PROSA: P2P Resource Organisation by Social Acquaintances

Vincenza Carchiolo, Michele Malgeri, Giuseppe Mangioni, and Vincenzo Nicosia

Dipartimento di Ingegneria Informatica e delle Telecomunicazioni
Facoltà di Ingegneria – Università di Catania
Viale A. Doria 6 – 95100 Catania, Italy
{car,malgeri,gmangioni,vnicosia}@diit.unict.it

Abstract. P2P overlay networks have been deeply studied in the last few years. The main problems of such networks are resources distribution and retrieving. In this paper ***PROSA*** is presented. It is based on a novel adaptive algorithm to build an efficient and semantically searchable P2P system. This algorithm is inspired by human relationships, since social communities possess some interesting properties (such as being "small–worlds") that allow fast and efficient routing of queries for resources.

1 Introduction

Peer-to-Peer (P2P) systems are computer networks where all hosts have the same functionalities and role. In P2P networks there is no difference between "client" hosts and "servers": a peer acts as a "client" host if it requests a resource from the network, and it acts as a "server" if it is requested a resource it is sharing. From this point of view, P2P networks differ a lot from Internet and, in general, from client–server networks.

In the last years the interest for overlay P2P networks has increased, mainly because bandwidth, computing power and cheapness of personal computers allow to implement such kind of "logic" networks. Examples of overlay networks include Gnutella, Freenet [1], CAN [2], Tapestry [3]. Each of them focuses on a particular aspect of P2P computing: Gnutella is totally unstructured, Freenet is practically anonymous, CAN is search–efficient and so on.

Some P2P structures proposed till now face the problem of efficient resources retrieval. In particular one of the more desirable feature in a P2P network is the possibility to perform query based on semantic resource description. Semantic queries are interesting because they are similar to the natural way a user describe concepts.

In unstructured networks, such as Gnutella, semantic query for resource can be performed, but for each request most part of the network is flooded, and there are no response guarantees either if the requested resource is present ([4]). In networks organised as Distributed Hash Tables (DHT) [1][2][3] semantic queries are not allowed, since resources are described by a certain hash of their content or description, so no "semantic proximity" can be neither defined nor used to discover them.

Some recent works [5][6] proposed to organise a P2P network in semantic groups of "similar" peers, to facilitate resource search and retrieval based on semantic queries.

S. Joseph et al. (Eds.): AP2PC 2006, LNAI 4461, pp. 135–142, 2008.

Our attempt is to define a P2P structure in which semantic proximity of resources is mapped onto topological proximity of peers. We propose a P2P network named ***PROSA*** inspired by social relationships and their dynamics, because social networks characteristics can be exploited to optimise query forwarding and answering. The paper is organised as follows: in Section 2 we point out some interesting aspects of social networks; in Section 3 we show how social relationships arise and how can be used to speed up information retrieval; in Section 4 we discuss our proposal and Section 5 presents a plan for future work.

2 Social Relationships and Small–World

The way social contacts and relationships are arranged, how they evolve and how they end, is matter for psychologists and social scientists research. Nevertheless some studies about social groups and their connections reveal that a "social network", i.e. the network of relationships among people from simple acquaintance to friendship, has many interesting properties that can be exploited in a real–world P2P structure. The Milgram experiment of 1966 [7] showed that a message from a "source" to a "destination" person can be delivered by forwarding it step–by–step to just one of the related people, in the direction of the destination. This experiment opened the research in the field of "small–world" networks [8]. A small–world network presents both small network diameter (i.e. the maximum distance, in number of hops, between two generic nodes of the network) and high clustering degree (i.e. good connections among similar or related nodes). The small–world property seems to be a characteristic of many human communities, such as mathematicians, actors, scientists. Our target is to develop a P2P system using rules and concepts inspired by human behaviours and relationships dynamics.

3 The Social Model

At the beginning of his life, a child has a small number of "social connections": his relatives. These contacts are the only interface between the baby and the outside world and are sufficient to a baby to grow. When a child goes to school, he is introduced to his teachers and class-friends. These relations are new "social links". We can call them "acquaintance–links". Having an acquaintance–link with somebody requires simply to know him. Naturally, not all social links have the same importance: if a child needs to solve a mathematic problem he will probably ask help to his math teacher or to the top student of the class (both of them being acquaintance–links), not to a randomly chosen person. Since the top student of the class probably can be useful in solving math problems, he becomes a "semantic–link" in the field of math for our child. Note that a semantic–link is not symmetric because the child knows that the math teacher is an expertise in the field of math (he solved some problems the child was not able to solve), but the teacher considers the child no more than a person that is probably interested in math (a simple acquaintance–link!): if he is not able to solve a math problem, he will not ask the child, but probably a colleague.

