Exploiting Features of Collaborative Social Networks in the Design of P2P Applications

José Mitre, Leandro Navarro-Moldes

Polytechnic University of Catalonia, Spain
{jmitre, leandro}@ac.upc.es

Abstract. This paper shows a model of a P2P network to facilitate collaborative activities, which takes advantage of features of a particular social collaborative network –small-world, clustering, community structure, assortative mixing, preferential attachment and small groups–. We present a description of this model based on collaborative groups and limited flooding. Our simulation results show that our model improves scalability and performance of flooding based searches (search results in *O(1)* hops), message cost, support for node instability, keyword searches, load balancing (number of replicas proportional to demand) in comparison with P2P networks that use DHT or flooding.

1 Introduction

In recent years, many studies have been aimed at finding the properties of social networks, which is a set or group of people, known as the actors, linked together or joined by some pattern of interaction [1]. These efforts arose not only from the interest inherent in patterns of human interaction, but also from the structure of the resulting network. These studies have focused on a number of properties that seem mainly to affect their performance. Among these properties, perhaps those most widely studied are degree distribution [2, 4], the "small world" effect [5-7], clustering [5], community structure [8-10], resilience to the deletion of network nodes [2, 11, 12] and navigability or search ability of networks [13, 14].

Along with other authors [2, 3], we claim that these properties have a great impact on the way that these social networks operate and offer valuable information about the way in which information travels and is routed through the network. Then, this information can be used in the design of networks of computers created by groups of people who work together in order to improve both the performance of the applications and the efficient use of resources.

The model we propose provides the basic mechanisms for modular P2P applications for affiliation collaborative groups of people geographically dispersed, enabling activities such as reading, discussion, writing and modification of documents and diffusion of awareness information, events, news, to be carried out in a distributed way. It is currently being implemented in Java with JXTA [15].

We can define a collaborative social network (CSN) as a social network in which the actors are people collaborating in some activity and the links joining them together denote their collaboration. Affiliation CSN, are a type of social network having certain natural topologic properties that make them quite well structured.

This paper is organized as follows: in Section 2 we describe some of the features of affiliation CSN that have been exploited in the design of our P2P model. In Section 3 we present an analysis of costs generated by searches in P2P networks using DHT, flooding and collaborative groups and how replication helps to improve the performance and availability of resources. In Section 4 we explain our P2P model as well as the mechanisms for peer communication and management. Sections 5 and 6 details our simulation environment and results. In Section 7 we present some applications focused on collaboration, and Section 8 presents our conclusions.

2 Collaborative Social Networks Features

Community Structure: [8-10] is the property of many social networks for forming communities through the union of people in groups. Grouping occurs for many reasons – shared interest, working for the same company, geographical proximity, etc. In many social networks it is possible for people of a similar type to be drawn together and then to divide up naturally into groups, so that the density of links within the group is greater than the density of the links among them [9].

Assortative Mixing: A social network is said to show assortative mixing [12] if in that network the people wishing to associate with others all have something in common.

Preferential Attachment: In the majority of social networks, the addition of new nodes occurs by preferential attachment [8], in such a way that the new nodes are connected to other nodes by preference; to nodes with a greater degree, for example, or to those that are most popular.

Clustering: is the probability of two people meeting if they share one or more mutual acquaintances. For example, in collaborative groups, people tend to introduce their collaborators to others belonging to other groups, thus fomenting new collaboration and thereby increasing the clustering coefficient [8].

Affiliation Networks: an affiliation network [16] is a network in which the actors are joined together by common membership of groups of some kind. Some studies show groups of academics, actors and business people as affiliation networks.

Degree Distribution: In real social networks the degree distribution follows a power law [13], which indicates a heterogeneous topology in which the majority of nodes have a small degree and a small fraction of highly connected nodes.

Small World: CSN form "Small Worlds". Typically, participants are separated by short paths [13] of known intermediates. Clearly, news of important findings can circulate more quickly in a network where nodes are more closely connected.

3 Analysis of cost search and replication

As was showed in the previous section, in CSN communities emerge. The effect is that the probability that objects of interest in a community are within the scope (inside the limits) of that community is very high. In economic terms, we could say that a search without considering communities is more costly, in terms of load (number of messages and peers), and the benefit, chance of and response time to finding an object, may be lower than a search inside a community, where more benefit is obtained at lower cost. Another argument giving additional support to CSN is the fact that people collaborating tend to use the same resources (sharing) during the time span of the collaboration, which is long term.

