A Decentralized Recommendation System based on Self-Organizing Partnerships

Giancarlo Ruffo, Rossano Schifanella, and Enrico Ghiringhello*

Dipartimento di Informatica, Università di Torino
Corso Svizzera, 185 - 10149, Torino (ITALY)
{ruffo,schifane}@di.unito.it

Abstract. Small World patterns have been found in many social and natural networks, and even in Peer-to-Peer topologies. In this paper, we analyze File Sharing applications that aggregate virtual communities of users exchanging data. In these domains, it is possible to define overlaying structures that we call "Preference Networks" that show self organized interest-based clusters. The relevance of this finding is augmented with the introduction of a proactive recommendation scheme that exploits this natural feature. The intuition behind this scheme is that a user would trust her network of "elective affinities" more than anonymous and generic suggestions made by impersonal entities.

Key words: Peer-to-Peer, Recommendation Management, Small World Networks, Social Networks

1 Introduction

Even if File-Sharing is not the only application of the peer-to-peer paradigm, it is indeed a unique environment where (a subset of) social attitudes of p2p users can be studied and deeply observed. This is mainly due to the huge popularity of tools like Gnutella, eMule, bitTorrent, and so on, that every day attract millions of users that share several terabytes of electronic information. Even if maybe nobody is able to extract generalizable applicative consequences from phenomena observed in a particular *ecosystem*, it is indeed true that the in depth understanding of the way users strictly cooperate for reaching their individual aims has a significant scientific relevance. Furthermore, the new knowledge about a given community, that uses a particular kind of application, can really be exploited for improving the application itself, and even for improving other applications based on the same paradigm.

Many file sharing users know very well how much is difficult to find something interesting just picking up, at random, another user's file list. This is mainly due to the fact that every person has different likings, and an item, interesting for one user, can be absolutely detestable for someone else. Conversely, when the search process is iterated on different queries in an unstructured network (e.g., Gnutella, FastTrack, and so on), it is quite surprisingly to observe that there is a constant core in the set of responding

* E. Ghiringhello joined this project during the preparation of his bachelor thesis.

users. In fact, many file sharing clients allow the participant of a network to create lists of *favourite users*, in order to directly explore the given shared file systems that are more likely to return something interesting to the querying user.

During the years, many social networks have been discovered and analysed, e.g., the scientific co-authorship network [1], the friend-ship network [2], and so on. In this paper, we want to analyse a particular instance of the *Preference Network*, that links users sharing common interests. In other words, we create a (family of) network(s) whose nodes are users of a file sharing system, and whose links connect pair of nodes that share one or more identical files. The main purpose of this study is to empirically prove that a Preference Network has a *Small World* topology [3, 4]. As a relevant consequence, we want to propose an applicative framework, where the small world property of the preference graph in unstructured p2p networks, can be exploited to return a decentralized recommendation service to the user.

2 Related Work and Road Map

This paper contains two contributions: (1) we introduce a family of interest based graphs on top of Gnutella, which connect users sharing at least m common files, and proving that they show small world topologies; (2) a practical recommendation scheme is proposed to the file sharing community, that takes advantage of the high clustering coefficient of the previously introduced Preference Networks.

The definition of preference networks is somehow similar to the one given for *data-sharing graphs* in [5, 6], with a subtle, but decisive, difference: two users are connected in a preference network when they are storing (and sharing) replica of at least m common files - uniquely identified by means of a hash code. Conversely, two users are connected in a data-sharing graph if they (try to) download at least m common objects during a time interval T - identified by way of their file names. Hence, the two structures differ (1) at scope level, (2) at data collection level, and (3) at temporal level. First of all, we are interested in the tastes of the users, and we want to connect people that have similar likings. Secondly, Leibowitz and al., as reported in [5], collected data from processing HTTP logs at a large Israeli ISP. They focused on traffic generated by KaZaa clients, and they refer to file names to find out if different users were downloading the same items. However, this does not capture some phenomena that are relevant in file sharing systems, such as the presence of fake files (i.e., items having names not matching their contents), the rapid downloading-deleting process (i.e., very often a user downloads a file and suddenly she deletes it because she realizes that it is out of interest), and the possibility for a file to have many replica with different names. For these reasons, we preferred to collect *QueryHit* [7] messages, stored by running for several days a (modified) Gnutella Ultra-Peer: these messages contain the precise information of (some of) the files actually shared by the network (with the hashed file identifier, too). Finally, we deliberately did not consider any temporal constraints, because we want to study persistent phenomena. In fact, we are making the assumption that if a user is still sharing a file, then she directly inserted it in the network, otherwise she previously downloaded it. In the first case, the user is trivially interested in that kind of content. In the second case, if the user downloaded it and she did not delete the file