It is clear that a semantic–link is more than a simple acquaintance–link: having a semantic–link with somebody requires at least an acquaintance–link plus some

additional information about his interests, culture, abilities, knowledge etc. In real life no great effort is needed in order to establish a semantic–link with somebody: you have just to share a knowledge field or a passion or simply an interest with a person and meet him in some circumstances, have a talk with him and no more. Once you know somebody shares a certain knowledge or passion with you, a semantic–link in that field with that person is established and you're ready to use that link the next time you need information, help, assistance or collaboration in that field. In real life we massively use semantic–links to speed up information retrieval. ***PROSA*** uses the social model as a reference to build an efficient small–world semantic–searchable P2P network, exploiting the power of social links.

4 Building a Social P2P Network

In a P2P system the performance of searching and retrieving resources is heavily dependent on the organisation of the network.

Our target is to create a P2P network based on acquaintance– and semantic–links, where peers join the network in a way similar to a "birth", then achieve more links to other peers according to the social model, i.e. by linking (semantically) with peers which have similar interests, culture, hobbies, works and so on, and maintaining a certain number of "random" acquaintances. If ***PROSA*** catches the dynamics of the social model, the resulting network should be a small–world. To implement such a model we need i) a system to model knowledge, culture, interests, and ii) a network management algorithm as much as possible similar to the social model.

4.1 Modelling Knowledge

In ***PROSA*** , knowledge (each resource shared by peers) is represented using the Vector Space Model (VSM) . In this approach each document is represented by a state–vector of (stemmed) terms called Document Vector (DV); each term in the vector is assigned a weight based on the relevance of the term itself inside the document. This weight is calculated using a modified version of TF–IDF [9] schema, as follows:

$$w_t = 1 + \log(f_t) \tag{1}$$

where f_t is the term frequency in the document. It has been proved [10] that this way of calculating relevance is a good approximation of TF–IDF ranking schema. The VSM representation of a document is necessary to calculate the relevance of a document with respect to a certain query. We model a query by means of a so–called Query Vector (QV), that is the VSM representation of the query itself. Since both documents and queries are represented by state–vectors, we define the relevance of a document (D) with respect to a given query (Q) as follows:

$$r(D,Q) = \sum_{t \in D \cap Q} w_{t,D} \cdot w_{t,Q} \tag{2}$$

Using VSM we obtain also a compact description of a peer knowledge. This description is called "Peer-Vector" (PV), and is computed as follows:

- For each document hosted by the peer, the frequencies of terms it contains are computed ($F_{t,D}$).
- Terms frequencies for different documents are summed together, obtaining overall frequency for each term:

$$F_t = \sum_t F_{t,D}$$

- Then a weight is computed for each term, using:

$$w_t = 1 + \log(F_t)$$

- Finally all weights are put into a state–vector and the vector is normalised.

The obtained PV is a sort of "snapshot" of the peer knowledge, since it contains information about the relevant terms of the documents it shares. The relevance of a peer (P) with respect to a given query (Q) is defined as follows:

$$r(P,Q) = \sum_{t \in P \cap Q} w_{t,P} \cdot w_{t,Q} \tag{3}$$

This relevance is used by the ***PROSA*** query routing algorithm. It is worth noting that a high relevance between a QV and a PV means that probably the given peer has documents that can match the query.

VSM is an effective way to represent a peer knowledge. If we take a look at a typical DV, we can see that it gives an idea of the corresponding document. For example in figure 1 the DV corresponding to the manual page of the Unix command "mount" is shown.

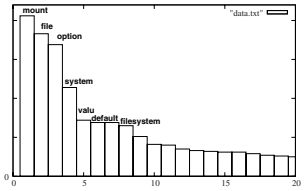


Fig. 1. A sample DV

You can see that the most relevant (stemmed) terms are: mount, file, option, system, valu, default, filesystem. Just six or seven terms give a precise idea of what kind of document we are dealing with. On the other hand if a user is searching for help about the "mount" command, he will probably build a query containing these terms, for example "mount default filesystem"; if a user is searching info about compiling sources with gcc, his query could look like "gcc compile source". The relevance of the first query with the "mount" manual page is about 0.014. The relevance of the second query is zero, since the document doesn't contain those terms (and it is not related with compiling source code!). So we can argue that the VSM is a good choice to rank resources with respect to a given query.