Let us imagine a P2P computer network used by a collaborative group. Let us suppose that a peer X creates an object A. If all peers, including X, were always connected, the probability that any other peer Y would find the object A is $P(A)=1$, given that we are assuming that A is always available (or X always connected, which is equivalent). But P2P networks, in general, are very instable due to continued node arrivals and departures. Given that peers tend to be end-user machines rather than dedicated servers, there is no guarantee that the peers will not be disconnected from the network at random. This generates low availability of resources and a great overload due to searches for these resources [17]. Thus, the probability of finding the object A decreases with the instability of peers. Now let us suppose we replicate that object A on another random peer. In that case, the availability of A is higher, as the probability of finding the object A or a replica is also higher. Thus, replication helps to cope with peer departures and failures and then to improve resource availability. However it also has a cost [18].

In next section there is an analysis for the cost implied in searching and replicating objects in P2P networks. We have analyzed two search strategies: those that use flooding algorithms (unstructured) and those that use DHT (structured). The two search strategies are studied with and without object replication, and using a structured P2P network based on the CSN topology. The cost figure is defined as a function of the amount data exchanged in messages during the search and the transfer of the target object.

3.1 Search without replication

The cost C_q of a search without replication is defined by the cost c of query messages, routing messages and the transfer of the requested object.

Let us initially assume a P2P network without failures (all nodes connected), where the object of interest is located at only one peer (no replication) which is, in the worst case, located at the far end of the flooding region. The total cost of a query is the sum of all messages generated to find the object, and the actual transfer of the object. This cost depends on the number of messages each peer sends to their neighbors (k), and on the scope of the search in the number of hops (TTL), which is related to the diameter of the network (ϕ).

We use four types of different messages generated by a Gnutella type query (GNet) [19] over a random network: the query message m, the message returned by the peer where the object is located m', the message requesting the transfer of an object m'' and the message containing the object o_i. Therefore:

$$C_q = N_m c(m) + N_{m'} c(m') + N_{m''} c(m'') + c(o) \qquad (1)$$

N_m corresponds to the number of messages of type m generated during the search. We can see that the number of messages and the number of potential responses increases exponentially with each hop.

Structured P2P networks that use Distributed Hash tables provide an important cost reduction in the search and location of objects. The search cost, related to the number of messages, in most DHT is $N_m \sim log_k(N)$, where N is the number of nodes, and k the order of the search tree, usually $k=2$.

	Number of messages N_m	Diameter	Topology
Unstructured search (gnutella)	$N_m = k\sum_{i=0}^{\phi_{GNet}} (k-1)^i$	ϕ_{GNet}	Random
Structured search (DHT)	$N_m = \log_2 N$	ϕ_{DHT}	K-degree tree
Social network based (CSN)	$N_m = k\sum_{i=0}^{\phi_{CSN}} (k-1)^i$	ϕ_{CSN}	Social network (clusters)

The total cost of a query C_q (equation 1) continues to be valid, with a different value of N_m: the search is more directed, and a much lower number of peers are traversed. After the object is located, the request and transfer process occurs as previously, and therefore the rest of the cost function does not change as in (1).

In CSN, given the properties of CSN (Section 2), we can assume the existence of groups with small diameter. Given that these groups are made up of people with common interests, and since the information relevant to a group can usually be found within their group, our hypothesis is that when searching for an object it is highly probable that the object will be found within the group interested in this object. In terms of the diameter of search (scope): $\phi_{CSN} << \phi_{DHT} << \phi_{Gnet}$

Therefore, the search cost will be dramatically reduced from the diameter ϕ of the global random network to the diameter of a smaller cluster structured by social links.

3.2 Search with replication

Replication has two notable effects: increasing resilience to node failures, and reducing the scope of search. Replication also has an overhead cost, but the influence on the cost for each query is small because the cost of replication must be split among all transfers of a given object. This factor is $r/p(o)$, where r is the number of replicas in the P2P network, and $p(o)$ the popularity of the object o, or the number of requests in a given period. Therefore:

$$C_q = N_m c(m) + N_{m'} c(m') + N_{m''} c(m'') + (1 + \frac{r}{p(o)}) c(o) \tag{2}$$

Replication strategies vary in time, number and location of replicas[20]. In Gnutella network replicas may be located randomly, in DHT they are located in precise locations dictated by the DHT graph, roughly at the same average distance. In CSN, replicas are located close to the demand: at one hop in terms of interest for most queries since it is based on the social network topology.

