An Adaptive Routing Mechanism for Efficient Resource Discovery in Unstructured P2P Networks

Luca Gatani[1], Giuseppe Lo Re[1,2], and Salvatore Gaglio[1,2]

[1] Dip. di Ingegneria Informatica, Università di Palermo,
[2] Istituto di Calcolo e Reti ad Alte Prestazioni, C.N.R.
Viale delle Scienze, 90128 Palermo, Italy
gatani@csai.unipa.it
{lore, gaglio}@unipa.it

Abstract. The widespread adoption of large-scale decentralized peer-to-peer (P2P) systems imposes huge challenges on distributed search and routing. Decentralized and unstructured P2P networks are very attractive because they require neither centralized directories, nor precise control over network topology or data placement. However their search mechanisms are extremely unscalable, generating large loads on the network participants. In this paper, to address this major limitation, we propose and evaluate the adoption of an innovative algorithm for routing user queries. The proposed approach aims at dynamically adapting the network topology to peer interests, on the basis of query interactions among users. Preliminaries evaluations show that the approach is able to dynamically group peer nodes in clusters containing peers with shared interests and organized into a *small world* topology.

1 Introduction

In the past few years, the peer-to-peer (P2P) paradigm has emerged, mainly by file sharing systems such as Napster and Gnutella. In the research community there has been an intense interest in designing and studying such systems. Due to the decentralization, these systems promise improved robustness and scalability, and therefore they open a new view on data integration solutions. However, several design and technical challenges arise in building scalable systems. While active, each peer maintains neighbor relationships with a small set of peers and it participates in the application protocol on the P2P network. These neighbor relationships, logical links between peers, define the topology of the P2P network. Therefore, the P2P topology forms an overlay on the IP-level connectivity of the Internet. One of the most challenging problems related to data-sharing P2P systems, is the content location. Content location determines whether the system resources can be efficiently used or not. Moreover, it greatly affects the scalability of P2P systems and their other potential advantages. Currently, there are two kinds of searching schemes for decentralized P2P systems [1]: structured searching scheme and unstructured searching scheme. Although structured systems such as Chord [2], Pastry [3], and CAN [4] scale well and perform efficiently,

O. Gervasi et al. (Eds.): ICCSA 2005, LNCS 3482, pp. 39–48, 2005.

they have many limitations. First, they have high requirements on data placement and network topology, which are not applicable to the typical Internet environment, where users are widely distributed, extremely transient, and come from non-cooperating organizations. Second, they efficiently support search by identifiers, but not general search facilities. Unstructured P2P systems like Gnutella [5] do not have the problems mentioned above and they are the most suitable in the current Internet. However, they use an overlay network in that the topology and file placement are largely unconstrained, so no clue emerges as to where content is located, and queries have to be flooded through the whole network to get results. Flooding is robust and reliable but highly redundant [6], producing many duplicated messages and much network traffic. Several researches are currently carried out on improvements in flooding-based message routing scheme: most notable are random walk, multiple parallel random walks, iterative deepening, and local indexes. An interesting paper by Adamic *et al.* [7] studies random walk search strategies in power-law topology, but although otherwise effective, this strategy places most of the burden on the high degree nodes and thus potentially creates additional bottlenecks and points of failure, reducing the scalability of the search algorithm. Lv *et al.* [1] propose a k-walker random walk algorithm that reduces the load generated by each query. They also propose the adoption of uniform random graph, and study query efficiency under different query and replication models. Liu *et al.* [8] introduce a location-aware topology matching technique, building an efficient overlay by disconnecting low productive connections and choosing physically closer nodes as logical neighbors. Another interesting query routing technique is the iterative deepening, suggested by Yang *et al.* [9], where unsatisfied queries are repeated several times, each time increasing their search depth. In [10], the authors propose that nodes maintain metadata that can provide "hints" such as which nodes contain data that can answer the current query. Query messages are routed by nodes making local decisions based on these hints. In this paper, we focus on the unstructured architectural model and, to address searching inefficiencies and scalability limitations, we propose an adaptive routing algorithm whose aim is to suppress flooding. The routing algorithm adopts a simple Reinforcement Learning scheme (driven by query interactions among neighbors), in order to dynamically change the topology of the peer network based on commonality of interests among users.

