Modeling Peer-to-Peer Networks with Interest-Based Clusters

Bertalan Forstner, and Dr. Hassan Charaf

***Abstract*—**In the world of Peer-to-Peer (P2P) networking different protocols have been developed to make the resource sharing or information retrieval more efficient. The SemPeer protocol is a new layer on Gnutella that transforms the connections of the nodes based on semantic information to make information retrieval more efficient. However, this transformation causes high clustering in the network that decreases the number of nodes reached, therefore the probability of finding a document is also decreased. In this paper we describe a mathematical model for the Gnutella and SemPeer protocols that captures clustering-related issues, followed by a proposition to modify the SemPeer protocol to achieve moderate clustering. This modification is a sort of link management for the individual nodes that allows the SemPeer protocol to be more efficient, because the probability of a successful query in the P2P network is reasonably increased. For the validation of the models, we evaluated a series of simulations that supported our results.

***Keywords*—**Peer-to-Peer, model, performance, network management.

I. Introduction

As in P2P networks the required network bandwidth is very important, more and more attempts were made to develop more efficient and scalable protocols. After examining some existing systems, we developed a new protocol, the SemPeer, which was introduced in [15]. SemPeer is a new layer on existing protocols that utilizes the semantic information available by the stored documents to transform the P2P network to be able to benefit from the locality in interest. We needed a mathematical model to examine the theoretical capabilities of the new protocol. After examining different P2P models, we found that it is reasonable to prepare a new mathematical model that capture the aspects of the different fields of interest related to the nodes (users) in the system. The new model also describes the effect of clustering in the small-world netwok. We have executed a series of simulations to validate our new models.

The rest of this paper is organized as follows. After these introductionary paragraphs a short summary of the SemPeer protocol is presented. Section 3 summarizes the related work, followed by the description of the clustering problem. Section 5 describes a model to predict the performance of the different protocols. In Section 6 the results are compared to the outcome of the simulators, followed by a conclusion.

Bertalan Forstner is with the Budapest University of Technology and Economics, Department of Automation and Applied Informatics, 1111 Budapest, Goldmann György tér 3, Hungary (e-mail: bertalan.forstner@aut.bme.hu).

Hassan Charaf is with the Budapest University of Technology and Economics, Department of Automation and Applied Informatics, 1111 Budapest, Goldmann György tér 3, Hungary (e-mail: hassan.charaf@aut.bme.hu).

II. The SemPeer Protocol

A. Semantic Approaches

In this paper we do not focus on the earlier peer-to-peer technologies such as Napster [9] that suppose the attendance of central servers. We would rather compare the results to the popular fully decentralized systems first that use unstructured content location algorithms. A primordial of them is Gnutella [4], which is also a good benchmark protocol. The SemPeer semantic protocol is also based on Gnutella as a new layer. A great advantage of this approach is that Gnutella- and SemPeer-based nodes can work together in the same network. Nodes with low computing or storage resources (for example mobile devices) can use the standard Gnutella protocol which has low resource requirements, while the more powerful nodes can benefit from the advanced SemPeer Protocol.

Recently some systems were developed to improve the search performance. These all are built on the fact that the fields of interest of the nodes can be determined, or nodes with probably greater hit rate can be found. The first group of these algorithms tries to do this on run-time statistics only. For example, [10] introduces the concept of "shortcuts": nodes that could answer our queries in the past will probably answer some of them in the future, thus, they are worth putting them on a shortcut (neighbor) list. In return for the small amount of required overhead, the nodes would not contain any information on the kind of documents that a node in the shortcut list contains, hence this system requires many run-time statistics to find the best shortcut neighbors.

The second group of the content-aware peer-to-peer algorithms uses metadata provided for the documents in the system. We disapprove some of these algorithms, because they assume such kind of information that one would not expect in a real system. For example, [11] assumes that the user knows the keywords of the documents being searched for. As these keywords are produced practically by some algorithmic methods ([7], [8]) based on the document itself, we lost accuracy right at the beginning of the search, because we cannot expect the user to produce this keywords in any way in the absence of the requested document.

Another shortcoming of the elaborated structured content location algorithms is that they cannot generalize the collected semantic information. The already mentioned [11] and [12] store and use metadata for selecting the neighbors for semantic routing. However, they do not utilize deeper information, such as semantic relationships that can be exploited from the available data.