In the following analysis of resilience to node failures, we consider that there may be multiple copies of each data object at distinct peers, but for generality we do not take into account the location of replicas.

The existence of *r* replicas in a P2P network with peers connected with probability p_c increases the chance of a successful query: finding at least one copy of an object. If P_h is the probability of finding at least one replica during a search, then, $p_h \approx 1-(1-p_c^{\alpha})^r$.

If the P2P network is fault-tolerant: it has multiple routes to reach peers with replicas, then p_h only depends on the probability of disconnection of the peer holding a replica, i.e.: $\alpha = 1$.

In P2P networks with search based on minimum spanning trees, a message between two nodes depends on all previous nodes on the search tree ($\alpha=\phi$) where ϕ is the diameter of the network. This value of P_h depends on the complete path to each replica which, in the worst case, it has ϕ peers. Therefore the total cost C of a successful query with *r* replicas is given by:

$$C(\phi) = \frac{1}{1-(1-p_c^{\alpha})^r} C_q \quad \text{where } (\alpha = 1 \text{ or } \phi) \tag{3}$$

Compared to the cost without replication ($r=1$), as *r* increases, C_q increases only very little (at least for popular objects), but as P_h has a potential growth with *r* towards 1, $C(\phi)$ decreases: with more replicas, the object tends to be found in one single query.

As one may readily observe, increasing in the number of replicas also increases the probability of reaching an object (accessibility), but it also increases the storage cost with an increasing number of replicas.

3.3 Discussion

While the search cost in DHT algorithms is in the order of the diameter ϕ of the network, search with flooding algorithms on topologies with *k* neighbors grows in the order of k^{ϕ}. The use of a CSN topology instead of a typical random network is always beneficial, since the object of interest can almost always be found in the proximity and within the same community, with a much smaller diameter. It can be located in fewer hops, thus enabling flooding-based search algorithms can generate much less traffic.

Replication helps to reduce the impact of node disconnection as shown by the term $(1 - P_c^{\alpha})^r$ from equation (3), but it also introduces a new cost with the number of replicas *r* in the network. There is a storage cost when the overall storage space is limited, which has not yet been considered. There is also a transfer cost to create each replica, and the location of the replicas may help to reduce the search cost.

Random replication does not guarantee that a replica is close to (few hops away) from peers who may need it. On-path replication, where an object can be replicated at every peer on the path of a successful query: from the peer with an object to the requesting peer. The disadvantage in this strategy is that the object transfer degrades have to go through several peers instead of a direct connection.

The cost of both replication strategies is significantly reduced if the P2P network used a CSN topology, since using any replication strategy replicas can be located very close to peers with most potential demand. Replication and search can be limited to members of a small community in which the object is located, without the necessity of diffusing the object throughout the entire network. This is because these group members have the greatest interest in the object, since it was generated within their interest group. The maintenance of the CSN topology has an overhead cost, but given that this topology evolves very slowly, we assume the cost is negligible.

Then, combination of replication mechanisms with a CSN topology can assist in appreciably reducing search costs in the network when compared with costs generated by traditional P2P networks, in addition to increasing the probability of access to the required resources. This affects both the performance of the applications and the efficient use of resources. As a result, the search cost is dramatically reduced going from the diameter of the global network to the diameter of a smaller cluster, and replicas are more effective since they are located close to the demand.

4 P2P model for CSN

This section describes our collaborative P2P model. First, we define the components that participate in the model.

Let a collaborative P2P network, which helps the collaborative work between people who might be geographically separated.

A *servent* (server and client) is a computer connected to the collaborative network. Every *servent* holds a list of known groups (GroupId List) and a list of group members identified by their ServentId. These lists may be incomplete, and they are kept consistent using an epidemic consistency algorithm [21] (the details are beyond the scope of this paper). *Servents* provide interfaces by means of which people can exchange messages, share information, carry out searches, compare data, and generally pursue the collaborative work.

A *servent* X is a *neighbor* of Y when X is directly connected to Y by a logical connection using this model, provided that X and Y belong to the same group. This logical connection is created if X and Y are direct collaborators.

A *group* is a sub-network of the network formed by the *servents* associated to people who share interest in common topics. New *servents* joining a group must follow the same rules of behavior as in real life; that is, by affiliation or invitation of *servents* to a given group. Thus the network topology would be similar to the topological structure of the real collaborative social networks (Section 2). Our model assigns a unique GroupId to every group.