The remainder of the paper is structured as follows. Section 2 provides background on the *small world* network paradigm. Section 3 illustrates the adaptive routing protocol proposed. The simulation setup and performance measurements are presented in Section 4. Finally, Section 5 concludes the paper.

2 Small World Networks

The *small world* phenomenon was first observed by Milgram [11], who discovered the interesting "six degrees of separation" in a social network. Although the notion of *small world* phenomenon originates from social science research, it has been observed that the *small world* phenomenon is pervasive in a wide range of

settings such as social communities, biological environments, and data communication networks. For example, recent studies (e.g., [12]) have shown that P2P networks such as Freenet may exhibit *small world* properties. Roughly speaking, a *small world* network can be viewed as a connected graph characterized by low *characteristic path length*(i.e., similar to the average path length in random networks) and high *clustering coefficient* (i.e., much greater than that of random networks). To mathematically define the two properties, let $G = (V, E)$ denote a connected graph modeling a *small world* network, $N = |V|$ the cardinality of the set of vertices, and $D(i, j)$ the length (in hops) of the shortest path between two vertices i and j in V. The characteristic path length $L(G)$ is defined as the number of edges in the shortest path between two vertices, averaged over all pairs of vertices. To define the clustering coefficient $C(G)$, suppose that a vertex $v \in V$ has k_v neighbors; then at most $k_v(k_v - 1)/2$ edges can exist between them (this occurs when every neighbor of v is connected at every other neighbor of v). Let C_v, the local clustering coefficient of v, denote the fraction of these allowable edges that actually exist. Define the clustering coefficient as the average of C_v over all v. While L measures the typical separation between two vertices in the graph (a global property), C measures the cliquishness (degree of compactness) of a typical neighborhood (a local property). A low average hop distance implies that one can locate information stored at any random node by only a small number of link traversals (low latency object lookup), while a high clustering coefficient implies the network can effectively provide contents even under heavy demands. Other works, in addition to our proposal of constructing a *small world* network, discuss the adoption of a *small world* topological structure in order to efficiently perform searching in P2P networks. Iamnitchi *et al.* [13] propose a solution for locating data in decentralized, scientific, data-sharing environments that exploits the small-worlds topology. Manku *et al.* [14] propose to build a one-dimensional *small world* network, by the adoption of a simple protocol for managing a distributed hash table in a dynamic peer network. The broadcast problem for communication in a *small world* network is considered in [15]. In [12], the authors propose a scheme for storing data in an unstructured P2P network such as Freenet, such that the P2P network may exhibit some of the *small world* properties.

3 Adaptive Routing Protocol

The key problem addressed in our work is the efficient and scalable localization of shared resources. The underlying idea is that an intelligent collaboration between the peers can lead to an emergent clustered topology, in which peers with shared interests and domains, tend to form strongly connected communities of peers. To this aim, we adopt an approach that dynamically selects the neighbors to which a query has to be sent or forwarded. The selection process is driven by an adaptive learning algorithm by which each peer exploits the results of previous interactions with its neighbors to build and refine a model (*profile*) of the other peers, describing their interests and contents. When an agent must

