Evaluating Unstructured Peer-to-Peer Lookup Overlays

Idit Keidar
EE Department, Technion

Roie Melamed
CS Department, Technion

ABSTRACT

Unstructured peer-to-peer lookup systems incur small constant overhead per single join or leave operation, and can easily support keyword searches. Hence, they are suitable for dynamic failure-prone environments. In this paper, we define metrics for evaluating unstructured overlays for peer-to-peer lookup systems. These metrics capture the search dependability and efficiency, and the granularity at which one can control the tradeoff between the two, as well as fairness. According to these metrics, we evaluate different graphs and overlays, including a Gnutella graph, a power law random graph, normal random graphs, a 3-regular random graph, and a 3-Araneola overlay. Our study shows that, according to our metrics, a 3-Araneola overlay achieves the best results, and hence it is an excellent solution for flooding-based peer-to-peer lookup system.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed applications*

General Terms

Graphs, Evaluation, Measurement

Keywords

Unstructured peer-to-peer overlays, lookup systems, dependability, efficiency, metrics

1. INTRODUCTION

In unstructured peer-to-peer lookup systems, peers self organize into unstructured overlay networks. Examples to such systems include eMule, Freenet, and Gnutella. Unstructured lookup systems incur small constant overhead per single join or leave operation, and can easily support keyword searches. Chawathe et al. [3] have argued that these two features of unstructured lookup systems are highly important, as users frequently join and leave lookup sessions, and keyword searches are more popular than exact-match queries. Indeed, most of the currently deployed lookup systems are unstructured ones.

In unstructured lookup systems, the search is not structural and may fail. However, queries usually succeed in locating files due to natural file redundancy [3], that is, popular files are held by many nodes. Most unstructured peer-to-peer lookup systems and some partially-structured ones employ flooding in order to locate a searched object, at least among a subset of the nodes, e.g., super-peers in KaZaA. Due to the natural file redundancy, it is usually enough to limitedly flood the network in order to locate a searched object [7]. The main reason for using flooding is due to the high search reliability achieved by it. Nevertheless, as with all other search techniques, the dependability of flood-based search depends on the robustness of the overlay: in a highly connected overlay, flooding achieves high reliability, even in dynamic failure-prone environments, whereas in a disconnected overlay, it may fail to locate an object that is stored in the system. Flooding also incurs low latency, and can locate many copies of a searched object. However, flooding is also inefficient, as it creates a high number of duplicate search messages, i.e., multiple copies of a query may be sent to a given node by its multiple neighbors. Another problem with flooding is the difficulty to choose the appropriate TTL (Time-To-Live), which controls the flooding propagation. A high TTL achieves high search reliability but also incurs high overhead. The flooding effectiveness versus the overhead it incurs mainly depends on characteristics of the overlay. These characteristics also determine how the flooding overhead is distributed among the different nodes, and the overlay's dependability. In this paper, we define metrics capturing the above important overlay features and evaluate a number of overlays according to these metrics.

Our first metric, *c*, is the overlay's connectivity, i.e., the minimal number of disjoint paths between a pair of nodes in the overlay. This metric measures the overlay's fault-tolerance in the presence of node failures and disconnections, and hence captures the search dependability.

The second metric, *fe (flooding efficiency)*, evaluates the flooding coverage versus the overhead it incurs. Assume a query q is propagated from a random node with a TTL of i. Then, *fe(i)* is defined as $\frac{N_i}{M_i}$, where N_i is the expected number of nodes that receives q and M_i is the expected number of copies of q that are sent. A high *fe* value implies a small number of duplicates, and hence high efficiency.

The third metric, *cg (coverage granularity)*, measures the difference in the coverage when increasing the TTL by one.

SAC'06 April 23-27, 2006, Dijon, France
Copyright 2006 ACM 1-59593-108-2/06/0004 ...$5.00.

A small *cg* allows one to build an adaptive dependable lookup system that adjusts to varying failure rates, where faults include node and link failures. For example, if *cg* is small, increasing the TTL by one upon multiple query failures will increase the search reliability at the expense of a slightly higher overhead. Likewise, reducing the TTL by one upon succeeding to locate many copies of searched objects will result in overhead reduction while achieving similar search reliability. For a given TTL i, we define *dcg(i)* as $\frac{N_{i+1}}{N_i}$, where N_i is as defined above.