A *member* is a *servent* belonging to a given group. All the *servents* must, by default, be *members* of at least one group. Disconnected *servents* will continue to be *members* unless they explicitly withdraw.

Based on the properties set out in Section 2, our model has three fundamental mechanisms for carrying out cooperation functions: 1) connection and join, 2) replication, and 3) search.

4.1 Connection and join mechanism

In many cooperation networks, users connect and disconnect from the network several times a day. It is therefore easy to see that a cooperation network must have mechanisms that manage the connection and disconnection of nodes from the network. First connection to a group is a special case, since then the new node must obtain membership information from the group. From the foregoing and for simplicity, we distinguish two types of connection: joining and connection.

4.1.1 Joining

In order for the topology generated by our model to maintain the same properties as those of a real CSN, *e.g.* Clustering and small-world, the *servents* must have means of connecting to groups that are similar to those used in real life (affiliation network). Therefore when a new *servent* joins a group, it will be by invitation or by application from the new *servent* to the group. For a *servent* to be connected to the network for the first time, the person must either establish contact with an existing group or create a new group. The *servent* provides a suitable interface to carry out both operations.

When a *servent* creates a new group, he must generate a unique GroupId that identifies the group throughout the CSN. The *servent* must have a ServentId that identifies him, adding to the ServentId list of the group and sending a message to a number of *servents* of the other groups using an epidemic dissemination algorithm, to notify them of his existence. They will feed in turn the initiating *servent* with information about the existing groups in the whole network.

If a *servent* wishes to join to one or more existing groups, he must first receive authorization from any *member* of that group and receive the potentially incomplete group's ServentId list. Once the person has chosen the interest group to which he wishes to be connected, the *servent* must send a message to any *member* of that group to apply to such group. If a member accepts the application to join, the ServentId of the new *servent* is added to the ServentId list of each member of the group, using an epidemic algorithm to spread the new ServentId. Once a *servent* becomes a member of one or more groups in the CSN, he can communicate with other members of the group/s and share information.

If someone no longer wishes to belong to a group, he must send a message with his ServentId, via the *servent*, to other (a few neighbors + epidemic propagation) members of the group or groups to which he belongs in order to cancel membership. The other *servents* must delete the ServentId from the group's ServentId list.

4.1.2 Connection

This operation is used for any further connection after joining a group and after having been disconnected for some time. When a *servent* is connecting he must send a message to all his *neighbors* (eventually by epidemic propagation, it will be known by all the members of the group) informing them that a connection has taken place. Once the *neighbors* have received the message, they must all update their local ServentId list. The.connecting *servent* will update his own ServentId list by sending a request to any *neighbor*.

In order to know about potential object changes that may have occurred while he was disconnected, the connecting *servent* must launch a search operation (Section 4.3) for events that might have taken place during his absence.

When a *servent* is instructed to disconnect from the CSN, he immediately informs all (a few neighbors + epidemic propagation) connected members that he is about to leave the network, in order to keep the ServentId list up-to-date. In case of connection failure, if a *servent* sending a message receives no reply from another *servent*, the sender will assume that some fault has occurred in the connection with the recipient, and will then proceed to update his ServentId list, indicating that a *servent* is not connected, or informing other members of the change in the ServentId list by epidemic propagation. In this way the list will eventually be up-to-date.

4.2 Replication Mechanism

Given that groups in CSN are made up of people with common interests, and since the information relevant to that group can usually be found within it, we claim that when searching for an object occurs, it is highly probable that the object could be founded within the group interested in this object, in few hops (small-world). Object replication will improve object availability (p2p networks are very dynamic), increase system resilience even during directed attacks to high degree *servents*, and it will improve the performance of search operations without overloading the network. Replication could be carried out solely for the members of the group where the object originates, not necessarily for *servents* outside the interest group or even the entire network. This is because these group members will have the greatest interest in the object, since it was generated within their group (assortative mixing).

We have seen that the frequency with which objects of interest for a particular group are created and modified is in fact low, and the greater part of communication consists of the exchange of ideas via e-mail or chat which do not need to be replicated. This has been confirmed by the analysis of one year event log for the activity performed by a collaborative group of people using BSCW, an application for collaborative work support. It shows that the number of reading events is several magnitude orders higher than the number of writing or modification events [22].