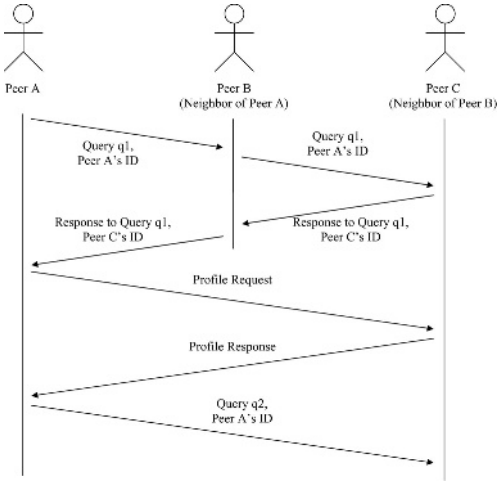


Fig. 1. The process of neighbor discovery

forward a query, it compares the query with its known profiles, in order to rank all known peers and to select the best suited to return good response. The network topology (i.e., the actual set of peers that are neighbors in the overlay) is then dynamically modified on the basis of the learned contexts and the current information needs, and the query is consequently routed according to the predicted match with other peers' resources. Since our goal is to allow peers to form communities in a fully distributed way, they need to find new peers and to evaluate their quality in relation to their own interests. Initially the network has a random, unstructured topology (each peer is assigned N_s neighbors randomly chosen), and queries are forwarded as in the scoped flood model. The peers can then discover other peers through known peers, during the normal handling of queries and responses. Each peer has a fixed number, N_m, of slots for known peers. This number can vary among peers depending on their available memory. Here, we assume that the N_m value is the same for each agent. For each known peer, a profile concisely describing the shared resources is stored. The actual set of N_a neighbors, i.e. those to whom queries are sent, is selected dynamically for each query at time step t among all the $N_k(t)$ known peers. In particular, when a peer receives a query locally generated, it compares the query with its stored profiles, applying a simple ranking algorithm for dynamically selecting peers to which it sends the query. The maximum number of selected peers, N_a, depends on peer bandwidth and computational power. In our test networks, we assume that N_a is fixed and equal for each agent. The system currently adopts Bloom filters [16] to build peer profiles and supports a basic query language where a query string is interpreted as a conjunction of keys. When presented with a query, the system searches the information in its profile database in order to obtain a list of candidate peers that might have data matching the query. When a peer receives a query by another peer, if it shares resources that match the request,

1. `Profile acquisition.` When a peer is first discovered, a profile is requested. After the peer's profile is acquired, a local peer description is initialized with the information stored in the Bloom filter.
2. `Profile updating.` When a response from a neighbor (or from a neighbors' neighbor) arrives, it is evaluated and used to update the description of known peers, adding the query keywords to the peer profile. Moreover, new peers that respond to issued queries are added to the list of known peers.
3. `Peer ranking.` When a new query has to be sent, all N_k known peers are ranked by similarity between the query and the peer descriptions, exploiting the membership information provided by Bloom filters.
4. `Query sending.` The new query is sent to the top N_a ranked peers. Then step 1 is newly considered.

Fig. 2. Basic steps of the algorithm proposed

it can directly respond. Moreover, it can forward the query to that neighbors, whose profiles match the query. To this aim, the peer uses the same selection algorithm applied to locally generated queries In order to prevent potential DoS attacks which exploit the response system, we impose that a peer replies to a forwarded query sending the response to the neighbor that has forwarded the query, and not directly to the originating peer. To limit congestion and loops in the network, queries contain a Time-To-Live (TTL), which is decreased at each forward, and queries will not be forwarded when TTL reaches 0. When a peer receives the responses for a locally generated query, it can start the actual resource downloading. Moreover, if a peer that has sent a response is not yet included in the list of known peers, a profile request is generated. The two peers contact each other directly (see Fig. 1). When the message containing the profile will arrive, the new peer will be inserted among the N_k known peers and its characteristics will be evaluated in order to select actual neighbors for new query. The stored profiles are continually updated according to the peer interactions during the normal system functioning (i.e., matches between queries and responses). Moreover, a peer can directly request a newer profile when necessary. In Fig. 2, we summarize the main steps of the proposed adaptive algorithm.