immediately after, we can reasonably assume that she is interested in it. Observe that we are consciously ignoring transient events featuring in data sharing, because in this phase of the study, we are not interested in how the snapshots of the network change in different instances of time: if a user is found showing a preference for a given item, then he will maintain an interest for it (e.g., if he likes the Beatles classic "Yesterday", then he will very likely love that song even in the future). On the contrary, the domain investigated in [5] is highly dynamic, and it is subject to change very rapidly. If the reader is interested to characterize dynamic phenomena in unstructured topologies, then she needs a parallel analysis, as explained in [8].

Preference Networks are, indeed, very similar to the structures described in [9]. In that study, the inefficient Gnutella search mechanism based on flooding is enhanced by means of interest based localities. Our paper adds two contributions to that work: first of all we experimentally prove that preference networks are small worlds. This result is generalizable outside Gnutella, because it is not related to the file sharing network topology or to the given search mechanism. Secondly, we propose a recommendation scheme that suggest users the next likely interesting files, without concerning about the localization of the item, because this can be performed with many known efficient solutions (e.g., by way of a Distributed Hash Table based algorithm).

Recommendations and Trust Management are fertile areas in the Peer-to-Peer scientific community [10]. Differently to previous work in this field, our recommendation scheme acts *proactively* pushing automatic suggestions to the user, presenting her an *unseen* file. This point is somehow related to many e-commerce services, that assist the user with sentences like "Customers who bought this CD also bought: The Rolling Stones - Aftermath". In fact, to our knowledge, this is the first proactive recommendation proposal that works in a complete distributed domain without any central repository nor any common background knowledge. Suggestions are made uniquely on the basis of natural and self organizing users' partnerships. Like in the real world, even in virtual social communities, people can meet others with common interests and trust them by word of mouth before buying or looking for items.

Sections 3 and 4 will be devoted to define preference networks and to provide an empirical proof of the small world features of such graphs. The recommendation scheme is introduced in Section 5. Section 6 reports some conclusion and the agenda of our ongoing work.

3 Graphs, Topologies and Preference Networks

Let $G = (V, E)$ be a graph, where V and E are respectively the set of vertices and the set of edges between nodes. Let L be the *average shortest path length*, and let C be the *clustering coefficient* of G. The clustering coefficient of a graph gives a measure of how many almost complete sub-graphs are in the topology. In fact, given $v_i \in V$, the clustering coefficient C_i is the ratio of the actual number of edges between neighbors of v_i for the maximum value of such a number. The clustering coefficient C of the graph is the mean value of all the C_is. Formally speaking, if we define the set of *neighbors* of v_i as $V_i = \{v_j\} : v_j \in V, e_{ij} \in E$, then the *degree* of v_i is $d_i = |V_i|$, i.e., d_i is the number of neighbors of the vertex. Note that D_i, the maximum number of links

between neighbors of v_i, can be defined in function of d_i; in fact, if G is a directed graph (i.e.,$e_{ij} \neq e_{ji}$), then $D_i = d_i \cdot (d_i - 1)$. Otherwise (when G is undirected), $D_i = \frac{d_i \cdot (d_i - 1)}{2}$. Let $E_i = \{e_{jk}\} : v_j, v_k \in V_i, e_{jk} \in E$ be the actual set of edges between neighbors of v_i. Hence, the *clustering coefficient* of v_i can be defined as:

$$C_i = \frac{|E_i|}{D_i}.$$

Observe that if C_i is equal to 0, it means that the neighbors of v_i are not connected each other (i.e., $E_i = \emptyset$).
Otherwise, if $C_i = 1$, then the sub-graph G_i is complete, where $G_i = (V_i \cup \{v_i\}, E_i \cup \{e_{ij} : e_{ij} \in E\})$.