We now present a way of managing replication in our model: When a member creates a new object or modifies one already existing, he must notify that to the group. Immediately after, the *servent* initiates the mechanism to replicate the object to his *neighbors* in order to reduce the number of replicated objects circulating through the network before arriving at their destinations. For example, let $G=\{A,B,C,D,E,F\}$ where $A,B...F$ are *servents* belonging to group G, and B,C,D are the *neighbors* of *servent* A. When *servent* A creates a new object or when he modifies an object already existing in the network, the given object is replicated only to their *neighbors* B, C and D, since they have higher need of that object than any other *servent*, given that B, C and D directly collaborate with A. This proximity replication criteria guarantees that the immediate collaborators will have a replica of the object of interest (assortative mixing). Given that the number of replicas is directly related to the *servent* degree, high degree *servents* will have more replicas, making the network resistant to failures or directed attacks, and balancing load especially for highly connected *servents*.

In the longer term, considering that *servents* have a limited storage capacity, they will have to apply a replacement policy to make room for new objects of higher interest, but they will still keep meta-information on alternative locations of the object, which is roughly equivalent to having the object (one additional hop; the replication mechanism helps *servents* learn the content or at least the location of objects of interest).

Initial simulation results confirm that the number of replicas of an object grows quickly with the number of related search operations, and with the degree (number of *neighbors*) of the originating node which is correlated with popularity.

4.3 Searching Mechanism

With the aim of reducing to a minimum the number of search messages circulating in the network, our search mechanism makes use of both the CSN capacity for forming small-world communities and the replication mechanism.

We have already mentioned that, because of the proposed replication mechanism, the cost of flooding based searches is drastically reduced.

Our model use two types of flooding search – local and external.

Local Search. Search undertaken by a *servent*, to bring his information up to date or looking for an object within the group. A local search can be made for three reasons: 1) When a new *servent* joins a group and needs to know about all the objects shared by the group. So when a new *servent* receives a message on concluding the initial connection process (Section 3.1), he should ask other *servents* for the objects shared by the group. 2) When a *servent* has reconnected after having been disconnected for a certain time; the *servent* must then seek to update information generated in his group during his absence. He will carry out a local search for new or modified objects. And 3) when a *servent* belonging to a group needs a particular object, he will carry out a local search for that object by sending a query to his *neighbors*. The *members* receiving the query message will send information about the object to the requesting *servent* if they have the target object.

External Search. Search carried out outside the group to which the *servent* initiating the search belongs. This situation may arise when a *servent* needs an object that is not available from any of the group members, and must therefore look for it in other groups. The user must explicitly undertake this kind of search when he or she wishes to search for an object throughout the entire CSN.

To make flooding search more efficient, *servents* have information (GroupId) about each existing group within the network. The *servent* initiating the search can locate at least one *servent* from each group and direct the search towards them.

Given our proposal that a number of the *servents* have object replicas of interest to the group, the probability of locating the desired object will be quite high, and search messages may go directly to those *servents* who are most likely in possession of the object. In terms of external searches, our mechanism differs from flooding algorithms in the number of *servents* involved: .we select at least one *servent* per group while flooding would contact all nodes up to a maximum number of hops (TTL).

Both in local and external search, objects would be downloaded via a direct connection between the *servent* possessing the object and the *servent* requesting it. Since the object is replicated, it can be done in parallel from various *servents* in order to make download process fault tolerant, faster and more efficient.

As may be easily appreciated, our search strategy is thus less costly *~O(1)* (objects located in a single hop in most cases) than classical flooding *~O(N)* and can be more flexible than DHT, typically *~O(log N)*.

5 Evaluation

The analysis described in Section 3 provides some insight into the potential benefits of our model. But in order to test the validity of our proposal and to evaluate the effects of features described in Section 2, we developed a simulation infrastructure implementing our model. We simulate our model using the j-sim simulator [23].

We implement the search and replication mechanisms on the top of collaborative networks. The topologies of these networks are based on a Newman's algorithm [9]. Using this algorithm we have generated networks with properties such as: clustering, community structure and small-world. Different randomly generated topologies are formed by interest-based groups of 10, 50 and 100 *servents*, where each node represents a person, and each link models a personal relationship in the collaborative social network, which corresponds to a link in our P2P network.

For each topology we run 1,000 differently seeded simulations, consisting of *N* requests (one for each *servent*) for a single object created on a random *servent*.