4 Experimental Evaluation

4.1 The Simulator

Since in the studies on deployed P2P networks [6, 17, 18], the dynamics in peer lifetimes and the complexity of these networks make it difficult to obtain a precise comprehensive snapshot, we use simulation to perform a preliminary evaluation of the proposed approach. Simulation of P2P networks can provide a thorough evaluation and analysis of their performance. To study the behavior of peer interactions in our system, we have implemented a simple simulator that allows

Table 1. Parameters used in Scenario 1

Parameter	Value
Number of peers N	100
Number of actual neighbors N_a	5
Number of initial neighbors N_s	5
Maximum number of known peers N_m	99
Number of groups N_g	10
Time To Live of queries TTL	3
Number of time steps N_ts	1000

us to model synthetic peer networks and run queries according to the routing protocol adopted. The goal of the simulator in the preliminary experiments reported below is to analyze the topology properties of emergent peer networks. Our simulator takes a snapshot of the network for every time step. In a time step of the simulator, all of the peers process all of their buffered incoming messages and send all of their buffered outgoing messages. This may include the generation of a local query as well as forwarding and responding to the queries received by other peers.

4.2 Simulation Scenarios

In the current evaluation, we perform four different kinds of experiments, considering four scenarios, each of them different from the others for a single simulation parameter. For each scenario, the aim is to study how the network statistics change when the parameter value changes. Since the initial random topology can affect the final results, for each scenario, we perform several independent simulations and we average the results. In order to study whether the proposed algorithm can generate network topologies that capture user interests, thus reducing query flooding problems, we model synthetic peers belonging to different groups of interest (let N_g denote the number of groups in the network). Each group is associated with a general topic. Within each topic, the resources are further classified into categories and sub-categories. In this preliminary evaluation, we consider four scenarios: (*i*) Scenario 1, used as a baseline for all the others experiments (its simulation parameters are reported in Table 1), (*ii*) Scenario 2, characterized by the N_a value variation, (*iii*) Scenario 3, characterized by the N_s value variation, (*iv*) Scenario 4: characterized by the TTL value variation.

4.3 Evaluation Metrics

For the analysis of experimental results, we consider the two network metrics previously introduced: the clustering coefficient and the characteristic path length. The clustering coefficient is computed in the directed graph based on each peer's N_a neighbors, with a total of $N_a(N_a - 1) = 20$ possible directed links between neighbors. The overall clustering coefficient $C(G)$ is computed by averaging across all peer nodes. The characteristic path length $L(G)$ is defined as the

average shortest path length across all pairs of nodes. Since in our simulations the network is not always strongly connected we used an alternative way to average shortest paths:

$$\tilde{L} = \left(\frac{1}{P} \sum_{p=1}^{\infty} l_p^{-1} \right)^{-1}, \tag{1}$$

where p is a pair of nodes and $P = N(N-1)$ is the number of all possible pairs. The characteristic path length $\tilde{L}(G)$ thus defined can be computed from all pairs of nodes irrespective of whether the network is connected. Another interesting network metric taken in account is the number of connected components. A connected component, which in the case of a directed graph is called strongly connected component (SCC), is a strongly connected subgraph, S, of the directed graph G, such that no vertex of G can be added to S and it still be strongly connected (informally, S is a maximal subgraph in which every vertex is reachable from every other vertex). C, $\tilde{L}$, and SCC are measured at each time step and averaged across simulation runs.

4.4 Simulation Results

Scenario 1 (a baseline). In the first scenario we investigate the system behavior when the simulation parameters are set to some representative, base values. The results obtained are used as a baseline to compare the algorithm performances in the others scenarios, when significant parameters are varied. Fig. 3

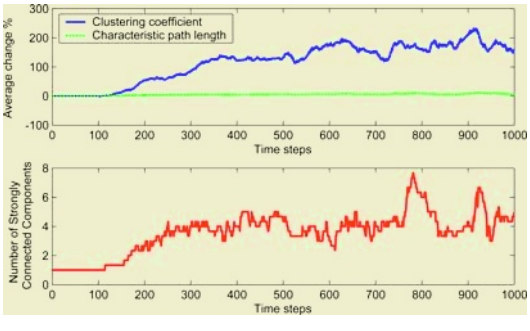
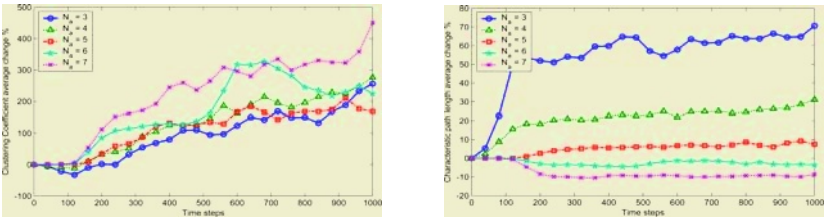


