A Community-Based Peer-to-Peer Model Based on Social Networks

Amir Modarresi[1], Ali Mamat[2], Hamidah Ibrahim[2], Norwati Mustapha [2]

Faculty of Computer Science and Information Technology Universiti Putra Malaysia

Summary

Improving search performance is an important issue in peer-to-peer (P2P) network systems. The structure of underlying models has a direct effect on the performance of the search algorithms. In unstructured system like Gnutella query flooding algorithm suffers from poor scalability and considerable network overhead. In structured systems, algorithms like CAN and CHORD provide better performance, but they need more administrative tasks and have limited functionality in search. Our proposed model is a semi-structured, based on social networks which uses flooding algorithm for searching. Nodes in the model are grouped into several communities and sub communities with similar interests which provide lower distance and better locality in search. A simulation of the model shows lower path and better clustering than a random network.

Keywords:

Peer-to-Peer computing, social network, community, Model

1. Introduction

P2P systems form a network structure where the concepts of social networks are applicable. These concepts help designers to catch more information about a group of people who are using the network and the result will provide better services for the group according to their interests and needs.

From theoretical point of view, P2P systems create a graph in a way that each node will be a vertex and each neighborhood relation between two nodes will be an edge of this graph. When no criterion is considered for choosing a neighbor, this graph will be a random graph [1]; however, two important factors change this characteristic in P2P [2]: 1) principal of limited interest which declares that each peer interests in some few contents of other peers and 2) spatial locality law. Since each node represents one user in the system, a P2P will be a group of users with different interests who try to find similar users. Such structure creates a social network. However, Barab´asi [3] has shown that in the real social network the probability of occurring a node with higher degree is very low. In other words, the higher the degree the least likely it is to occur. This relation is defined by the power law distribution, i.e. $p(d) = d$ where *k>0* is the parameter of distribution, for degree of network nodes. The network model which has been defined with characteristics in (3) has a short *characteristic path length* and a large *clustering coefficient* as well as a degree distribution that approaches a power law. Characteristic path length is a global property which measures the separation between two vertices; whereas clustering coefficient is a local property which measures the cliquishness of a typical neighborhood.

As an example, we envision the scenario of sharing knowledge among researchers. Since each researcher has a limited number of interests, he can communicate with other researchers who work in the same area of interests. Because of many limitations like distance and resources researcher usually work with their colleagues in the same institute or college. Sometimes these connections can be extended to other places in order to get more cooperation. This behavior defines a social network with some dense clusters where these clusters are connected by few connections like figure 1. If one researcher is represented by one node, a P2P system will be created which obeys social network characteristics.

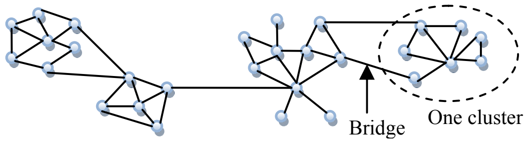

Figure 1: Many related clusters create a community

2. Related Works

Different structures and strategies have been introduced for P2P system for better performance and scalability. This part mainly reviews those approaches which focus on community and peer clustering.

Locality proximate clusters have been used to connect all peers with the same proximity in one cluster. Number of hop counts and time zone are some of criteria for detecting such proximity. In [4] the general clusters have been introduced which support unfixed number of clusters. Two kinds of links, local and global, connect each node to other nodes in their own cluster or nodes in other clusters. This clustering system doesn't concern about content of nodes. Physical attributes are the main criteria for making clusters. In [5] a Semantic Network Overly (SON) has been created based on common characteristics in an unstructured model. Peers with the same contents are connected to each other and make a SON which is actually a semantic cluster. The whole system can be considered as sets of SONs with different interest. If a peer, for example, in SON S_1 searches contents unrelated to his group, finding proper peer is not always very efficient. If there is no connection between S_1 and the proper SON, flooding must be used.