In each simulation cycle, we randomly designate a *servent* to be the object initiating a search, among those without a replica: at the end of the simulation, every *servent* will have done just one search and will hold one replica.

Since that search cost is directly related with the search scope, the goal of our experiments is to measure the cost introduced by our model by measuring the long path necessary to find an object and the load generated by queries.

Our simulation results, see Figure 1, reveal that the greatest long-path length to reach a replica is very small (less than 3, order of *O(1)*). It is also possible to see that approximately after that 50% of *servents* have executed a query, and have got a replica; the long-path to reach a replica is almost 1. This result is comparable with results for DHT based P2P networks. Based on other studies [24], the characteristic diameter in Gnutella is smaller than 12 hops and over 95% of the nodes are at most 7 hops away from one another. In our case, the diameter is smaller than 6 for each cluster of 500 *servents* and almost 90% of the *servents* are at most 5 hops away. Nevertheless, with less than 3 hops a query can always be resolved and on average 1.2 hops to big groups (4 in Gnutella). Therefore we obtain lower search cost and better performance. In addition, popular objects (high number of searches) are easier to find (more replicas) than non-popular objects (low number of searches).

Figure 2 shows both the number of generated messages until the moment the first answer to a query is received (inferior scope), and the amount of messages generated by queries until the TTL expires (superior scope). We achieve low load using a TTL=5, but we could reduce it without significant effect to the hit search using

TTL=3. This value does not change substantially with the size of a cluster, as it depends on the small-world property of the cluster.

Figure 3 shows the amount of participating *servents* until the first answer to the query is obtained, and the number of participating *servents* until the TTL expires. As can be seen, the amount of participating *servents* decreases with time given that as time passes more copies will exist in the cluster and therefore objects will be located faster (when an object is found, the query does not propagate beyond that peer), thus limiting both the amount of messages generated and the number of participating *servents*. Replication helps to reduce the search cost. A typical search in Gnutella can cover up to 1000 *servents* (more than 2 orders of magnitude).