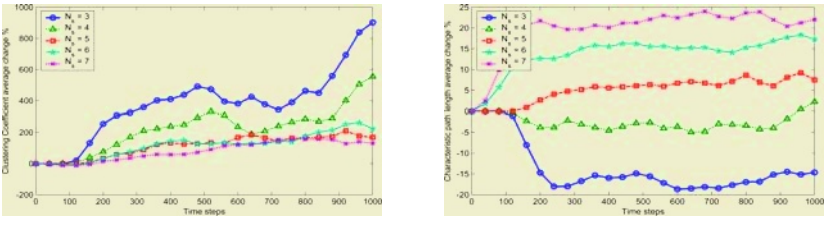
Fig. 3. Scenario 1: clustering coefficient, characteristic path length, and SCCs

plots the evaluation metrics taken into account. In particular, the top chart shows that the characteristic path length remains roughly equal to the initial random graph characteristic path length while the clustering coefficient increases rapidly and significantly, reaching a value that is, in average, 150% larger than that of the initial random graph. These conditions define the emergence of a *small world* topology in our peer network [19]. This is a very interesting finding, indicating that the peer interactions cause the peers to route queries in such a way that communities of users with similar interests cluster together to find quality results



(a) Clustering coefficient (b) Characteristic path length

Fig. 4. Scenario 2



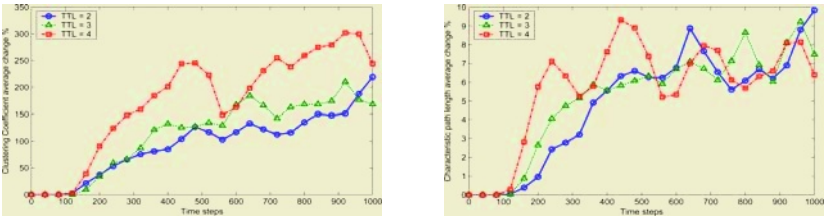
(a) Clustering coefficient (b) Characteristic path length

Fig. 5. Scenario 3

quickly, while it is still possible to reach any peer in a small number of hops. In the bottom of Fig. 3, the number of strongly connected components (SCC) is reported. It is easy to observe that the network splits into a few SCCs (it remains connected in the weak sense, based on undirected links). The number of SCCs is smaller than the number of groups: while the network is becoming more clustered and localized, peers continue to have access to most peers in other groups.

Scenario 2 (varying N_a). In this experiment, we choose to vary the N_a value, i.e. the maximum number of peers to which a query can be sent. The base N_a value is set to 5. We perform additional simulations investigating also systems with $N_a = 3, 4, 6, 7$. In Fig. 4(a) the clustering coefficient is plotted for the different values of N_a. The x-axis reports the time steps and the y-axis the percentage average variation of clustering coefficient. It is expected that the coefficient grows when N_a becomes larger, denoting the constitution of bigger peer groups with shared interests. The characteristic path length for different N_a values is plotted in Fig. 4(b). We observe that the characteristic path length stabilizes around a value that is lower for larger values of N_a. That is because there are more links and the network has a greater degree of compactness.

Scenario 3 (varying N_s). In this experiment, we study how the N_s value (i.e., the number of peers known at the simulation start-up) affects the system



(a) Clustering coefficient (b) Characteristic path length

Fig. 6. Scenario 4

performance. We perform simulations with $N_s = 3, 4, 5, 6, 7$. From the results (see Fig. 5) we note that the emerging networks present similar topological structures. When the N_s values grow the statistical differences between the initial network structure and the final one are less evident.