Common interest is another criterion for making proper overlay. In [6] all peers with the same interest make a connection with each other, but locality of peers in one interest group has not been concerned. In [7] all peers with the same interests are recognized after receiving many proper answers based on their interests. Such peers make shortcuts, a logical connection, to each other. After a while a group of peers with the same interests will be created and the richer peer in connection will be the leader of the group. Since this structure is based on unstructured system and receiving proper answer in the range of the issued queries, we cannot expect that all peers with the same interests in the system are gathered in one group.

In [8] communities have been considered. Authors have described community as the gregariousness in a P2P network. Each community is created by one or more peers that have several things in common. The main concern in this paper was connectivity among peers in communities. They have explained neither the criteria of creation nor size of each community. In [9] communities have been modeled like human communities and can be overlapped. For each peer three main groups of interest attributes have been considered, namely personal, claimed, and private. Interests of each peer and communities in the system are defined as collections of those attribute values and peers whose attributes conform to a specific community will join it. Since 25 different attributes have been used in the model, finding a peer which has the same values for all of these attributes is not easy. That is why a peer may join in different communities with partial match in its attributes. Although the concept of communities is the same as our work, in our model a shared ontology defines the whole environment and one community is part of the environment. There is also a bootstrapping node in each domain in order to prevention of node isolation. Our model also uses such nodes, but their main role is controlling sub communities. [10] uses a shared ontology in unstructured P2P for peer clustering. Each peer advertises his expertise to all of his neighbors. Each neighbor can accept or reject this advertisement according to his own expertise. Expertise of each peer is identified by the contents of files which the peer has stored. Since the ontology is used, a generic definition for the whole environment of the model is provided which is better than using some specific attributes.

Super peers have also been used for controlling peer clustering and storing global information about the system. In [11] super peers are used in partially centralized model for indexing. All peers who obey system-known specific rules can connect to a designated super peer. It creates a cluster that all peers have some common characteristics. Search in each cluster is done by flooding, but sending query to just a group of peers will produce better performance. According to these rules, super peers who control common rules must create larger index; therefore they need more disk space and CPU power. In [12] instead of using rules, elements of ontology are used for indexing. In this structure each cluster is created based on indexed ontology which is similar to our method. All peers with the same attribute are indexed. Our model also uses super peers and elements of ontology for indexing, but instead of referring to each node in the cluster, super peers refer to the representative of that cluster which controls sub communities of a specific community. This will reduce the size of index to number of elements in ontology which is usually less than the number of peers in a large system and provide better scalability.

3. Overview and Basic Concepts of the Proposed Model

Our proposed model is community-based and semi structured. It uses ontology for defining the environment of the system and creating communities. It also uses super peers for referring to these communities. Sub communities are considered for better locality inside each community. Below the main concepts of the method are introduced.

3.1. Community Concepts

A social network can be represented by a graph G(V, E) where V denotes a finite set of actors, simply people, in the network and E denotes relationship between two connected actors such that $E \subseteq V \times V$. Milgram [13] has shown that the world around us seems to be *small*. He experimentally showed that average shortest path between each two persons is six. People usually make a social cluster based on their interests but in different size. Such clusters which are usually dense in connections are connected to each other by few paths. All of these clusters with similar characteristics create a community. In each community: 1) each person must be reachable in reasonable steps (what Milgram named as small world) and 2) each person must have some connections to others which are defined by clustering coefficient. With such characteristics some structure like tree or lattice cannot show the behavior of social network. As stated previously in section 1, each dense cluster in the network is connected to few other clusters. In each cluster, some individuals who are called hubs are more important than others, because they have more knowledge or connections than other individuals. In order to join to a cluster as a new member either a known person or a member of the cluster must be addressed.

3.2. Definition of the Model

Providing a rigid structure increases administrative task burden; therefore we try to define our model as simple as possible that all nodes can contribute in the model.