Furthermore, the *clustering coefficient of graph* G is defined as in [3]:

$$C = \frac{\sum_i C_i}{|V|}.$$

Even if Newmann [1] describes in a different manner the clustering coefficient, we recall that both definitions bring to comparable results. Therefore, in the following lines the reader should remember that the previous definitions were used during our analysis, and that a different clustering definition would not affect the given conclusions.

We can intuitively think at a small world graph, as a loosely connected network of (almost) complete sub-graphs. Hence, a graph G is checked against this property by comparison with a random graph G_{rand} with the same number of vertices and edges. Let L_{rand} and C_{rand} be respectively the average shortest path length and the clustering coefficient in the random graph, we say that G is *small world* if $C \gg C_{rand}$ and $L \approx L_{rand}$.

In order to further model our domain, let us assume that a set of users $U = \{u_1, u_2, \ldots, u_n\}$[1] is sharing a set of items $S = \{s_1, s_2, \ldots, s_l\}$. We map users to items with function $f : U \to \mathcal{P}(S)$, where $\mathcal{P}(S)$ is the power set of S. Of course, $\bigcup_{i=1}^{n} f(u_i) = S$. Moreover, in our environment, we assume that for some i and j, $f(u_i) \cap f(u_j) \neq \emptyset$, i.e., users may share (some) identical files. Observe, that this assumption is realistic in an unstructured overlay network, where *users share what they want to*.

These hypotheses enable us to introduce the concept of a *preference network*, that we can model with a graph where each vertex corresponds to a different user, and a link is connected between two users that share at least m items; i.e., $G^m = (U, E^m)$, where $e_{ij}^m \in E^m \Leftrightarrow |f(u_i) \cap f(u_j)| \geq m$.

In next section, we want to show that there is a natural partnership between users. In particular, we found that preference networks with $2 \leq m \leq 8$ have a small world topology, meaning that we can assume a transitivity property between users: let u_a and u_b be two users sharing at least m common files, and let u_c be a user that shares at least m files with u_b, then it is very likely that u_a and u_c share at least m files. This would be just a corollary of the small world property, because if G^m is small world, than it will

[1] In the rest of the paper, we will assume a bijection between users and nodes of the p2p file sharing network. Hence, we can use u_i to indicate the i-th node as well as the i-th user.

have a clustering coefficient C with a high value. This *triangulation* between users is very relevant to our study, because these self-organizing communities can be exploited to a very efficient and fully decentralized recommendation scheme, as described in Section 5.

4 Data Collection

In order to study the small-world properties of the so defined *preference networks*, we perform the following preliminary steps: (1) implementation of a Gnutella network crawler, (2) data collection, (3) post-processing of the gathered traces, and (4) generation of the preference graphs.

As described above, we focus on persistent phenomena of data-sharing relationships between users, hence we are interested in tracing *QueryHit* messages exchanged between peers. The modern Gnutella network topology consists in a *two-tier* overlay where a set of interconnected *ultrapeers* forms the top-level overlay to which a large group of *leaves* are connected. Leaves never forward messages: they send queries to the ultrapeers and wait for a set of *QueryHit*s matching the searching criteria. Otherwise, an ultrapeer acts as a *proxy* to the Gnutella network for the leaves connected to it. Ultrapeers are connected to each other and to *regular* Gnutella hosts. *QueryHit* messages return back to the querying user by reverse path forwarding. This ensures that only those servents that routed the *Query* message will get the returning *QueryHit* message. Therefore, a *ultrapeer* receives all *QueryHit* messages addressed to its leaves.

