have acknowledged the existence of small-world networks in the Internet topology [34, 35]; the power grid of the western United States; various social networks [23], such as the collaboration graph of film actors; Erdös numbered research scientists; and even in the neural network of the worm *C. elegans*.

Further, Granovetter [36] discusses the existence and shows the importance of weak social ties (links) between highly connected clusters of friends.

The similarity of P2P networks to social networks and the fact that humans direct peer links led us to believe that P2P networks would also exhibit small-world behavior. In fact, this has already been observed in existing P2P networks, such as Gnutella [2].

Scale-Free Networks

Scale-free networks are characterized by the uneven distribution of connections (links) in the nodes of the network. Unlike a random network that exhibits a Poisson distribution of node degrees, a scale-free network demonstrates a degree distribution that decays as a power law. Hence, scale-free networks have sometimes been described as power-law networks.

During the study on complex networks, [24, 25] observed that many networks demonstrated topological properties that were different from the predictions made by random network theory. In particular, the existence of some very well connected "hub" nodes dramatically influenced the behavior of these scale-free networks during random node failures and spreading of information. It has been shown that various networks, such as the collaboration graphs of actors and scientists, were developed due the feature of preferential attachment [37, 38, 39]. This feature describes the probability of a node acquiring new links as an increasing function of the links that it currently has.

In order to correctly model the P2P network, it is important that we also incorporate the scale-free property into the network topology.

Creating Our Own P2P Network

We needed to provide a mechanism to ensure that our P2P network topology would exhibit the properties of a small-world network and would also show a power-law distribution for frequency vs. degree.

Our next approach [9] involved enforcing certain rules on new peers that wanted to join the P2P system. We were inspired by the work of M. Steyvers and J. Tenenbaum [40] on semantic networks, and extended the domain of their model to a P2P network that involved peers and links.

The new peer, X, has to follow a two-step procedure (described below) in order to join a P2P system.

Let N = {set of known peers} *(See section 2.3.2)*

$\xrightarrow{\wp}$ = "connects (links) to" with probability $\wp$

$\wp_A \propto$ degree of node A (k_A)

$$X \xrightarrow{\wp_A} A, \quad A \in N \qquad \textbf{Step (1)}$$

$$\forall\ B = m'_A \in M'_A, \quad X \xrightarrow{1} B \qquad \textbf{Step (2)}$$

where,

M_A = {set of neighbors of A}, $|M_A| = k_A$

$M'_A \subseteq M_A$ such that $|M'_A| = d$ (globally defined)

$i \in M'_A$ with probability $\wp_i \propto k_i$ *iff* $i \in M_A$