The SemPeer protocol [15] uses a well-known WordNet taxonomy, [13]. WordNet is an online lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets. Using this taxonomy we can avoid strict metadata matching, and construct a more efficient content retrieval system.

B. SemPeer: Extending Gnutella with Semantic Capabilities

Some initiatives are launched to make the Internet semantic, namely provide it with metadata. Ontology-based information retrieval makes the search more intelligent than string matching alone [2]. We already mentioned WordNet, and another good example is the Dublin Core Metadata Initiative. This project is dedicated to facilitate the widespread adoption of interoperable metadata standards and to develop specialized metadata vocabularies for describing resources that enable more intelligent information discovery systems. The viability and benefit of this initiative has been proven by the numerous projects built on it [3].

The Peer-to-Peer approach enables to make information retrieval more efficient using a model well-known from the everyday life. In the real world, working relationship is established among the people with a labor of the same topic. For example, if one's job is connected to the 19th century French literature, one's associates will have the same field of interest and probably have experience, books (documents), that is, relevant information, on the topic. If some related information needs to be found, then probably nobody would start with asking random people, but the mentioned experienced colleagues.

The Internet and the Peer-to-Peer makes it possible to contact those people with whom we cannot enter into relations, because of geographical or other barriers. In the basic peer-to-peer protocols, the mentioned circumstances do not play any role in the selection of adjacent nodes, thus search for the documents starts with querying the randomly selected neighbors. However, there are some methods elaborated to acquire ontology from documents [7], [8]. Then with some algorithmic methods, SemPeer creates profiles for the nodes that will describe the owner's fields of interests, for example with the appropriate weighting of the semantic categories provided by Dublin Core, or using the WordNet taxonomy. This profile creation can be fully automated. Thus, the construction of the peer network is not random, but we fundamentally consider that the fields of interest, namely, the profiles of the connecting nodes overlap as much as possible. As the individual nodes select their neighbors this way, we can assume that nodes in distance of two or more hops (the neighbors of our neighbors) also have a similar profile. This has the benefit of making the information retrieval more efficient, as the nodes reached during the lifetime of the request (TTL) have relevant information with greater probability than selecting the neighbors in a random way. With a good TTL and routing strategy, we can decrease the network traffic by minimizing the number of the request messages.

There are some other approaches that try to organize the nodes into clusters for better performance (for example [19]), but they are requesting all the nodes to use the advanced P2P algorithm. A summary of the challenges of these types of clustered P2P networks can be found in [20].

III. Related Work

Considerable research effort has been involved in the examination of the performance of networks with client-server architecture [21]. There are some models elaborated to analyze the throughput, response time and other parameters of the network. However, there are only very few papers concerning these issues of P2P networks. The main research directions can be characterized by the following types of network models.

The aspects of connection distribution of the large-scale P2P networks are modeled in [23]. This work describes the measures that affect the quality of service of the network, such as network latency or the short-circuit effect. However, it does not answer such questions such as the probability of success or the influencing parameters.

We found a very useful model in [22]. The main goal of this model was to capture network throughput for three different classes of P2P networks. The one that describes the P2P architecture of distributed indexing with flooding architecture is suitable to obtain probabilistic results for Gnutella networks. However, it can hardly be transformed to use with clustered SemPeer networks, but we can use it to validate new models in extreme cases, as we will do it later in this paper.

After examining the available models we found that we should elaborate a new one to fully describe the novel SemPeer protocol.

IV. Clustering Problem in Sempeer Protocol

Due to the SemPeer protocol, nodes with similar semantic profiles are connected with high probability. To be able to describe the connectivity of a graph in a formal way, we use a modified version of the clustering coefficient graph measure introduced by Watts and Strogatz [18].

The original formula can be introduced as follows. First, we define a graph in terms of a set of n vertices $V = \{v_1,v_2,...v_n\}$ and a set of edges E, where e_{ij} denotes an edge between vertices v_i and v_j. Below we assume that v_i, v_j and v_k are members of V.

We define the *neighborhood* N for a vertex v_i as its immediately connected neighbors as follows:

$$N_i = \{v_j\} : e_{ij} \in E \tag{1}$$

The degree k_i of a vertex is the number of vertices in its neighborhood $|N_i|$.