To hit the mark, we have modified the Gnutella servent $Phex$ [11][2], an open-source client written in Java language. Our crawler is forced to access the network in the *ultrapeer* mode, and to trace down all the *QueryHit* messages it stores and forwards. Collected data contains information about the user that answers the query and the related resources he shares. Each user is identified by the IP address, whereas the hash code of the resource is exploited to unambiguously identify the file. The crawler collects a

CHARACTERISTICS OF TRACES COLLECTED

Time Interval	7 days
# Users (distinct IP)	136.752
# Files (distinct SHA)	473.848
# Query Hits	1.601.610
# Query Hits (mp3 songs)	798.821

Table 1. Data collected by the Gnutella ultrapeer crawler from 19 October to 26 October 2005.

seven days of Gnutella traffic (from 19 October to 26 October 2005). As described in Table 1, the traces are composed by more than 1.5 millions data entries generated by about 130.000 distinct IP addresses and involving 473.848 different SHA-1 file hashes.

[2] We used the Phex version 2.6.4.89

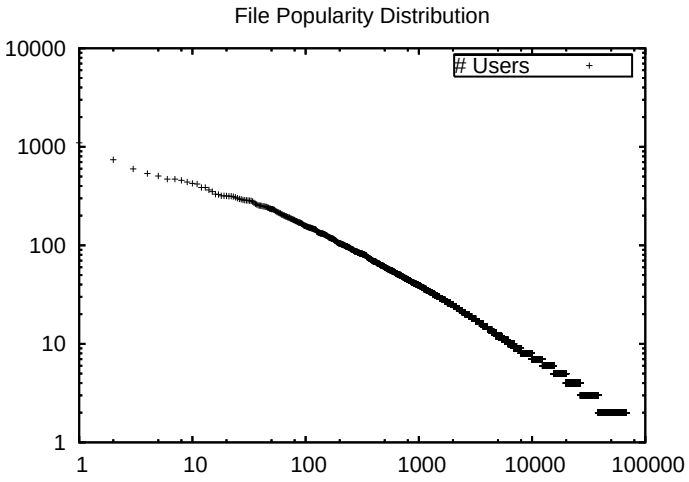

Fig. 1. File popularity distribution, plotted in a log-log scale, follows Zipf's law.

Notice that each *QueryHit* message can contain more than one reference to resources matching the search criteria[3]. Figure 1 shows the file popularity distribution observed in our Gnutella snapshot. Let us notice that it follows Zipf's law, as already observed in [6].

After the raw data is collected, we apply a filtering phase composed by the following steps:

– *Private network IP address filtering:* our model binds each IP address to a distinct user. Indeed, it is possible that the same IP address corresponds in fact to different users, e.g. shared workstations or presence of NAT/proxy. A private network environment provides a concrete example of this effect: let us suppose that the IP $192.168.1.10$ publishes a set of resources R. We could relate the IP $192.168.1.10$ to a particular user u and we could wrongly assert that u shares the files belonging to R. In fact, many distinct users in different networks can obtain this address, so that the *QueryHit* content cannot distinguish between these users. The effect is the presence of distinct IPs that seem to share large sets of files, affecting the fairness of the preference graphs [4]. To get rid of this effect, we filter out all IP addresses that belong to the private network class specification[5].
Notice that the opposite phenomenon can be observed as well. For example, in a DHCP-based network the same user can obtain different IP addresses in distinct sessions. Therefore, a set of resources R that effectively belong to a user u, can

[3] We found that all the received *QueryHit* messages contained at most five query results.

[4] Indeed, these IPs behave like hubs, so they should amplify the small-world properties showed by the preference graphs.

[5] We filter out the following sets of IP addresses[12]: $10.x.x.x$, $192.168.x.x$ and the range from $172.16.0.0$ to $172.31.255.255$.

be seen as sum of shared items from many users. Obviously, this phenomenon can smooth the hub behavior of the user u. However, we think that this effect does not impact our study due to the relatively short time of trace collection[6].

– *Focusing on MP3 songs:* in our evaluation we consider only entities related to mp3 song files. Filtering out other content types allows a better analysis about user preferences relationships in a specific field, e.g. the music likings domain. Moreover, the cleaned dataset reduces the complexity inherent to the graphs generation task. However, Table 1 shows that about 50% of the overall resources were mp3 songs.