Our final metric, *lb (load balancing)*, evaluates how the flooding overhead is distributed among the nodes. Assume a query is initiated from a random node with a certain TTL. In a random overlay, the probability that a random node is requested to forward this query to its neighbors is proportional to the node's degree. Therefore, it is desirable that overlays would be degree-balanced, in order to incur similar overhead on all nodes. This is becoming more important now, as many ISPs have started to limit the maximal bandwidth consumption of every user. For random overlays, we define *lb* as $\frac{d_{max}}{d_{min}}$, where d_{max} (d_{min}) is the maximal (minimal, respectively) node degree.

We evaluate different graphs and overlays according to the above four metrics. We start by evaluating a Gnutella graph, which is a typical file sharing application graph. We proceed by applying our metrics on several synthetic graph structures, including a power law graph, normal random graphs, and a 3-regular random graph (a *k-regular random graph* with N nodes is a graph chosen uniformly at random from the set of k-regular graphs with N nodes). Finally, we evaluate an Araneola's overlay [8], which is a distributed approximation of a k-regular random graph. Our results show that a 3-regular random graph and a 3-Araneola overlay achieve the best (virtually identical) results.

In addition, we examine the join overhead in each of the graphs mentioned above. We observe that a Gnutella graph and an Araneola overlay incur the lowest construction and maintenance overhead: in these two graphs structures, each join (or leave) operation is handled locally and entails the sending of a small constant number of messages. In normal random graph constructions, a join or leave operation is also handled locally, though such operation incurs sending $O(\log N)$ messages, where N is the number of nodes in the system. In a power law graph, some nodes have a high degree, proportional to N, and hence joining/leaving of such nodes inevitably entails high overhead. In contrast to the above four graph structures, there are no known distributed constructions of k-regular random graphs. Therefore, with this graph structure, a single join or leave operation requires reconstructing the graph anew, and hence leads to an overhead of $\Omega(N)$ messages.

This paper proceeds as follows: Section 2 discusses peer-to-peer lookup systems. In Section 3, we describe in detail the tested graphs, and in Section 4 we evaluate these graphs according to our metrics. In Section 5, we analyze the join cost in each of the graphs. Section 6 concludes the paper.

2. RELATED WORK

Structured lookup systems, e.g., Pastry, can achieve perfect search reliability, and incur the sending of only $O(\log N)$ messages per search operation. However, such systems incur high joining overhead of $O(M \log(N))$ messages, where M is the number of objects held by the joining node. Assuming $N = 30,000$ and $M = 90$ as in Gnutella [1], a single join operation incurs a prohibitive overhead of more than $1,300$ messages. In addition, structured lookup systems do not support keyword searches, which are highly popular.