The model *M* has a set of peer *P* where: $P = \{p_1, p_2, \ldots, p_i$. Each peer p_i can have *d* different direct neighbors which makes set N_i and each of them are identified by d_{ij} defined j^{th} neighbor of peer *i* as: $N_i = \{d_{i1}, d_{i2}, \ldots, d_i$. The number of neighbors for each peer is controlled by power law distribution; therefore few nodes have so many connections. d_{ij} also defines one specific peer in *P* like p_k. As a direct neighbor, p_k is one logical hop away from p_i which makes an overlay above physical network. Physical connection between p_i and p_k may not a one hop connection.

The shared ontology *O* is used to define the environment of the system. Interests of peers are identified according to the ontology. O is stored in each peer in order to understand the structure of the environment.

Based on ontology *O* many logical communities can be identified. Each community is populated by nodes with the same interests. Therefore all peers with the same interest can be identified by that community. This is in contrast of unstructured systems and even those systems which construct local shortcut among peers; however, in those systems all peers with the same characteristics are not reachable, in our model this is possible. The calculation of characteristic path length shows this matter in the next section. All the communities in the system are identified by *C* as we say:

$$C = \bigcup_{l=1}^{r} c_l \, , c_l \neq \emptyset$$

Contents of shared files in each p_i identify the interest of p_i. The content of each file is introduced by RDF language comprises with shared Ontology O. by using such a notation, posing more flexible queries is possible. If p_i has different kinds of files which distinguish different interest, p_i can contribute in different community c_l, as a result, two communities can be connected to each other via p_i. If all communities are connected to each other all peers are reachable. Namely: $\forall c_i, c_j : c_l \cap c_j \neq$. This condition depends on the contents of data and peers in the system. Other elements of the model which are introduced below, guarantee this condition.

Inside each community, there are some peers who are rich in contents and connections. These peers are called hubs. They create set h_l for c_l. Formally, we have: $\forall C_l : h_l = \{p_1, p_2, \ldots, p_s$ Where *S* is number of sub communities or hubs in the community, identified by policy of the system. Each hub defines a sub community inside the community. The creation of sub communities increases cluster coefficient of the system and move the model toward the small world. Although this phenomenon is occurred in all unstructured system, unrecognizable hubs and sub communities decrease the chance of finding a proper hub in the system. Consequently, most peers start the search process from peers who are weak in the system. This process increases more network resource consumption and longer delay.

Each community contains at least one member as a known member who is the representative of that community. This role is usually granted to the first peer who defines a new community c_l and identified by r_l. We can consider a fellow f_l for representative r_l in community c_l, for reducing failure rate of the community when representative leaves the network. When the community populated, r_l refer just to hubs inside the community. Since number of sub communities inside each community is few, representatives do not need extra resources like CPU power or disk space. As the first known member of the community, representative can help other peers to settle in better place. Since in the real world, each community is a set of clusters or sub communities and members of each cluster usually obey some kind of proximity, such a structure must be considered in the model. Good criteria to address the proximity can be number of hop, or IP address as a less precise metric. While all peers in one community have similar interest, located peers with closer number of hop, it may provide closer distance among peers. Such configuration gives better response time for queries whose answers are in one community. In other word, locality of interest will be established in a better form in the community. This is done by introducing all hubs in the community to a peer who likes to join that community. The new peer can calculate his distance from each hub by sending a control message. The result for the first connection of the new peer will be a hub, a good source of contents and connections, with a close distance. Peers according to their desire and/or capabilities can make more connections with other peers. This changes the structure of the model from tree-like structure to a graph which increases cluster coefficient of the system.

Figure 2 shows one community which is defined by a shared ontology and contains two sub community and its representative.