G^m	#	#	Preference Graph		Random Graph	
	Nodes	Edges	L	C	L_{rand}	C_{rand}
G^2	22777	428931	3.29	0.43	3.418	0.0017
G^3	9807	81088	3.53	0.37	4.351	0.0017
G^4	4779	23378	3.68	0.35	5.336	0.0020
G^5	2612	8519	3.81	0.33	6.655	0.0025
G^6	1501	3617	3.93	0.28	8.316	0.0032
G^7	891	1780	4.12	0.27	9.815	0.0045
G^8	591	990	4.4	0.25	12.371	0.0057

Table 2. *Average shortest path length* L and *clustering coefficient* C for the preference networks.

After the filtering step, we generated the preference graphs G^m, where nodes are users and a link connects two users that share at least m items (see Section 3). We created several graphs, from G^2 to G^8, and for each of them we computed the average shortest path length (L) and the clustering coefficient (C). These metrics are estimated also for a random graph with identical number of nodes and edges (L_{rand} and C_{rand}). As Table 2 shows, all G^m graphs reveal small-world patterns: in fact, we have that, for all the preference networks, $C \gg C_{rand}$ and $L \approx L_{rand}$ (very interestingly, it always happens that $L < L_{rand}$).

Notice that the data-sharing relationships collected by means of tracing *QueryHit* messages represents obviously only a fraction of the resources shared within the network. Therefore we reasonably think that a global vision could strongly confirm and enhance the small-world properties observed.

In Table 3 we compare the values of L and C with other known domains showing small-world phenomenons [13].

[6] We executed different monitoring sessions using the Gnutella servent's IDs to discriminate between users. These IDs are generated by running a cryptographic hash function on a random input value, so that it changes after each login. We did not observe relevant differences in the results w.r.t. to the ones obtained from the filtered data, thus enforcing our findings.

$Network$	L	C	$Reference$
WWW, site level, undir.	3.1	0.1078	Adamic, 1999
Movie actors	3.65	0.79	Watts and Strogatz, 1998
LANL co-authorship	5.9	0.43	Newman, 2001a, 2001b, 2001c
MEDLINE co-authorship	4.6	0.0666	Newman, 2001a, 2001b, 2001c
SPIRES co-authorship	4.0	0.726	Newman, 2001a, 2001b, 2001c
NCSTRL co-authorship	9.7	0.496	Newman, 2001a, 2001b, 2001c
Math. co-authorship	9.5	0.59	Barábasi et al., 2001
Neurosci. co-authorship	6	0.76	Barábasi et al., 2001
E. coli, substrate graph	2.9	0.32	Wagner and Fell, 2000
E. coli, reaction graph	2.62	0.59	Wagner and Fell, 2000
Ythan estuary food web	2.43	0.22	Montoya and Sole' , 2000
Silwood Park food web	3.40	0.15	Montoya and Sole' , 2000
Words, co-occurrence	2.67	0.437	Ferrer i Cancho and Sole , 2001
Words, synonyms	4.5	0.7	Yook et al., 2001b
Power grid	18.7	0.08	Watts and Strogatz, 1998
C. Elegans	2.65	0.28	Watts and Strogatz, 1998

Table 3. Example of C and L for several real networks.

5 A Recommendation Scheme Based on Preference Partnership

Users' preferences can be used for improving search mechanisms in overlay network, as suggested in [5], to enforce ontologies definitions and routing in semantic (p2p) networks [14, 15], and also to locate content avoiding expensive flooding search mechanisms [9]. We are interested to the highly informative power of self-organizing communities characterized by (almost) complete sub-graphs: users in the same cluster share each-other a subset of common items and are likely interested to other files popular in the cluster. The transitivity property may be used for enabling *reserved information lanes* between users, in order to announce items that are potentially of interests for members of the same cluster.

First of all, let us introduce a notation that we will use in the rest of the section. Given $u_x, u_y \in U$, and an item $s_k \in S$, we describes the event that u_x downloaded file s_k from u_y (or, similarly, u_y uploaded s_k to u_x) with $u_y \stackrel{s_k}{\rightarrow} u_x$. Of course, if $u_y \stackrel{s_k}{\rightarrow} u_x$, then $s_k \in f(u_x) \cap f(u_y)$ (even if the adverse implication is not always applicable).