The clustering coefficient C_i for a vertex v_i is the proportion of links between the vertices within its neighborhood divided by the number of links that could possibly exist between them. For a directed graph, e_{ij} is distinct from e_{ji}, and therefore for each neighborhood N_i there are $2k_i(k_i - 1)$ links that could exist among the vertices within the neighborhood. Thus, the clustering coefficient is given as:

$$C_i = \frac{\left|\{e_{jh}\}\right|}{2k_i(k_i - 1)} : v_j, v_h \in N_i, e_{jh} \in E \tag{2}$$

This measure equals 1 if every neighbor connected to v_i is also connected to every other vertex within the neighborhood, and 0 if no vertex connected to v_i is adjacent to any other vertex connected to v_i.

Because of the nature of the Gnutella-based protocols, the high connectivity of the nodes with similar semantic profiles could lead to a very high clustering coefficient. This results in a query to arrive multiple times in different ways to some of the nodes in the group. Because of the connectedness, fewer nodes can be reached by a query, and also unnecessary computational resources are required. This can be described in a more formal manner as follows.

Consider a set of nodes, where the clustering coefficient is zero, i.e. no neighbors are connected with each other (Figure 1.a). In this case the number of nodes a query can reach is written as

$$M = \sum_{i=1}^{TTL} k^i \tag{3}$$

In Eq. (3), TTL represents the Time-To-Live parameter: a query should be progpagated through TTL hops. Now we consider the worst case, when the clustering coefficient is 1. In this case the neighboring nodes form a fully connected directed graph, thus, the number of nodes reached by a query are decreased to k (Fig. 1.b).

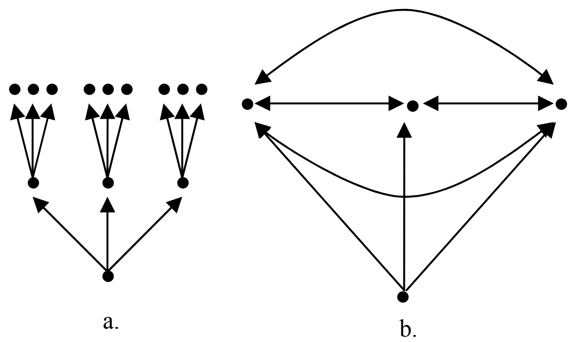

Fig. 1 Directed graphs with extreme clustering coefficients, k=3, TTL=2.a. C=0 b. C=1

In a standard Gnutella network the coefficient will be near zero as the graph can be regarded as a random mesh. In case of SemPeer this measure can be quite high, depending on the popularity of the given group. In Figure 2 it can be seen on a typical example that after certain queries the standard SemPeer reaches a saturation point: the clustering coefficient reaches a value, where the number of nodes reached by a query is strongly decreased, superseding the benefit from the intelligent neighbor selection. In bad circumstances it can occur that the SemPeer protocol delivers fewer positive answers than Gnutella does.

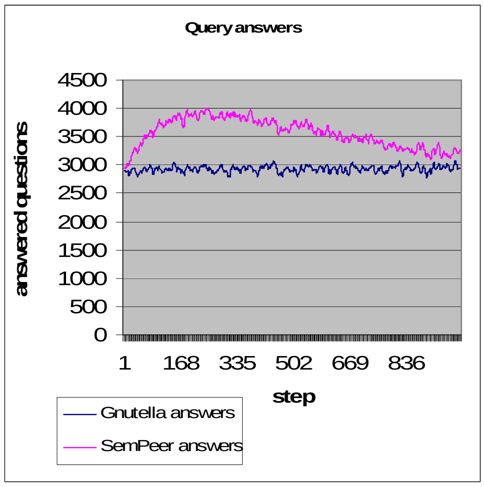

Fig. 2 Effect of the high clustering coefficient to the query answering capability of the SemPeer protocol in a typical example

In the case of SemPeer not only the high connectedness of the neighbors causes a problem as well as some other kinds of links: first, the links backward in the propagation tree; second, links between nodes in the same level (siblings in our wording); and third, links to neighbors of a sibling node. The first type decreases the nodes reached by a query in an obvious manner. The second and third types can also cause ineffective query propagation, because when a query is issued by a node, it can be propagated back with high probability to a node that has already received it. These three types of connections should be avoided. They can be seen on the graph representation marked with dotted lines in Fig. 3.