4.2 Managing Connections

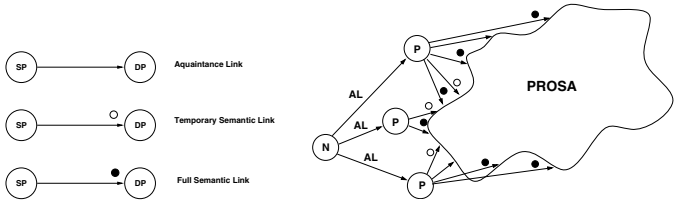
In ***PROSA*** we want to use some principles inspired by observations about natural evolution of social groups. In particular we want to simulate the way people "link" to other people. As stated above, relationships among people are usually based on similarities in interests, culture, hobbies, knowledge and so on. And usually these kind of links evolve from simple "acquaintance–links" to what we called "semantic–links". To implement this behaviour three types of links are introduced: i) Acquaintance–Link (AL) ii) Temporary Semantic–Link (TSL) iii) Full Semantic–Link (FSL). TSLs represent relationships based on a partial knowledge of a peer. They are usually stronger than ALs and weaker than FSLs.

Since relationships are not symmetric (remember the case of the child and the teacher), it is necessary to specify what are the source peer (SP) and destination peer (DP) of a link. Figure 2(a) shows the representations for the three different types of links.

Remembering links. To efficiently use the right link in any given situation, each peer maintains a list of known peers, that we call Peer List (PL). Each entry of the PL contains two fields: an address and a vector. For example, if the network overlays a TCP/IP network, the address of the linked peer is the couple IP address/TCP port. If the link is a simple AL, the peers doesn't know the corresponding PV: in this case an empty PV is placed into the vector field. If the link is a TSL, then the peer doesn't know the PV of the linked peer, but a Temporary Peer Vector (TPV) is built based on the query received in the past from that peer. Finally, if the link is a FSL, the PV is put in the vector field.

A new peer was born. A new peer which wants to join ***PROSA*** , just searches other peers (for example using broadcasting, or by selecting them from a list of peer that are supposed to be up, as in Freenet or Gnutella) and adds some of them in his PL as ALs. The joining phase is represented in figure 2(b), where "N" is the new peer; N chose some other peers (P) at random as initial ALs. These peers are connected, via ALs, TSLs or FSLs to other peers into ***PROSA*** , and allow N to start forwarding queries until it meets other peers.

Links dynamics. In ***PROSA*** links dynamics are strictly related to queries. When a user of ***PROSA*** requires a resource, he performs a query and specify a certain number



(a) Link Types (b) A new node joining ***PROSA***

Fig. 2.

of results he wants to obtain. The relevance of the query with the resources hosted by the user's peer is first evaluated, using equation 2. If none of the hosted resources has a sufficient relevance with respect to the query, the query has to be forwarded to other peers. The mechanism is quite simple:

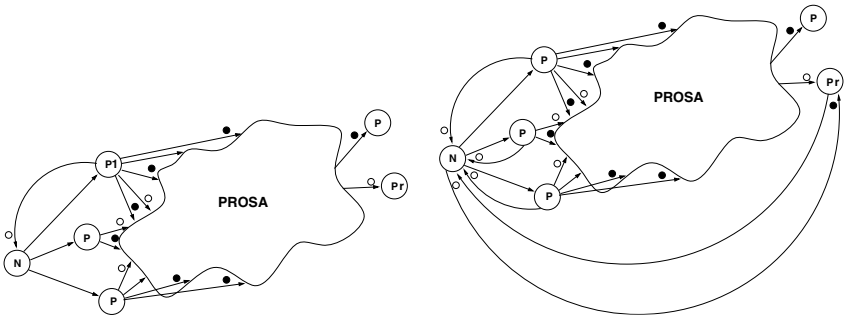
- A query message containing the QV, a unique QueryID, the source address and the required number of results is built.
- If the peer has neither FSLs nor TSL, i.e. it has just AL, the query message is forwarded to one link at random.
- Otherwise, the peer computes the relevance between the query and each entry of his Peers–List.
- It selects the link with a higher relevance, if it exists, and forward the query message to it.

When a peer receives a query forwarded by another peer, it first updates its PL. If the requesting peer is an unknown peer, a new TSL to that peer is added in the PL, and the QV becomes the corresponding Temporary Peer Vector (TPV). If the requesting peer is a TSL for the peer that receives the query, the corresponding TPV in the list is updated, adding the received QV and normalising the result. If the requesting peer is a FSL, its PV is in the PL yet, and no updates are necessary. [1] After PL update, the relevance of the query and the peer resources is computed. There are three possible cases:

- No document has a sufficient relevance. In this case the query is forwarded to another peer, according to link relevance.
- The peer has a certain number of relevant documents, but they are not enough to full-fill the request. In this case a response message is sent to the requester peer, specifying the number of matching documents and the corresponding relevance. The message query is forwarded to all the links in the PL whose relevance with the query is higher than a given threshold (semantic flooding). The number of matched resources is subtracted from the number of total requested documents before forwarding.
- The peer has sufficient relevant documents to full-fill the request. In this case a result message is sent to the requesting peer and the query is no more forwarded.