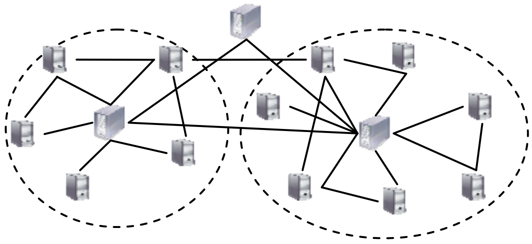

Figure 2: One Community with two sub Communities and its representative

M also has a set of super peer *SP* where: $SP = \{sp_1, sp_2, \dots, sp_m\} \text{ and } m \ll$. Sp_i refers to representatives of each community; therefore each community is identifiable in the system. Sp_i also stores the shared ontology of the system. This helps Sp_i to have a great view from all the system. As a bootstrap server, Sp_i can guide each new peer to a proper community just by knowing the interest of the peer. Since communities are mostly created based on the elements of the ontology, and it is much less than number of peers in the system, the size of index in the Sp_i will be smaller than other super peers who work in semi structured model and need to index all peers or group of peers in the system. On the other hand, it provides the interconnectivity of whole system.

3.3. Joining a peer

Since all interests in the system are represented by the shared ontology, a peer p_i can introduce his interest based on this ontology. This interest is identified by the contents of the sharing files which are represented by RDF language. When p_i wants to join the system, he sends his interest to the super peer Sp_j. Sp_j returns the proper community by sending the address of its representative, related to the interest of p_i. P_i starts to communicate with representative r_l and get address of all hubs in the community. By communicating with each hub, p_i chooses the closest hub according to lowest delay and shortest physical hop. The algorithm below shows this process.

```
Let $I_{int}$ defines interest of pi
Let $Rep_{Add}$ defines the address of the representative of a proper community
Let $Hub_{Add}[]$ defines the address of all hubs in the community
Peer $p_i$ sends $I_{int}$ to super peer $sp_j$ do
while ($Rep_{Add}$ has not arrived and timer has not expired)
if (RepAdd!=null) then
    ask $rep_{Add}$ for hubs
    if ($Hub_{Add}$!=null) communicate with $Hub_{Add}$ to find the closest hub and make a connection
    else $p_i$ is sets as the hub of the community
else $p_i$ sets as the representative of the community
```

3.4. Message Communication Function

Each peer p_i has a routing table in order to direct queries to proper destination. Along each record in the table, the weight of the connection, interest of the connected peer, number of physical hops and delay during last query is stored. Each query is sent to peers who have similar interest and highest weight with the posed query. If peer p_i has many connections to hubs and other peers, hubs will have the first priority. Peers will be chosen when they only have enough weight for answering queries. The reason for this action is that our model creates more effective connections than a random model. At the time of joining peers choose a strong peer (hub) as their first neighbor. Other neighbors are chosen later during the life time of the peer when he receives proper answers from a specific peer many times. On the other hand in a random network a peer may starts the search process with many weak peers in contents and connections which increase delay and traffic. By directing queries to richer peer delay and traffic are decreased.

3.5. Query Resolution

Each peer p_i has a repository contained a description for all shared items based on ontology *O* defined by RDF language. For example, for a publication confront of ACM ontology [14], we can have piece of code like below:

```
<Publication rdf:about="dblp:persons/books/ph/Tomlin90">
<title>Geographic Information Systems and Cartographic Modelling</title>
<acm:topic
rdf:resource="http://daml.umbc.edu/ontologies/classification#ACMTopic/Information_Systems"/>
</Publication>
```

Title and topic from such a notation can identify select and where-clause of a RDF-based query language like SPARQL [15]. In this way different type of queries can be issued. For a query related to interest of a community, a range query can be answered one hop away. For simple query related to the interest of the community, answers are provided in shorter distance than random model. For all unrelated queries to the interest of the community, queries are sent to the super peer sp_j in order to get the address of the proper community based on interest of the query and then it is sent to the representative of the identified community. Queries can also be relaxed based on relationship of the communities which are defined in the shared ontology, if number of answers doesn't stratify users. Figure 3 shows main component of each peer.