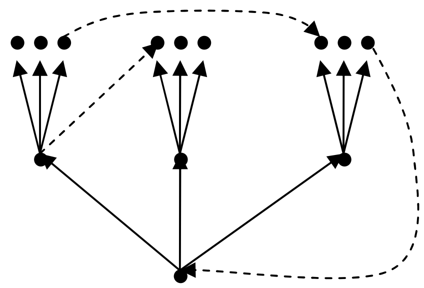

Fig. 3 The dotted links decreases the number of reached nodes

To be able to measure this kind of connectedness we introduce a modified clustering coefficient. This measure has to be 0 if the nodes reached by a query constitute a tree, and it approaches 1 as the number of the counterproductive links increases.

Let $\{E_r^*\}$ stand for the set of counterproductive links in the propagation tree of a query initiated by the node v_r. For the sake of simplicity, we assume that each node has the same number of neighbors. In that case the modified clustering coefficient for the node r is

$$C_r = \frac{\left|\left\{E_r^*\right\}\right|}{\sum_{m=1}^{TTL}\left[k^m\left(\sum_{n=0}^{m-1}\left(k^n\right)+k^m-1\right)\right]+\sum_{m=1}^{TTL-1}k^m(k^{m+1}-k)} \quad (4)$$

The explanation of this formula is the following. The denominator in (4) is the maximum number of the three different types of counterproductive links in a query. These three types are detailed below. The maximum number of backward links is

$$\sum_{m=1}^{TTL}\left[k^m\left(\sum_{n=0}^{m-1}\left(k^n\right)\right)\right], \quad (5)$$

because there are k^m nodes reached by the query at the m^{th} step from issuing, and such a node can be connected with all the nodes visited in previous steps. The count of all the possible links to the siblings is

$$\sum_{m=1}^{TTL}\left[k^m\left(k^m-1\right)\right] \quad (6)$$

because a node reached at the m^{th} step can be connected with the other nodes (k^m-1) reached at the very same step. Finally, the maximum number of links directed to the neighbors of the sibling nodes is counted to be

$$\sum_{m=1}^{TTL-1}k^m(k^{m+1}-k). \quad (7)$$

The task is to ensure that a query does not arrive to a node more than once in different ways because of the high clustering (short-circuit effect [23]). To find an optimal graph structure, we first define a minimum size for the loops in the SemPeer layer, that is, the number of nodes in a loop cannot be less than a predefined value. It can easily be seen in Formula (4) that if this value is not less than TTL+1, we eliminate the backward links ($n<m$ case).

In the advanced SemPeer protocol, we define partitions for the nodes in the system, and each node in a partition can only connect to a node in the next partition. This also eliminates the connections between nodes on the same level (the $n=m$ case). Each node has to identify the partition that it belongs to. To achieve this, we form a number from the address of each node with modulo division to define the corresponding partition. In that case the network topology is similar to that in Fig. 4.

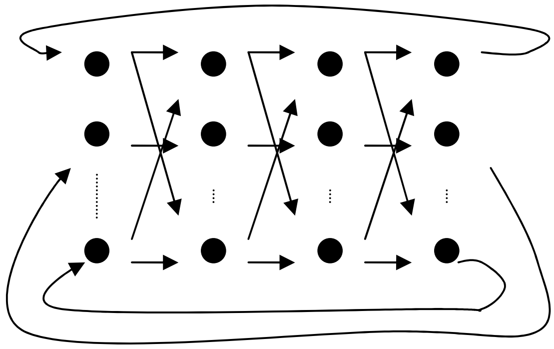

Fig. 4 The nodes of the network are partitioned

Until now we have not found an optimal distributed strategy to eliminate the third type of counterproductive links.

V. Modeling the Networks

Our goal is to increase the query hit, thus, we need a simple model to approximate the probability of a successful query in the standard Gnutella as well as in the SemPeer network. We regard the P2P network as a directed graph. Consider that the fields of interest of a user (represented by a node) can be determined with a single topic. We assume t different topics, the same number of nodes (V_t) and documents (D_t) with each topic, and every node obtaining D_n documents. The documents and the initiating links between the nodes are selected randomly with uniform distribution.