Scenario 4 (varying the TTL). In this experiment, we consider how to choose the TTL value, i.e. the maximum number of links traversed by a query. The base TTL value is set to 3. We perform additional simulations investigating also systems with $TTL = 2, 4$. It can be seen from Fig. 6 that the emerging of a *small world* topological structure is faster for higher TTL values. This is mainly caused by the fact that, during query propagation, an higher TTL value is associated with a deeper exploration of the network graph, allowing a peer to learn in a faster way the characteristics of the other peers.

5 Conclusion

In this paper, we presented a novel mechanisms for improving search efficiency in unstructured P2P networks, and we evaluated its efficiency. To address major limitations of unstructured P2P networks, we propose an adaptive routing algorithm in order to dynamically change the topology of the peer network, based on commonality of interests among users. Preliminary results confirm the idea that adaptive routing can properly work and that *small world* network topological structure can emerge spontaneously from the local interactions between peers. This is a very interesting finding, indicating that the peer interactions cause the peers to structure the overlay in such a way that communities of peers with similar interests cluster together in order to find quality results quickly, while it is still possible to access any network peer in a small number of hops.

References

1. Lv, Q., Caoa, P., Cohen, E., Li, K., Shenker, S.: Search and replication in unstructured Peer-to-Peer networks. In: Proc. ACM ICS. (2002)
2. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F.: Chord: A scalable Peer-to-Peer lookup service for internet applications. In: Proc. ACM SIGCOMM 01. (2001) 149–160

3. Rowstron, A., Druschel, P.: Pastry: Scalable, distributed object location and routing for large-scale Peer-to-Peer systems. In: Proc. of the IFIP/ACM International Conference on Distributed Systems Platforms. (2001)
4. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content-addressable network. In: Proc. ACM SIGCOMM 01. (2001) 161–172
5. Limewire: The Gnutella protocol specification (ver. 0.4). (http://www.limewire.com/developer/gnutella_protocol_0.4.pdf)
6. Ripeanu, M., Foster, I., Iamnitchi, A.: Mapping the Gnutella network: Properties of large scale Peer-to-Peer systems and implications for system design. IEEE Journal on Internet Computing, Special Issue on Peer-to-peer Networking (2002)
7. Adamic, L.A., Lukose, R.M., Puniyani, A.R., Huberman, B.A.: Search in power law networks. Phys. Rev. E64 (2001) 46135–46143
8. Liu, Y., Liu, X., Xiao, L., Ni, L.M., Zhang, X.: Location-aware topology matching in P2P systems. In: Proc. IEEE INFOCOM. (2004)
9. Yang, B., Garcia-Molina, H.: Effcient search in Peer-to-Peer networks. In: Proc. IEEE ICDCS. (2002)
10. Crespo, A., Garcia-Molina, H.: Routing indices for Peer-to-Peer systems. In: Proc. IEEE ICDCS. (2002)
11. Milgram, S.: The small world problem. Psychology Today **2** (1967) 60–67
12. Zhang, H., Goel, A., Govindan, R.: Using the small-world model to improve Freenet performance. In: Proc. IEEE INFOCOM. (2002)
13. Iamnitchi, A., Ripeanu, M., Foster, I.: Small-world file-sharing communities. In: Proc. IEEE INFOCOM. (2004)
14. Manku, G.S., Bawa, M., Raghavan, P.: Symphony: Distributed hashing in a small world. In: Proc. of USENIX Symposium on Internet Technologies and Systems. (2003)
15. Comellas, F., Mitjana, M.: Broadcasting in small-world communication networks. In: Proc. 9th Int. Coll. on Structural Information and Communication Complexity. (2002) 73–85
16. Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors. Communications of the ACM **13** (1970) 422–426
17. Saroiu, S., Gummadi, K., Gribble, S.: A measurement study of Peer-to-Peer file sharing systems. In: Proc. ACM Multimedia Conferencing and Networking. (2002)
18. Sen, S., Wang, J.: Analyzing Peer-to-Peer traffic across large networks. IEEE/ACM Transactions on Networking **12** (2004) 212–232
19. Watts, D., Strogatz, S.: Collective dynamics of small-world networks. Nature **393** (1998) 440–442