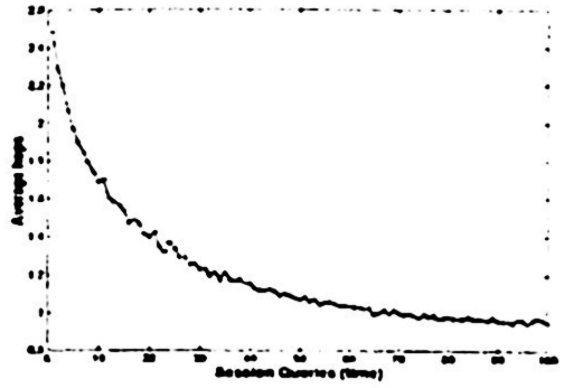

Fig. 1. Average number of hops

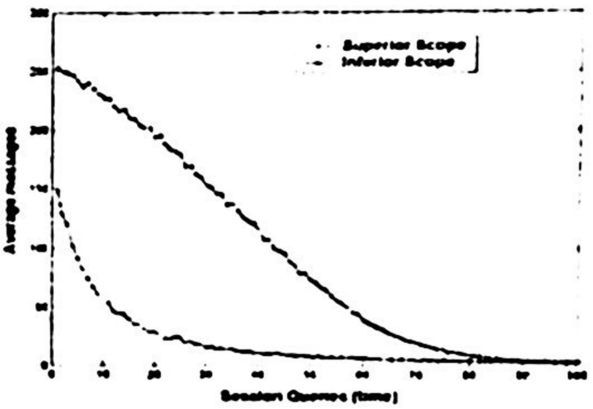

Fig. 2. Average number of messages

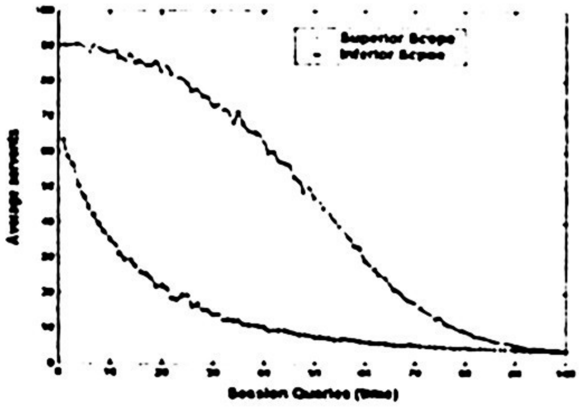

Fig. 3. Average number of Servents

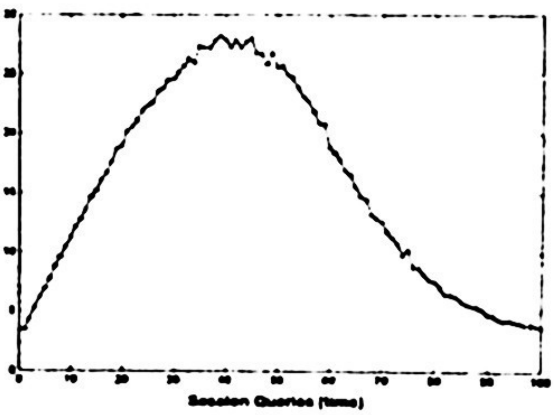

Fig. 4. Average number of Responding Servents

Our experimental results show that there always exists a *servent* that can resolve a query, i.e. search always is successful at locating objects. The minimum average number of *servents* that respond to a query is 3.5 at the beginning of our simulations, when there are only few object replicas (on the order of the average *servent* degree). The top number of *servents* answering a query is obtained in the middle of the experiment: roughly 50% of the nodes can give an answer to a query after 50% of the nodes have obtained a copy of an object. In 99% of the cases there is more than one *servent* to respond to a query. Nevertheless, *servents* rarely need all of the results of a search. By reducing the scope of search, we can greatly reduce the amount of messages that need to be sent and improve the scalability.

Figure 4 shows that in the last queries fewer *servents* respond to the query; this is the result of *servents* with a replica not propagating the query beyond. In Figures 3 and 4 we can see that replication helps to improve scalability given that the more replicas having the cluster, the lower number of participating *servents* will participate to resolve the query. As can be seen in Figure 1, in a cluster with up to 100 *servents*, it is always possible to obtain a query response with TTL=3. With this value, the number of involved *servents* and the amount of messages would decrease drastically. These values do not change significantly for other topologies with a similar or larger number of nodes (tested up to 500 peers), as they depend on the small-world property.

6 Related work

Previous work carried out on collaboration networks [25, 26] has been centred on trying to improve network performance, leaving aside the relevant fact that computer networks give support to social networks with distinctive statistical properties. Unlike previous work, the main idea of our proposal is based on the properties belonging to collaborative social networks.

Iamnitchi [27] put forward ideas about making use of the small-world property and clustering in scientific social networks. In [27], some mechanisms are proposed to facilitate searching, but without suggesting any particular model.

Other [27,28] related work concentrates on identifying clusters of interest to improve the performance of search process so that queries can be steered to peers that are more likely to have an answer. Unlike these works, we do not identify clusters, given that clusters are formed by users through explicit affiliation with groups. Cluster identification algorithms could assign a user node to a cluster with only a subset of files of interest. Letting the user select which groups wants to join guarantees he will be a member of communities of his interest, and have the relevant documents close to hand.

7 Conclusions

A great deal of research work seeks to develop better methods of locating data in P2P networks. These efforts are aimed at improving scalability, greater reliability under dynamic conditions, more efficient searching, and improved performance. The main problem with these systems, however, is that they ignore the fact that computer networks, such as P2P, are made up of people who in turn form social networks with statistical properties which affect the way these networks function.

In this work, we present a proposal for a new P2P model for collaboration networks using the social network topology, and exploiting the inherent characteristics of such networks: small-world, clustering, community structure, assortative mixing, preferential attachment and small and stable groups.

We also show how the combination of interest-based replication mechanisms with CSN properties can assist in appreciably reducing message overload in the network, compared with overload generated by searching in traditional P2P systems, thus improving performance.

We argue that our model use simple search strategy capable of dramatically decreasing the search cost (search results in *O(1)* hops) compared to a traditional P2P application on the same context. The simulation results show that our model is scalable and has good performance comparable with those that use DHT.

Although this work is focused on collaborative networks, we believe that many of the ideas set out in this paper can also be applied to other types of P2P networks.

The model presented here is an initial approach to making use of CSN topological properties, and as such we are aware that there is still room for improvement. At present, we are engaged in the implementation of a prototype that will enable us to assess improvements in possible extensions of the model.

Acknowledgements

This work has been supported by the Mexican Institute of Petroleum (Instituto Mexicano del Petróleo) and the Spanish MCYT project TIC2002-04258-C03-01.