We can approximate the probability of a successful query in case of Gnutella as shown below:

$$P_{Success,Gnutella} = 1-\left(1-\frac{1}{tD_t}\right)^{D_n E_q} \quad (8)$$

In this formula, we compute the probability of not founding the requested document at any reached node, and subtract it from 1. The expression in bracket is the probability of selecting any disinterested document from all the documents that exist in the network. E_q is the number of reached nodes:

$$E_q = \sum_{i=1}^{TTL}\left[(1-C)k\right]^i \quad (9)$$

where C, the modified clustering coefficient is the average of all the C_r s.

A link can be counterproductive with the following probability:

$$P = \frac{\sum_{m=1}^{TTL}\left[k^m\left(\sum_{n=0}^{m-1}\left(k^n\right)+k^m-1\right)\right]+\sum_{m=1}^{TTL-1}k^m(k^{m+1}-k)}{t\,V_t\sum_{i=0}^{TTL}k^i},$$

From that the average number of productive connections per node follows:

$$(1-C)k = k * \left(1 - \frac{\sum_{m=1}^{TTL}\left[k^m\left(\sum_{n=0}^{m-1}\left(k^n\right)+k^m-1\right)\right]+\sum_{m=1}^{TTL-1}k^m(k^{m+1}-k)}{tV_t\sum_{i=0}^{TTL}k^i} \right) \quad (10)$$

From (10), C can be approximated by the following fraction:

$$C = \frac{\sum_{m=1}^{TTL}\left[k^m\left(\sum_{n=0}^{m-1}\left(k^n\right)+k^m-1\right)\right]+\sum_{m=1}^{TTL-1}k^m(k^{m+1}-k)}{tV_t\sum_{i=0}^{TTL}k^i} \quad (11)$$

In the case of SemPeer, we regard the steady state when nodes are connected only to other nodes from the same cluster. Therefore searching happens only in the set of documents related to only one topic, but also the clustering coefficient rises because the multiplier t in the denominator of the Formula 11 decreases to 1. The approximate probability of a successful query will be

$$P_{Success,SemPeer} = 1-\left(1-\frac{1}{D_t}\right)^{D_n E_q} \quad (12)$$

The clustering coefficient decreases when the modified SemPeer protocol eliminates the first two types of counterproductive links, it can be computed as (13):

$$C = \frac{\sum_{m=1}^{TTL-1}k^m(k^{m+1}-k)}{V_t\sum_{i=0}^{TTL}k^i} \quad (13)$$

This theoretical maximum is hard to reach because of the distributed property of the algorithm, but as the protocol transforms the network, the probability should tend to to this maximum value. It is still an open question how this convergence can be accelerated.

A. Comparing the Models

As described in the introduction, we compared our model to the one introduced in [24] when validating the results of the Gnutella case. The simplified version of the model described in that paper proposes the following formula for the probability of a successful query:

$$P_{Success,Gnutella} = \frac{TTL}{(V_t t)^{1/\beta}} \quad (14)$$

where $\beta>1$ is a parameter related to the connectivity of the topology formed by the peers. As in the case of the Gnutella architecture (distributed indexing with flooding architecture) the value TTL^β equals with the number of nodes reached by a query, and we regarded the clustering zero in the random graph, the β parameter can easily be calculated from our parameters from the following formula:

$$\beta = \log_{TTL} E_q \quad (15)$$

The comparison with this model in the reasonable cases gave the same probability. We have not found such a model that can use for comparison in the SemPeer cases.

B. Validating the Model with a Simulator

To validate our results and examine the behavior of our protocol we have evaluated a series of simulations on the GXS Peer-to-Peer Simulator [14]. Practically GXS is a single-threaded message dispatching utility operating in batch mode. A simulation itself is a script file containing a sequence of default (built-in) and protocol-specific (user-implemented) commands; and the results are dumped into a stream. GXS uses the concept of steps, which means that it handles a whole set of parallel events timed to a given step-number. There is a default command to increase the step number that simulates the progress of the time. The simulator supports two types of events: message arrival and expiration of timers. The events are handled in a first-come-first-served manner, since shuffling of ‘parallel’ events has no statistically recognizable effect on the results.

We assigned a field of interest to every simulated node, and generated documents with different keywords in these areas. After the initialization phase, every node starts to query for documents that it does not have. The keywords of the queried documents correspond to the semantic profile of the initiator node. This does not mean that the node knows any metadata about the document; this is only a way to simulate queries for documents in the same field of interest.

