

An Adaptive Routing Protocol for Ad Hoc Peer-to-Peer Networks

Luca Gatani
DInfo, Univ. di Palermo
Viale delle Scienze
90128 Palermo - Italy
gatani@csai.unipa.it

Giuseppe Lo Re
DInfo, Univ. di Palermo
ICAR, CNR
Viale delle Scienze
90128 Palermo - Italy
lore@unipa.it

Salvatore Gaglio
DInfo, Univ. di Palermo
ICAR, CNR
Viale delle Scienze
90128 Palermo - Italy
gaglio@unipa.it

Abstract

Ad hoc networks represent a key factor in the evolution of wireless communications. These networks typically consist of equal nodes that communicate without central control, interacting in a peer-to-peer way. In such a network, an efficient and scalable data retrieval constitutes a challenging problem. Unstructured P2P networks avoid the limitations of centralized systems and the drawbacks of structured approaches, because they impose few constraints on topology and data placement, and support highly versatile search mechanisms. However their search algorithms are usually based on simple flooding schemes, showing severe inefficiencies. In this paper, in order to address this major limitation, we evaluate the adoption of a local adaptive routing protocol, suitable for a self-organizing ad hoc environment. The routing algorithm uses a simple reinforcement learning scheme (driven by query interactions among peers), in order to dynamically adapting the topology to peer interests. In our simulation this approach is able to dynamically group peer nodes in clusters containing peers with shared interests and organized into a small world network.

1 Introduction

The rapid increase in the usage of mobile devices and the accompanied maturity in wireless technologies, such as 802.11, have made wireless networks almost as ubiquitous as traditional wired networks [5]. An important characteristic of such networks is ad hoc connectivity. Ad hoc networks are characterized by the lack of required infrastructure, their decentralized operations and their ease of constitution. Their multi-hop nature and the possible lack of a fixed infrastructure introduce new research problems such as network configuration, device discovery, and topology maintenance, as well as ad hoc addressing and self-routing.

Several approaches and protocols have been proposed to address ad hoc networking problems, and multiple standardization efforts are under way within the Internet Engineering Task Force, as well as academic and industrial research projects. In ad hoc networks, wireless hosts can communicate with each other in the absence of a fixed infrastructure. These networks typically consist of equal nodes that communicate over wireless links without central coordination. Some of the most interesting applications and scenarios for such networks are characterized by nodes interacting in a peer-to-peer way. Peer-to-peer (P2P) networks are ad hoc networks in which an overlay network is built over the Internet. In a P2P network, two or more peers can use appropriate information and communication systems to collaborate spontaneously without requiring central coordination. The P2P paradigm has emerged in the past few years, mainly by file sharing systems such as Napster and Gnutella. In the research community there has been an intense interest in designing and studying P2P systems. Due to their decentralization, these systems promise an improved robustness and scalability, and therefore they also open a new view on data integration solutions. However, several design and technical challenges arise in building scalable systems. The P2P networks differ from the conventional client-server approach in several ways. The most distinguishing characteristic is that a peer acts both as a client, and a server of the system. For example, in file sharing, a peer both requests files from its peers, and serves files to them. A second important difference is constituted by the transient lifetime of peers and their asynchronous arrivals and departures. While active, each peer maintains neighbor relationship with a small set of peers and participates in the application protocol on the P2P network. This neighbor relationship, i.e. a logical link between peers, defines the topology of the P2P network. Therefore, the P2P topology forms an overlay on the IP-level connectivity of the Internet. Messages of the P2P application protocol are exchanged between peers on this overlay topology. P2P networks are a special case of con-

tent distribution networks (i.e., overlay networks that use content-based routing on top of the IP network), where all the nodes have equal roles. Messages are not routed on the basis of the destination addresses, rather on the basis of the message contents. 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, affecting the scalability of P2P systems and their other potential advantages.

1.1 Motivation and Contribution

Currently, there are two kinds of searching schemes for decentralized P2P systems [9]: structured searching scheme and unstructured searching scheme. Although structured systems such as Chord [17], Pastry [16], and CAN [14] scale well and perform efficiently, they present many limitations. Firstly, 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 coming from non-cooperating organizations. Secondly, they efficiently support search by identifiers but not general search facilities. Thirdly, they only offer file level sharing. Unstructured systems like Gnutella [7] do not present the problems mentioned above and they are most widely used by current Internet users. However, unstructured P2P systems 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 [15], producing many duplicated messages and much network traffic, which involves great network load. The technique described in this paper tries to improve the scalability and efficiency of the resource discovery under the unstructured ad hoc P2P environment through an adaptive routing algorithm suppressing flooding. Our approach aims at dynamically adapting the network topology to peers' interests, on the basis of a peer neighbor selection algorithm. Each peer builds and maintains profiles of other peers, describing their interests and resources. Given a query, this is consequently routed according to the predicted match with other peers' profiles. Experimental evaluations show that the approach is able to exploit query interactions among users, in order to dynamically group peer nodes in clusters containing peers with shared interests and organized into a *small world* topology.