For each user $u_i \in U$, we define:

$$F_0(u_i) = \{u_j : (i \neq j) \wedge \exists s_k (u_i \stackrel{s_k}{\rightarrow} u_j \vee u_j \stackrel{s_k}{\rightarrow} u_i)\},$$

that is the *set of contacts* of u_i. Roughly speaking, the node of user u_i maintains a list of other users that exchanged some files with u_i. In order to exploit the triangulation property of the preference networks which the node is connected to, we consider also the *set of contacts of the first order* of u_i:

$$F_1(u_i) = \bigcup_{u_j \in F_0(u_i)} F_0(u_j).$$

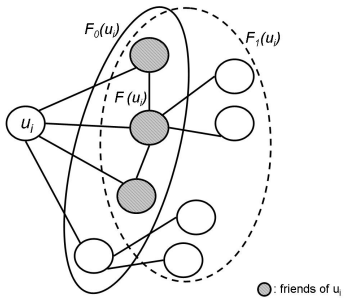

Fig. 2. Example: contacts and friends of u_i.

We introduce the *list of friends (or partners)* of u_i as it follows:

$$F(u_i) = F_0(u_i) \cap F_1(u_i).$$

Note that relationships in $F(u_i)$ are stronger than in $F_0(u_i)$ (see Figure 2).
The node u_i stores an integer value $m(u_i, u_j)$ for each reference in $F(u_i)$. More precisely, we define the *partnership degree* of the pair u_i and u_j as $m : U^2 \rightarrow \mathcal{N}^+$, where $m(u_i, u_j) = |f(u_i) \cap f(u_j)|$, that is the number of files that they have in common. For the sake of simplicity, in the rest of the section, we will use the notation m_{ij} instead of $m(u_i, u_j)$.

At an implementation level, list $F(u_i)$ has a constant size, and it is ordered on the basis of the value of the partnership degree m_{ij}: the user on the top of the list has more files in common with u_i than with the others. On the contrary, the less "interesting" user (e.g., $m_{ij} \approx 0$), is likely to be removed from the list. The reader should observe that, given $\overline{m}$, it is possible to extract, from this list, the (known) neighbors of u_i in the preference graph $G^{\overline{m}}$; in fact, it is easy to note that $m(u_i, u_j) = \overline{m} \Rightarrow u_j \in U_i^{\overline{m}}$, where $U_i^{\overline{m}}$ is the set of neighbors of u_i in $G^{\overline{m}}$.

For example, let us suppose that user u_x downloaded s_1 and s_2 from u_y. Moreover, he downloaded s_3, s_4 and s_5 from u_z, and s_6 from u_v. Finally, we have also that $F_0(u_x) = F_1(u_x)$. As a consequence, we have that: $f(u_x) = \{s_1, s_2, s_3, s_4, s_5, s_6\}$, and $F(u_x) = \{u_y, u_z, u_v\}$. Furthermore, after having interacted with u_y, u_z and u_v, the p2p client of u_x got also their file lists. So, u_x knows that $f(u_y) = \{s_1, s_2, s_4, s_7\}$, $f(u_z) = \{s_3, s_4, s_5, s_7, s_8\}$ and $f(u_v) = \{s_6, s_7, s_3, s_4, s_5\}$. The values of function m are updated after each interaction. After the last download, they are as it follows: $m_{xy} = 3, m_{xz} = 3$, and $m_{xv} = 4$. $F(u_x)$ is ordered as it follows: (u_v, u_y, u_z).

We need also to identify the set of files[7] owned by the friends of u_i, but that are not possessed by u_i:

$$\text{Co-f}(u_i) = (\bigcup_{u_j \in F(u_i)} f(u_j)) - f(u_i).$$

[7] Note that the p2p client of u_i does not store all the friends' files, but only an unique reference to them (e.g., their SHA-1 hashes).

The state of a running node includes also a *file map*, that returns the partners owning a given resource, that is not possessed by u_i. Hence, we define the family of functions:

$$map_i : \text{Co-f}(u_i) \rightarrow \mathcal{P}(F(u_i)),$$

where $map_i(s_k) = \{u_j \in F(u_i) : s_k \in \text{Co-f}(u_i) \cap f(u_j)\}$.
In the previous example, we have that Co-f$(u_i) = \{s_7, s_8\}$, $map_i(s_7) = \{u_y, u_z, u_v\}$, and $map_i(s_8) = \{u_z\}$.