VI. Results

We describe a case study in this section to be able to compare the simulation results with those computed with the formulas. The test case was a simulation with 24000 nodes with 15 main fields of interest. Each node contained 40 documents from the set of 2600 documents per topic. Each node was connected to exactly 5 other nodes randomly. We set the TTL parameter of the protocol to 4.

From formula (8), the average response ratio in the case of Gnutella protocol is expected to be 0,5209. (We calculated the modified clustering coefficient to be 0,031427). The simulation result is illustrated in Figure 5. We can state that the simulation fully supported our model in the case of Gnutella.

In case of SemPeer protocol, we executed two simulations. In the first case, we used the SemPeer protocol without our modification proposal, hence, the clustering coefficient rose to a quite high value of 0,471407. In that case the model (Formulae 13 and 12) predicted the answer ratio to rise as high as 0,6922, which was again fully equivalent to the simulation results.

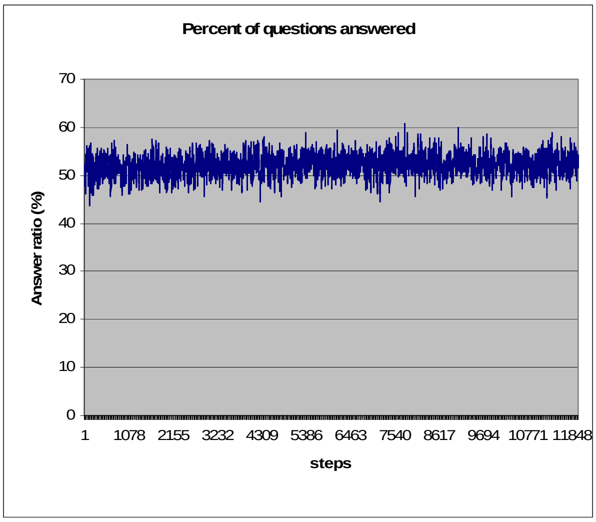

Fig. 5 Simulation results: percent of answered questions at Gnutella

With our protocol modification proposal the theoretic limit of the answer ratio was computed to be 0,9999 (the clustering coefficient decreased to 0,064504). Due to the document distribution and the number of connections, this boundary is difficult to achieve. The result obtained by the simulator shows that the probability of finding a document in this case will increase up to near 99 percent (Fig. 6).

From (15) we could calculate the missing parameter for the model of Ge to be β=4,9, which gave the same probability for the Gnutella network as (8) did.

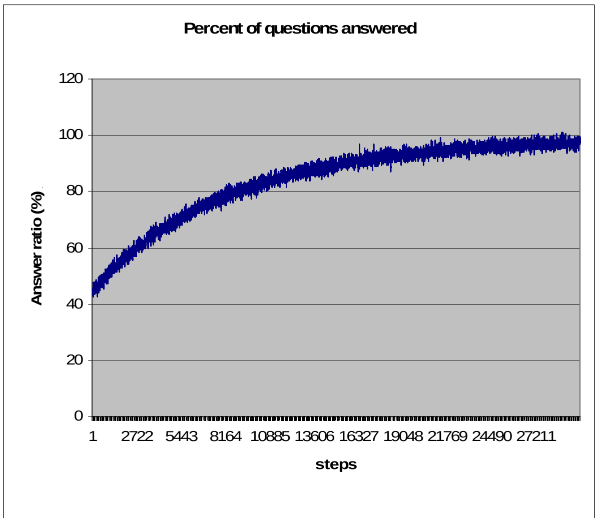

Fig. 6 The answer ratio increases at the case of the modified SemPeer Protocol

VII. Conclusion

We conclude that the SemPeer protocol can perform much better than the Gnutella and we have established a mathematical model, which can predict the gain precisely for the original SemPeer protocol. The advantage of the new protocol is the more intelligent neighbor selection that helps to increase the probability of the query hit and to decrease the amount of network traffic. However, every node should maintain a semantic profile that requires resources for computing and storing.

The modification to the novel protocol has some practical limitations yet that could be avoided with better document distribution, but this improvement is still subject of future research.