The remainder of the paper is structured as follows. Section 2 illustrates the adaptive routing protocol proposed. The simulation setup and results for a preliminary performance evaluation are presented in Section 3. We discuss related work in Section 4. Finally, we conclude with summary remarks in Section 5.

2 Adaptive Routing Protocol

2.1 Basic Idea

The key problem addressed in our work is the efficient and scalable localization of shared resources. Queries issued by a user are routed to neighbor peers in the overlay network, in order to find resources that satisfy them. 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. However, we adopt an approach that dynamically selects the neighbors to which a query has to be sent or forwarded. The selection process is performed with the aim to detect peers that with high probability share resources satisfying the query. The selection 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. Each peer is characterized by one (or several) general interest and shares resources according to its interest. The characteristics of each peer are summarized in a peer profile. When an agent must forward a query, it compares the query with its known profiles, in order to rank all known peers and 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. 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. The expected theoretical network topology should have *small world* properties [19] and experimental results in Section 3 confirm the hypothesis. In such a topology, a flood-based routing mechanism (with limited scope) is well suited, since it allows that any two peers can reach each other via a short path, while maximizing the efficiency of communication within clustered peer communities.

2.2 Algorithm

Since our goal is to allow peers to form communities in a fully distributed way, they should find new peers and evaluate their quality in relation to their own interests. When a peer enters the network for the first time, a bootstrap protocol returns some existing peers to get started. The new peer can then discover other peers through this known peers. In the proposed system, a peer would discover new peers through its current neighbors, during the normal handling of queries and responses. Each peer maintains a fixed

Table 1. Message set

Message type	Usage	Fields
Query	searching for a specific resource	(weighted) query keywords, query ID, source peer ID, generation timestamp, TTL
Query response	responding to a Query message	resource ID, query ID, responder ID, source peer ID, generation timestamp, TTL
Profile request	requesting a peer profile	request ID, source peer ID, target peer ID, generation timestamp
Profile response	responding to a Profile request	profile, request ID, responder ID, source peer ID, generation timestamp,

number, N_m , of slots for known peers. This number can vary among peers depending on their available memory (a peer must properly prune other peers' information when needed). Here, we assume that the N_m value is the same for each agent. For each known peer, a profile which concisely describes 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. Each peer applies a simple ranking algorithm for dynamically selecting peers to which query must be sent. The maximum number of selected peers, N_a , depends on peer bandwidth and computational power to process neighbor data. In our test networks, we assume that also N_a is fixed and equal for each agent. The system currently adopts Bloom filters [2] 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, it can answer directly. Furthermore, it can forward the query to those neighbors, whose profiles match the query. To this aim, the peer uses the same selection algorithm applied to locally generated queries (note that the peer automatically excludes both the peer that has forwarded the query, and the peer that has generated the query). 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. For this request, the two peers contact each other directly. When the message containing the profile will arrive, the new peer

will be inserted among the N_k known peers and its features will be considered in order to select actual neighbors for the following queries. The maintained profiles are continuously 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 more up-to-date profile if necessary. Table 1 describes the four basic messages our protocol uses to perform resource searching. The main steps of the algorithm can be summarized as follows.

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 extracted by 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.

Finally, we point out some featuring properties of the adaptive algorithm. First, Bloom filters allow to build peer profiles that concisely represent shared resources in a way which is very suitable for storage and remote transmission. Second, the analysis of peer interactions during the normal system functioning allows a continuous, reactive updating of profiles. Third, the peer ranking based on similarity between queries and peer features allows to adapt the topology to the information contents in a fully distributed way.