5.1 *Die Wahlverwandtwchaften*: the intuition

The intuition behind the proposed recommendation scheme is based on the observation that friends of a given peer build a cluster of nodes with different partnership degrees. We previously observed in Section 4 that nodes can be naturally gathered together on the basis of common interests. Moreover, we noted that there are peers that are more kindred to some partners than others; in fact, we found that a preference network G^m is a small world, even with growing values of m. But not all the nodes involved in preference networks with lower degrees than m are still involved in G^m. Thus, some relationship between nodes in the same cluster is stronger than others: even in the file sharing community, *elective affinities* [16] rule the social behavior of the users.

We want to sort files in Co-f(u_i) by means of the following criteria:

1. *popularity* in the cluster of partners of u_i;
2. *partnership degree* of friends storing the missing files;

The *recommendation list* is defined as the ordered sequence:
$R(u_i) = (s_{k_1}, s_{k_2}, \ldots, s_{k_\ell})$, where $\ell = |\text{Co-f}(u_i)|$, and $\forall h = 1, \ldots, \ell : s_{k_h} \in$ Co-f(u_i). Files in $R(u_i)$ are sorted (and, hence, recommended), on the basis of the weight defined below:

$$w(s_{k_h}) = \frac{\sum_{u_j \in map_i(s_{k_h})} (m_{ij})}{max_d(|map_i(s_{k_d})|)},$$

i.e., $\forall s_{k_d} \in R(u_i) : w(s_{k_{d-1}}) \leq w(s_{k_d}) \leq w(s_{k_{d+1}})$.

In our example, files will be recommended to u_i in this order: (s_7, s_8). In fact, we have that $w(s_7) = 3.\bar{3}$, and $w(s_8) = 1.0$. Of course, in a practical environment, we can set a threshold, in order to filter out recommendations with low weight. In the previous case, if such a threshold is set to 2.0, only file s_7 would be submitted to the user's attention. The estimation of this value is one of the tasks of on-going work.

5.2 Discussion

Let us numerically quantify the popularity of a file and the average partnership degree of nodes hosting a given item as it follows:
Given a node u_i, the *popularity* of a missing file s_{k_h} is calculated by way of the family of functions pop$_i$: Co-f$(u_i) \rightarrow]0, 1]$, where

$$\text{pop}_i(s_{k_h}) = \frac{|map_i(s_{k_h})|}{max_d(|map_i(s_{k_d})|)}.$$

Given a node u_i, the *degree* of a missing file s_{k_h} as the average partnership degree of nodes in $F(u_i)$ that stores s_{k_h}. This value is calculated by way of the family of functions $\deg_i$: Co-f$(u_i) \rightarrow \mathcal{R}^+$, where

$$\deg_i(s_{k_h}) = \frac{\sum_{u_j \in map_i(s_{k_h})} (m_{ij})}{|map_i(s_{k_h})|}.$$

Trivially, $w(s_{k_h}) = \text{pop}_i(s_{k_h}) \cdot \deg_i(s_{k_h})$.

The following theorem simply shows that the recommendations are sorted according to the criteria inspired by the preference networks which the node is connected to: a missing file is suggested for its popularity amongst the friends of the users and for *affinity* degree of the node that stores the given file.

Theorem 1: Given a node u_i, and two files s_{k_x} and s_{k_y} in Co-f(u_i), s.t., $w(s_{k_x}) > w(s_{k_y})$, then the following statements are true:

1. If the files have the same popularity, then s_{k_x} is owned mostly by nodes with a higher average partnership degrees w.r.t. s_{k_y}.
2. If the files are owned by nodes with the same average partnership degree, then s_{k_x} is more popular than s_{k_y} in Co-f(u_i).