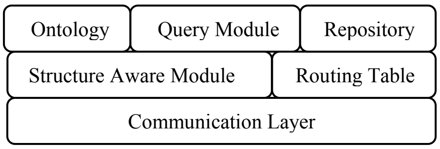

Figure 3: Main components of each peer and their relationship

4. Simulation Setup

We wrote a simulator to create a computer based community model to show the behavior of the system and in what extend they are close to a social network. We summarize an example as an instantiate of our model. A computer scientist regularly has to search publications or correct bibliographic meta data. A scenario which we explain here is community of researchers who share the bibliographic data via a peer-to-peer system. Such scenarios have been expressed in [16] and [10]. The whole data environment can be defined by ACM ontology [14]. Each community in the system is defined by an element of the ontology and represented by a representative node. Each community comprises of many sub communities or clusters which are gathered around a hub. Figure 4 depicts this example.

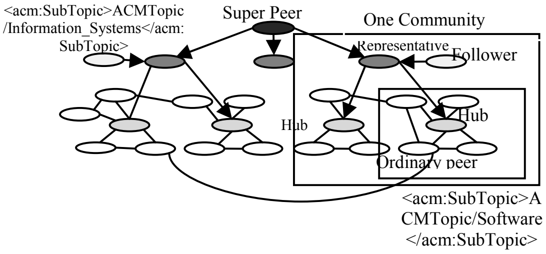
<acm:SubTopic>ACMTopic /Information_Systems</acm: SubTopic>
Super Peer
One Community
Representative
Follower
Hub
Hub
Ordinary peer
<acm:SubTopic>ACMTopic/Software </acm:SubTopic>

Figure 4: Principle elements of the model

We define the number of peers in the model in advance, 1000 nodes, and identify a capacity for making connections with other peers based on power law distribution. In this simulation, joining and leaving of peers are not considered. The first peer who joins the community is chosen as the representative of the community. The second one is his follower which works as a back up when representative is failed. Based on the definition of the model many peers who are richer in connection are chosen as hubs. Number of hubs and sub communities are defined based on the capability of hubs in establishing connections and number of peers in the whole community. Since hubs are normal peers with higher capacity for accepting connection, if all the connections have already been used another hub will be chosen. Such a restriction in connection limitation has many reasons. First, it allows controlling the connection distribution in the system. Second, after all hubs are full, the new peer must connect to other normal peers. This mimics the behavior of joining a member to a community by another member. If the new peer has capacity more than one connection, other neighbors will be chosen randomly. First all the members inside the same sub community are chosen because they may have shorter distance and then, if all peers cannot accept any more connections, the other peers from other sub communities are chosen. These kinds of connections create potential bridges among sub communities which make different sub communities are connected except representative of the community. It increases the cluster coefficient of the model. Since the locality is important, such connections will be established when the target peers is rich in favor contents.

Table 1: The value of cluster coefficient for a random network and the model with different sub communities

Max Connection	100 Hubs	50 Hubs	10 Hubs	Random
10	.42	.28	.14	.005
20	.43	.41	.24	.008

Table 2: The value of characteristic path length for a random network and the model with different sub communities

Max Connection	100 Hubs	50 Hubs	10 Hubs	Random
10	2.9	3.67	4.63	5.006
20	2.85	2.91	3.46	4.21

Watts in [17] has shown a small world graph is a graph which is located among regular and random graphs. Such a graph has a characteristic path length as low as random graph and cluster coefficient as high as regular graph. The highest cluster coefficient belongs to fully connected graph and shortest path is obviously 1. So we calculate cluster coefficient and characteristic path length for the model.

Table 1 shows the result of cluster coefficient for a community with 1000 nodes and a random network with 1000 nodes. The capability for accepting the maximum connections and the number of hubs, sub communities, are changing.

As it can be expected, by defining sub communities the cluster coefficient is increased even with just one sub community. With the small number of hubs and less capability to accepting connection, many peers are connected to each other without any connection to any hubs. This effect defines the longer characteristic path length in table 2. When number of connection increases, the cluster coefficient is also increased. Moreover, there will be more chance for other peers to connect to hubs. It decreases the characteristic path length. When capability of accepting connections is high, more than number of peers in the community, the graph of the model is moving toward complete graph. This explains larger value for cluster coefficient, but because of many points of references in the model, hubs, the characteristic path length is decreased. It shows that, although the model moves toward complete graph it is not a random one. Needless to say, when peers have high capability in accepting connections, many other clusters are created inside the sub community. Since they are implicit, reaching for them won't be very fast, except through its explicit sub community.