3 Simulation Analysis

3.1 Simulation Setup

In order to study the behavior of peer interactions, we developed a simple Java simulator that models synthetic peer

networks and runs queries according to the routing protocol adopted. The goal of our simulations is to analyze the properties of emergent peer networks, studying whether the proposed algorithm can generate network topologies that capture user interests, thus reducing query flooding problems. 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. 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 the current evaluation, we perform different kinds of experiments, considering several scenarios, each of them different from the others for a single simulation parameter. For each experiment, the aim is to study how the network statistics change when the parameter value changes. The first experiment is used as a baseline for all the others experiments (its simulation parameters are shown in Table 2). Other experiments are characterized by the variation of the main simulation parameter values (i.e., N , N_a , N_s , N_m , N_g , and TTL values). Since the initial random topology can affect the final results, for each experiment, we perform several independent simulations and we average the results.

3.2 Small world networks and Evaluation Metrics

Roughly speaking, a *small world* network can be viewed as a connected graph characterized by low *characteristic path length* between two randomly chosen nodes in the network (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 mea-

Table 2. Base values of parameters

Parameter	Symbol	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_t	1000

asures the cliquishness (degree of compactness) of a typical neighborhood (a local property). Since in our simulations the network is not always strongly connected, we adopt a more practical definition for the characteristic path length, using the harmonic means of shortest paths:

$$\tilde{L}(G) = \left(\frac{N}{N-1} \sum_{i,j \in V} d(i,j)^{-1} \right)^{-1}. \quad (1)$$

The characteristic path length $\tilde{L}(G)$ thus defined can be computed from all pairs of nodes irrespective of whether the network is connected. C and \tilde{L} are computed in the directed graph based on each peer N_a neighbors, with a total of $N_a(N_a - 1)$ possible directed links between neighbors. The metrics are measured at each time step and averaged across simulation runs.

3.3 Simulation Results

Due to paper length restrictions, in the following we only report the results for the base experiment and for the experiments studying N_a , N_m , and TTL value variations. The first experiment is used to study 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. 1 plots the evaluation metrics taken into account. In particular, the 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 quickly, while it is still possible to reach any peer in a small number of hops.

In the second experiment, we choose to vary the N_a value, i.e. the maximum number of peers to which a query

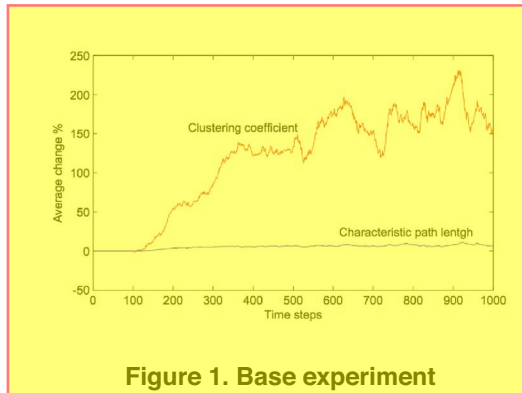


Figure 1. Base experiment

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. 2(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. 2(b). We observe that the characteristic path length stabilizes around a value that is lower for larger values of N_a . That is why there are more links and the network has a greater degree of compactness.

In the following experiment, we analyze the choice of the N_m value, i.e. the number of slots for known peers. This number represents the size of the cache containing the peer profiles. We perform simulations with $N_m = 10, 20, 30, 50, 99$. We consider a LRU replacement policy for cache management. In Fig. 3 the main results are plotted. We note that the network metrics are significantly affected by N_m variations. As it is easy to guess, a larger cache grants a better performance, i.e. higher values for clustering coefficient and lower values for characteristic path length. Nevertheless, it is worth to note that a moderate limitation on the number of cache entries is not necessary harmful, since it restricts the allowable peers to a more stable subset, avoiding excessive changes in peer connectivity at overlay layer. For instance, a cache limited to half size the number of all peers allows the creation of peer communities with a very high degree of compactness, retaining properties comparable to those observed when the cache is unlimited.

Finally, in the last experiment we consider the choice of 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. 4 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 peers

to learn in a faster way the characteristics of the other peers.