References

[1] Oram, A. (edited by), *Peer-to-Peer: Harnessing the benefits of a distruptive technology* (O'Reilly & Associates, Inc., 2001).
[2] OLeary, D., "Using ai in knowledge management. Knowledge bases and ontologies", *IEEE Intelligent Systems 13* (1998) pp. 34-39.
[3] The Dublin Core homepage, http://dublincore.org/. (Projects built on the Dublin Core, (http://dublincore.org/projects/).
[4] The Gnutella homepage, http://gnutella.wego.com
[5] Resnik, P., "Semantic similarity in taxonomy: An information-based measure and its application problems of ambiguity in natural language", *Journal of Artificial Intelligence Research 11* (1999) pp. 95-130.
[6] Csúcs, G. et al., "Peer to Peer Evaluation in Topologies Resembling Wireless Networks. An Experiment with Gnutella Query Engine", *ICON2003: The 11th IEEE International Conference on Networks* (Sydney, 2003) pp. 673.
[7] H. Assadi, "Construction of a Regional Ontology from Text and Its Use within a Documentary System International Conference on Formal Ontology and Information Systems", *FOIS-98, IOS Press, Amsterdam* (WebDB-2000), Springer-Verlag, Berlin, 2000, pp. 60–71.
[8] J.-U. Kietz, A. Maedche and R. Volz, "Semi-Automatic Ontology Acquisition from a Corporate Intranet", *Proc. Learning Language in Logic Workshop* (LLL-2000), ACL, New Brunswick, N.J., 2000, pp. 31–43.
[9] The Napster homepage, http://www.napster.com
[10] K. Sripanidkulchai, B. Maggs, H.Zhang, "Efficient content location using interest-based locality in peer-to-peer systems", *Infocom, 2003.*
[11] Joseph S., "P2P MetaData Search Layers", *Second International Workshop on Agents and Peer-to-Peer Computing* (AP2PC 2003).
[12] Marcello W Barbosa, Mellssa M Costa, Jussara M Almeida, Virgilio A P Alameida, "Using Locality of reference to improve performance of peer-to-peer applications". *WOSP'04 & ACM SIGSOFT Software Engineering Notes* V29n1(Jan 2004), pp.216-227.
[13] The WordNet project homepage, http://www.cogsci.princeton.edu/~wn/
[14] Bertalan Forstner, Gergely Csúcs, Kálmán Marossy, "Evaluating performance of peer-to-peer protocols with an advanced simulator", *Parallel and Distributed Computing and Networks*, 2005, Innsbruck, Austria.
[15] Bertalan Forstner, Hassan Charaf, "Neighbor Selection in Peer-to-Peer Networks Using Semantic Relations", *WSEAS Transactions on Information Science & Applications, 2(2),* February 2005. ISSN 1790-0832, pp. 239-244.
[16] Paul Erdős, Alfred Rényi, "On the Strength of Connectedness of a Random Graph", *Acta Math. Acad. Sci. Hungary, 12,* 1961, pp. 261-267.
[17] Mark S. Granovetter, "The Strength of Weak Ties", *American Journal of Sociology, 78* (1973), pp. 1360-1380.
[18] D. J. Watts, S. H. Strogatz, "Collective Dynamics of 'Small-World' Networks", *Nature, 393* (1998), pp. 440-442.
[19] Wolfgang Nejdl et al, "Super-peer-based routing and clustering strategies for RDF-based peer-to-peer networks", *20th International Conference on World Wide Web*, Budapest, Hungary, May 2003. ISBN: 1-58113-680-3, Pp. 536-543.
[20] Wolfgang Nejdl, Wolf Siberski, Michael Sintek, "Design issues and challenges for RDF- and schema-based peer-to-peer systems", *ACM SIGMOD Record, 32(3),* September 2003.
[21] D. A. Menasc´e, V. A. F. Almeida, and L. W. Dowdy, "Capacity Planning for Web Services: metrics, models, and methods". Prentice Hall, 2001.
[22] Z. Ge, D. R. Figueiredo, S. Jaiswal, J. Kurose, and D. Towsley, "Modeling Peer-Peer File Sharing System" In Proceedings of INFOCOM 2003, San Francisco, USA, Apr 2003.
[23] Mihajlo A. Jovanovic, "Modeling Large-scale Peer-to-Peer Networks and a Case Study of Gnutella" MsC Thesis.