Proof. Note that the hypothesis says that $w(s_{k_x}) > w(s_{k_y})$, that means that $\text{pop}_i(s_{k_x}) \cdot \deg_i(s_{k_x}) > \text{pop}_i(s_{k_y}) \cdot \deg_i(s_{k_y})$.
It is easy to show that, when $\text{pop}_i(s_{k_x}) = \text{pop}_i(s_{k_y})$ (> 0), then it follows that $\deg_i(s_{k_x}) > \deg_i(s_{k_y})$, which proves the first part of the theorem.
The second enunciation states that, on the contrary, $\deg_i(s_{k_x}) = \deg_i(s_{k_y})$ (> 0); in this case, we have that $\text{pop}_i(s_{k_x}) > \text{pop}_i(s_{k_y})$, which proves the theorem.

5.3 Implementation

For evaluation purposes, we implemented a recommendation module that can be integrated to a previously installed Phex servent. During testing, we noted that only after few interactions, the list of friends of the given node becomes quite long: Phex acts in a multi-download fashion, hence after only one download, we have as many contacts as the number of different parts of the divided file. Indeed, lists of friends and files grow very quickly, and as a consequence, in order to preserve lightness of the peer's state, lists are forced to be limited in size. Optimizations based on Bloom filters are under study. Anyway, the small world property of the preference networks makes the files and the partners lists' lengths to stabilize after a while.

Of course, we need to spread the module to many users, and ask them to allow us to monitor part of their activities (without asking them to tell us what they share, for privacy reasons). There are maybe many different scenarios that cannot be foreseen at the moment, and that only a controlled trial period of the package can reveal.

6 Conclusions and Ongoing Work

We defined the concept of preference networks, and we showed that such graphs, built by means of data collected by a modified Gnutella Ultra-Peer, are characterized by small world topologies. Moreover, starting from these findings, we described a fully decentralized recommendation scheme that can be easily implemented in many popular p2p file sharing clients. The most of the relevance of such a result is that only information given by self organizing communities and natural clusters of partnerships are taken into consideration, without defining any semantic knowledge and any ontology.

We also implemented a open source prototype for the Phex Gnutella servent, that will be used for future analysis and evaluations.

Acknowledgment

Part of this work has been financially supported by the Italian FIRB 2001 project number RBNE01WEJT “Web MiNDS”.

References

1. M.E.J.A. Newman. A study of scientific co-authorship networks. *Journal Physics Review*, 20, 2000.
2. T. J. Fararo and M. Sunshine. *A Study of a Biased Friend-ship Network*. Syracuse University Press, 1964.
3. D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440442, 1998.
4. M. E. J. Newman. Models of the small world. *J. Stat. Phys.*, 101:819–841, 2000.
5. N. Leibowitz, M. Ripeanu, and A. Wierzbicki. Deconstructing the kazaa network. In *Proc. of the Third IEEE Workshop on Internet Applications*. IEEE Press, June 2003.
6. I. Foster A. Iamnitchi, M. Ripeanu. Small-world file-sharing communities. In *The 23rd Conference of the IEEE Communications Society (InfoCom 2004)*, Hong Kong, 2004.
7. Gnutella 0.6 Protocol Specification. http://www.gnutella2.com/index.php/-main_page#the_protocol.
8. D. Stutzbach, R. Rejaie, and S. Sen. Characterizing unstructured overlay topologies in modern p2p file-sharing systems. In *Proc. of the ACM SIGCOMM Internet Measurement Conference*, October 2005.
9. K. Sripanidkulchai, B. Maggs, and H. Zhang. Efficient content location using interest-based locality in peer-topeer systems. In *InfoCom*, 2003.
10. Girish Suryanarayana and Richard N. Taylor. A survey of trust management and resource discovery technologies in peer-to-peer applications. Technical report, UC Irvine, 2004.
11. Phex Gnutella Client. http://phex.kouk.de/mambo/.
12. Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de, and E. Lear. Address allocation for private internets. RFC 1918, Internet Engineering Task Force, February 1996.
13. R. Albert. *Statistical mechanics of complex networks*. PhD thesis, 2001.
14. Crespo and Garcia-Molina. Semantic overlay networks for p2p systems. Technical report, Computer Science Department, Stanford University, 2002.
15. N. Borch. Improving semantic routing efficiency. In *Proc. of the 2nd Inter. Workshop on Hot Topics in Peer-to-Peer Systems (HOT-P2P'05)*, July 2005.
16. J. W. von Goethe. *Die Wahlverwandtschaften*. 1809.