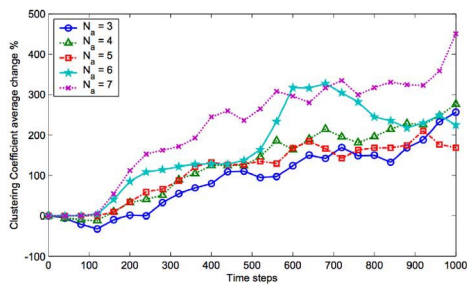
4 Related Work

4.1 Searching in Unstructured P2P Networks

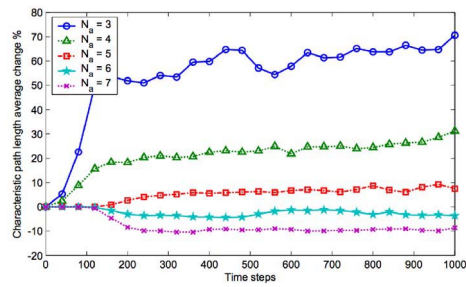
Several researchers are working on improvements in flooding-based message routing schemes; most notable are: random walk, multiple parallel random walks, iterative deepening, and local indexes. An interesting paper by Adamic *et al.* [1] studies random walk search strategies in power-law topology (on the application layer), 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. Portmann and Seneviratne [13] study cost-effective broadcast where the message should reach each node once or as few times as possible. Their method, called rumor mongering, is based on forwarding a message to some of the neighbor nodes. The method is better than flooding in the total number of message transmissions and in the average node cost (especially in highly redundant topology), while the messages still reach a comparable number of nodes. However the method does not reduce the number of nodes reached and therefore can only have limited effectiveness gain. Lv *et al.* [9] propose a k -walker random walk algorithm that reduces the load generated by each query. They also propose a topological structure (uniform random graph) that is most suitable for such routing method, 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.* [20], where unsatisfied queries are repeated several times, each time increasing their search depth. In [4], the authors propose that each node maintain metadata that can provide “hints” as which nodes may contain data that can answer the current query. Query messages are routed by nodes making local decisions based on these hints.

4.2 Small World Networks

The *small world* phenomenon was first observed by Milgram [12], 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 this phenomenon is pervasive in a wide range of settings such as social communities, biological environments, and data communication

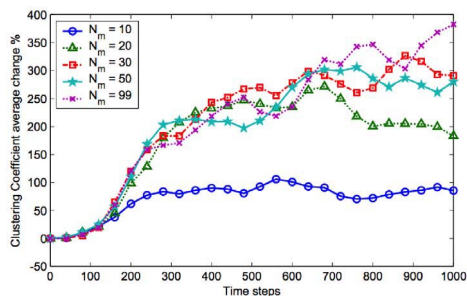


(a) Clustering coefficient

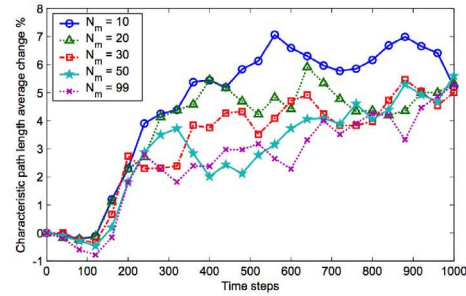


(b) Characteristic path length

Figure 2. Experiment varying N_a



(a) Clustering coefficient



(b) Characteristic path length

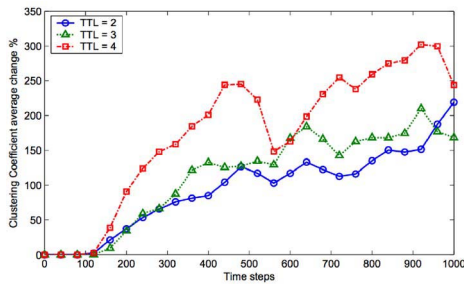
Figure 3. Experiment varying N_m

networks. For example, recent studies (e.g., [21]) have shown that P2P networks such as Freenet may exhibit *small world* properties. 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.* [6] propose a solution for locating data in decentralized, scientific, data-sharing environments that exploits the *small world* topology. Manku *et al.* [10] 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 [3]. In [21], 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. In [18], Tassier and Menczer adopts an evolutionary model, based on local selection algorithms, to study U.S. labor markets: the evolved networks display mixtures of regularity and randomness, as in *small world* networks. Finally, Menczer *et al.* [11] propose a local adaptive routing algo-

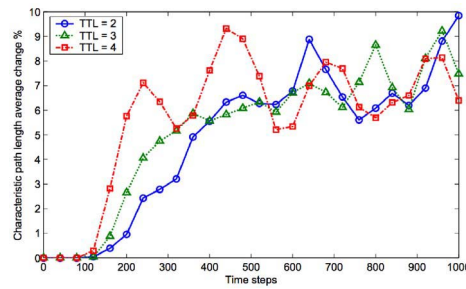
rithm that dynamically modify the network topology toward a *small world* structure, using a learning scheme similar to that considered in this paper. However, they design their protocol with the aim of supporting an alternative model for peer-based Web-search, where the scalability limitations of centralized search engines can be overcome via distributed Web crawling and searching.