This situation is showed in figure 3(a), where peer "N" forwards a query to one of his ALs randomly chosen, since it has niether TSLs nor FSLs. In our example the chosen peer is "P1". As soon as P1 receives the QV, it automatically establish a TSL with N (see figure 3(a)) and then it forwards the query if needed. When the requesting peer receives a response message it presents the results to the user. If the user decides to download a certain resource from another peer, the requesting peer contacts the peer owning that resource and asks it for download. If download is accepted, the resource is sent to the requesting peer, together with the Peer Vector of the serving peer. This case is illustrated in figure 3(b), where peer "N" received a response from peer "Pr"

[1] In ***PROSA*** a TPV is similar to a "hint"; the assumption made here is that a peer querying for a certain resource would eventually find it, and could successfully answer similar queries in the future. So it makes sense to save a weak link to a querying peer, since that link could be useful to answer future queries.



(a) Query forwarding: new TSL arise (b) Query forwarding: new FSL arises

Fig. 3.

and decided to download the corresponding resource. Note that Pr established a TSL with N, because it received a QV from it, and N established a FSL with Pr, because it successfully received a resource from it.

4.3 Comments on the Algorithm

The algorithm presented in Sections 4.1 and 4.2 is an attempt to model human relationships and their behaviour in a P2P system.

The network management system used in ***PROSA*** , allows links to move from simple acquaintance to weak relationship, and the algorithm proposed works in a way similar to relationships dynamics in real world. A Temporary Peer Vector can be considered as a partial description of a person you don't know very well. It's just an approximation, but is is better than nothing. It is also worth noting that the proposed algorithm allows the growth of fuzzy "semantic groups". A semantic group is a group of peers with similar knowledges. It is the algorithmic transposition of "social groups". In real life people may belong to different social groups, according to their interests. Peers that are "interested" in a particular topic, usually perform query in that topic. When they receive responses, they acquire new semantic links to peers sharing resources belonging to that topic. This is quite similar to "moving" in the direction of the semantic group made of all the peers sharing that kind of knowledge. On the other hand, if a peer changes his interests (i.e. if different topics are required) it smoothly "discards" links to unwanted topics, because the size of the PL is limited: if new semantic links are put into the PL, the old ones will be gradually pushed out. The PL represent the current "social" state of a peer, a "snapshot" of the semantic groups he belongs to.

5 Conclusions and Future Work

In this paper a novel adaption algorithm for P2P system organisation has been presented. The algorithm is heavily based on observation of the social world. In particular it emulates the way social relationships among people naturally arise and evolve. Our hope is that the resulting system could present some of the desirable properties of social

communities, in particular the "small–world" characteristic, which is peculiar of social groups and allow efficient routing and high clustering.

The next step is to develop a simulator of ***PROSA*** to test the described algorithm. In particular we are going to check if similar peers are clustered together to form "semantic groups" and "social communities". Another interesting research is measuring the quality of responses to query, in terms of both quantity and relevance. We are also going to introduce weighted links among peers, since not all social relationships have the same relevance in a person life, and this can heavily impact on the quality of social groups.

References

1. Clarke, I., Sandberg, O., Wiley, B., Hong, T.W.: Freenet: A distributed anonymous information storage and retrieval system. In: Federrath, H. (ed.) Designing Privacy Enhancing Technologies. LNCS, vol. 2009, Springer, Heidelberg (2001)
2. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content addressable network. Technical Report TR-00-010, Berkeley, CA (2000)
3. Zhao, B.Y., Kubiatowicz, J.D., Joseph, A.D.: Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, UC Berkeley (2001)
4. Loo, B., Huebsch, R., Stoica, I., Hellerstein, J.: The case for a hybrid p2p search infrastructure. In: Voelker, G.M., Shenker, S. (eds.) IPTPS 2004. LNCS, vol. 3279, Springer, Heidelberg (2005)
5. Bawa, M., Manku, G.S., Raghavan, P.: Sets: search enhanced by topic segmentation. In: SIGIR 2003: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pp. 306–313. ACM Press, New York (2003)
6. Zhu, Y., Yang, X., Hu, Y.: Making search efficient on gnutella-like p2p systems. In: Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International, pp. 56a–56a. IEEE Computer Society, Los Alamitos (2005)
7. Milgram, S.: The small world problem. Psychol Today 2, 60–67 (1967)
8. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature 393, 440–442 (1998)
9. Salton, G., Buckley, C.: Term weighting approaches in automatic text retrieval. Technical report, Ithaca, NY, USA (1987)
10. Schutze, H., Silverstein, C.: A comparison of projections for efficient document clustering. In: Prooceedings of ACM SIGIR, Philadelphia, PA, pp. 74–81 (1997)