References

1. S. Wasserman and K. Faust, "Social Networks Analysis", Cambridge University Press, Cambridge. 1994.
2. R. Albert, A.-L. Barabasi, "Statistical mechanics of complex networks". Rev. Mod. Phys., vol. 74, pp. 47-97, 2002.
3. V. Krebs, "The social Life of Routers, Applying Knowledge of Human Networks to the Design of Computer Networks". The Internet Protocol Journal, Vol 3, Num. 4, 14-25, Dec. 2000.
4. S.N. Dorogovtsev and J.F.F. Mendes "Evolution of networks". Advances in Physics 51, 1079-1187. 2002; arXiv:cond-mat/0106144.
5. D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks". Nature 393, 440-442, 1998.
6. L. A. N. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley, "Classes of small-world networks". Proc. Natl. Acad. Sci., Vol. 97, No. 21, 11149-11152, 2000.
7. M. A. Jovanovic, F. S. Annexstein, K. A. Berman. "Scalability Issues in Large Peer-to-Peer Networks - A Case Study of Gnutella". University of Cincinnati, Technical Report 2001.
8. M. E. J. Newman, "Clustering and preferential attachment in growing networks". cond-mat/0104209.
9. M. E. J. Newman and M. Girvan, "Mixing patterns and community structure in networks". Proceedings of the XVIII Sitges Conference on Statistical Mechanics, Springer Verlag, Berlin, 2003.
10. R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt, A. Arenas, "Self-similar community structure in organisations". arxiv.org/pdf/cond-mat/0211498, Nov 2002.
11. R. Albert, H. Jeong, and A.-L. Barabasi, "Error and attack tolerance of complex networks". Nature 406, 378-382 July 2000.
12. D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts, "Network robustness and fragility: Percolation on random graphs". Phys. Rev. Lett. 85 5468-5471, 2000.
13. L. A. Adamic, R. M. Lukose, A. R. Puniyani and B. A. Huberman "Search in Power-Law Networks Internet". Physical Review E, 64 46135, 2001.

14. A. Iamnitchi. M. Ripeanu. I. Foster "Locating Data in (Small-World?) Peer-to-Peer Scientific Collaborations". 1st International Workshop on Peer-to-Peer Systems, Springer-Verlag. 2002.
15. JXTA Homepage: www.jxta.org
16. M. Newman, D. Watts, S. Strogatz, "Random graph models of social networks". arXiv:cond-mat 0202208. Proc. Natl. Acad. Sci., to appear.
17. D. Liben-Nowell, H. Balakrishnan, D. Karger. "Observations on the Dynamic Evolution of Peer-to-Peer Networks". In the proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS '02). March. 2002; Cambridge, MA.
18. K. Aberer, M. Hauswirth, M. Punceva, R. Schmidt, "Improving Data Access in P2P Systems". IEEE Internet Computing. 6(1). January/February 2002.
19. Clip2 Distributed Search Solutions, "Gnutella Protocol Specification v0.4", 2001, www9.limewire.com/developer/gnutella_protocol_0.4.pdf
20. Q. Lv, P. Cao, E. Cohen, K. Li, and Scott Shenker. "Search and Replication in Unstructured Peer- to-Peer Networks". Proc. ACM ICS 2002.
21. R. A. Golding. "Weak-Consistency Group Communication and Membership". PhD thesis, University of California, Santa Cruz, Computer and Information Sciences Technical Report UCSC-CRL-92-52, December 1992.
22. J. M. Marquès, L. Navarro. "WWG: a Distributed Infrastructure to support groups". Proceedings of the ACM Conference: Group 2001 (Group'01).
23. J-sim homepage: www.j-sim.org
24. K. Sripanidkulchai, B. Maggs, and H. Zhang "Efficient Content Location Using Interest-Based Locality in Peer-to-Peer Systems", Infocom 2003.
25. BSCW Homepage (Basic Support for Cooperative Work), http://bscw.gmd.de/
26. Groove Homepage: www.groove.net.
27. A. Iamnitchi. "Resource Discovery in Large Resource-Sharing Environments. PhD Thesis. Adriana". Department of Computer Science The University of Chicago Dec 2003.
28. K. Sripanidkulchai, B. Maggs, and H. Zhang, "Enabling efficient content location and retrieval in peer-to-peer systems by exploiting locality in interests", ACM Computer Communication Review, vol. 32, Jan. 2002.