Since the results show the path length for one community, by adding 2 extra steps the value for the whole model is calculated. This is the average path length when one peer in one community tries to reach another one in different community through the available super peers in the model.

5. Conclusion

Defining an environment by an ontology let all related peers gather in one community. In this way all peers are reachable in the system. On the other hand, dividing a community to many sub communities increases clustering coefficient and decreases path length which result better information retrieval. A reasonable tradeoff between maximum number of connections in the system and number of sub communities can reduce resource consumption in nodes and index size in representatives. In other words, a semi structure P2P model like our model can be constructed with regular nodes instead of powerful nodes by using sub communities.

References

[1] *On Random Graphs.* Erdős, P. and Rényi, A. 1956.

[2] *Efficient Content Location in Peer-to-Peer Systems.* Chen, H., Z., Huang and Gong, Z. s.l. : In proceedings of the 2005 IEEE International Conference on e-Business Engineering (ICEBE'05), 2005.

[3] *Emergence of Scaling in Random Networks.* Barab´asi, Albert-L´aszl´o and Albert, R´eka. s.l. : Sience, 1999, Vols. 286:509-512.

[4] *General Clusters in Peer-to-Peer Networks.* Hu, T. -H and Sereviratne, A. s.l. : ICON, 2003.

[5] *Semantic Overlay Networks for P2P Systems.* Crespo, A. and Garcia-Molina, H. USA : Agents and Peer-to-Peer Computing (AP2PC), 2004.

[6] *Efficient Content Location Using Interest-Based Locality in Peer-to-Peer Systems,.* Sripanidkulchai, K., Maggs, B. M. and Zhang, H. s.l. : INFOCOM, 2003.

[7] *An Interested-based Architecture for Peer-to-Peer Network Systems.* Chen, Wen-Tsuen, Chao, Chi-Hong and Chiang, Jeng-Long. s.l. : AINA 2006, 2006.

[8] *Interconnected Peer-to-Peer Network: A Community Based Scheme.* Shijie, Z., et al. s.l. : AICT/ICIW 2006, 2006.

[9] *Structuring Peer-to-Peer Networks Using Interest-Based Communities.* Khambatti, M., Dong Ryu, K. and and Dasgupta, P. s.l. : DBISP2P 2003, Springer LNCS 2944, 2003.

[10] *Bibster - A Semantics-Based Bibliographic Peer-to-Peer System.* Haase, P., et al. s.l. : In international Semantic Web Conference 2004, 2004.

[11] *Super Peer-Based Routing and Clustering Strategies for RDF-Based Peer-to-Peer Networks.* Nejdl, W., et al. s.l. : In proceedings of WWW 2003, 2003.

[12] *HyperCuP—Hypercubes, Ontologies and Efficient Search on P2P networks.* Schlosser, M., et al. Bologna, Italy : In International Workshop on Agents and Peer-to-Peer Computing, 2002.

[13] *The Small World Problem.* Milgram, S. s.l. : Psychology Today, 1967, Vols. 1(1):61-67.

[14] ACM. *1998 ACM Computing Classification System.* [Online] http://www.acm.org/class/1998.

[15] *SPARQL.* [Online] www.w3.org/TR/rdf-sparql-query.

[16] *OAI-P2P: A Peer-to-Peer Network for Open Archives.* Ahlborn, B., Nejdl, W. and Siberski, W. s.l. : In 2002 International Conference on Parallel Processing Workshops (ICPPW'02), 2002.

[17] *Collective Dynamics of 'Small World' Networks.* Watts, D. and Strogatz, S. P.440-442, s.l. : Nature Journal, 1998, Vol. 393.