5 Conclusion

In this paper, we presented an alternative message routing approach in unstructured ad hoc P2P overlay networks and by means of simulations evaluated its efficiency. Unstructured P2P systems are loosely coupled, highly autonomous systems without hierarchy. Usually, flooding-based routing mechanisms are used for sending messages through the application layer overlay. We suggested an adaptive routing algorithm to dynamically change the topology of the peer network based on commonality of interests among users, so as to avoid the problems of limited



(a) Clustering coefficient



(b) Characteristic path length

Figure 4. Experiment varying *TTL*

scalability and high network load. The preliminary results presented here confirmed the idea that adaptive routing can properly work and that *small world* network structure can emerge spontaneously from the local interactions between peers. We noted that in all the simulations the network diameter remains quite constant, while the cluster coefficient increases rapidly and significantly. These conditions characterize the emergence of a *small world* topology in the peer network [19]. This is a very interesting finding, indicating that our approach is able to structure the overlay in such a way that it is possible both to locate information stored at any random node by only a small number of hops (low latency object lookup), and to find quality results quickly and even under heavy demands (high clustering coefficient). Further research could elaborate on simulating the dynamics of network changes (joins, departs, content changes), on analyzing alternative representation for queries and profiles (e.g., LSI, semantic annotations), and on evaluating alternative learning algorithms for adaptive query routing.

References

- [1] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman. Search in power law networks. *Phys. Rev. E* 64, pages 46135–46143, 2001.
- [2] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, 1970.
- [3] F. Comellas and M. Mitjana. Broadcasting in small-world communication networks. In *Proc. SIROCCO*, pages 73–85, 2002.
- [4] A. Crespo and H. Garcia-Molina. Routing indices for peer-to-peer systems. In *Proc. IEEE ICDCS*, pages 23–32, 2002.
- [5] A. Helal, B. Haskell, J. Carter, R. Brice, D. Woelk, and M. Rusinkiewicz. *Any Time Anywhere Computing: Mobile Computing Concepts and Technology*. Kluwer Academic Publishers, 1999.
- [6] A. Iamnitchi, M. Ripeanu, and I. Foster. Small-world file-sharing communities. In *Proc. IEEE INFOCOM*, 2004.
- [7] Limewire. The Gnutella protocol specification (ver. 0.4). http://www.limewire.com/developer/gnutella_protocol_0.4.pdf.
- [8] Y. Liu, X. Liu, L. Xiao, L. M. Ni, and X. Zhang. Location-aware topology matching in P2P systems. In *Proc. IEEE INFOCOM*, 2004.
- [9] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Search and replication in unstructured peer-to-peer networks. In *Sigmetrics*, June 2002.
- [10] G. S. Manku, M. Bawa, and P. Raghavan. Symphony: Distributed hashing in a small world. In *Proc. USENIX Symp. on Internet Technologies and Systems*, March 2003.
- [11] F. Menczer and R. Belew. Adaptive retrieval agents: Internalizing local context and scaling up to the web. *Machine Learning*, 39(2-3):203–242, 2000.
- [12] S. Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.
- [13] M. Portmann and A. Seneviratne. Cost-effective broadcast for fully decentralized peer-to-peer networks. *Computer Communications*, 26(11):1159 – 1167, Jan. 2003.
- [14] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proc. ACM SIGCOMM*, pages 161–172, Aug. 2001.
- [15] M. Ripeanu, I. Foster, and A. Iamnitchi. Mapping the Gnutella network: Properties of large scale peer-to-peer systems and implications for system design. *IEEE Internet Computing Journal*, 6(1), 2002.
- [16] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proc. IFIP/ACM Middleware*, Nov. 2001.
- [17] I. Stoica, R. Morris, D. Karger, and M. F. Kaashoek. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proc. ACM SIGCOMM*, pages 149–160, Aug. 2001.
- [18] T. Tassier and F. Menczer. Emerging small-world referral networks in evolutionary labor markets. *IEEE Transactions on Evolutionary Computation*, 5(5):482 – 492, Oct. 2001.
- [19] D. Watts and S. Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440–442, 1998.
- [20] B. Yang and H. Garcia-Molina. Efficient search in peer-to-peer networks. In *Proc. IEEE ICDCS*, pages 23–32, 2002.
- [21] H. Zhang, A. Goel, and R. Govindan. Using the small-world model to improve Freenet performance. In *Proc. IEEE INFOCOM*, pages 1228–1237, 2002.