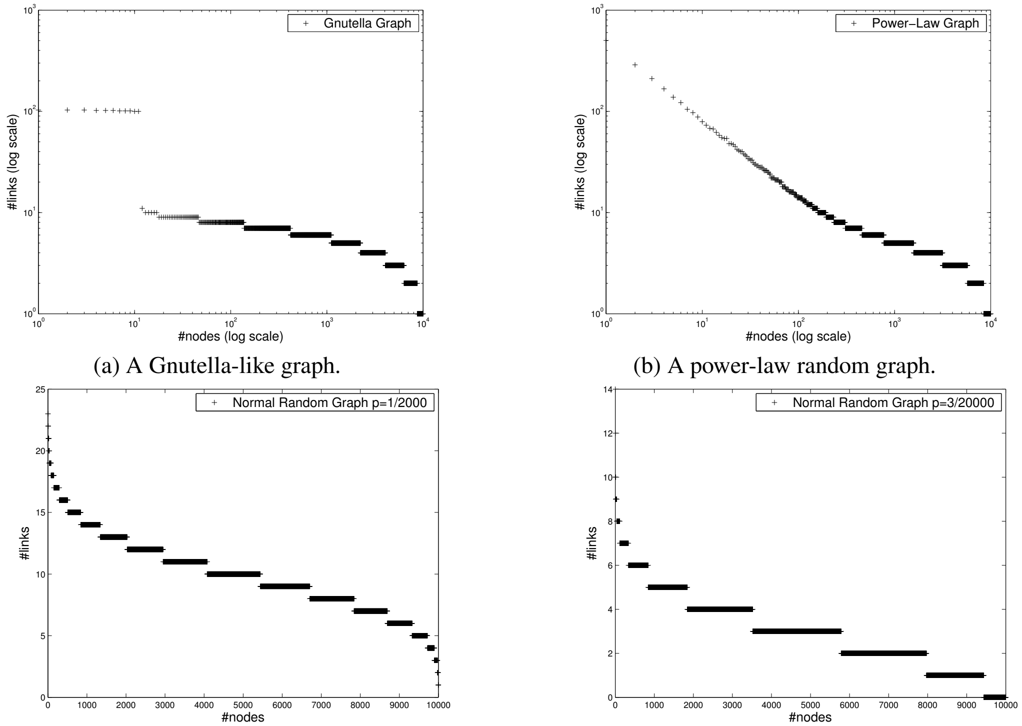
Partially structured lookup systems, e.g., KaZaA, usually rely on some infrastructure, e.g., super-peers. Therefore, such systems can achieve higher scalability compared to pure unstructured lookup systems. However, the infrastructure can be expensive to construct and maintain. Moreover, super-peers have high bandwidth consumption. In addition, infrastructure-based systems are much more vulnerable to malicious attacks than pure peer-to-peer systems. Moreover, in this paper, we show that the major problems of pure peer-to-peer unstructured lookup systems, e.g., low search efficiency resulting in a high search overhead, which lead to abandoning the pure peer-to-peer model for super-peers, can be eliminated with the use of a good overlay.

Unstructured lookup systems such as Gnutella can scale up to tens of thousands of users, without relying on any infrastructure [3]. In such systems, the search may fail. However, queries usually succeed in locating files due to natural file redundancy [3], that is, popular files are held by many nodes. Search algorithms typically used in unstructured lookup systems are based on flooding and/or random walks [7]. In a random walk, a query is forwarded to a randomly chosen neighbor at each step, until the object is found. While this search technique can incur smaller overhead than flooding, it also dramatically increases the search latency. In addition, in typical dynamic wide-area environments, a random walk usually fails to achieve a similar search reliability to that achieved by flooding. Therefore, most currently deployed peer-to-peer lookup systems employ flooding as their search algorithm. In this paper, we focus on improving the flooding efficiency in unstructured lookup systems.

Lv et al. [7], propose a search algorithm based on multiple random walks, which resolves queries for popular objects almost as quickly as flooding, while reducing the network traffic. However, this search technique is not feasible for low-replicated objects or for failure-prone settings. In addition, Lv et al. evaluate the efficiency of flooding over several graph structures. Their results show that flooding over a normal random graph achieves the best efficiency among the tested graphs. Lv et al., however, do not examine low-degree balanced graphs such as a 3-regular random graph. In this paper, we show that flooding over such a graph (or an approximation of such a graph) achieves much higher efficiency than flooding over normal random graphs. Moreover, we show that a limited flooding over a 3-regular random graph achieves similar efficiency to that achieved by random walks, while achieving higher reliability and incurring lower latency.

3. THE EVALUATED OVERLAYS

In our study, we use six undirected graph topologies. In all of the graphs, there are $10,000$ nodes. We start with a Gnutella-like graph. This graph was constructed using a node degree distribution of a real Gnutella graph taken from [9]. In order to allow a fair comparison among all the six topologies, we extrapolated the data from [9] in order to create a $10,000$



(a) A Gnutella-like graph.

(b) A power-law random graph.

(c) A normal random graph, $p = \frac{3}{20000}$.

(d) A normal random graph, $p = \frac{1}{2000}$.

Figure 1: Distribution of node degrees in four graphs. Note that we use log scale for the power-law random and Gnutella graphs, while for the normal random graphs we use a linear scale.

node graph. We kept an average of 3.4 links per-node as in [9], and a node degree distribution similar to the one in [9]. In such a graph, there is a small number of highly-connected nodes, with 100 or more links, and the majority of the nodes have a degree between 3 and 10. Similar characteristics also occur in other peer-to-peer file sharing applications [5, 3]. Fig 1(a) shows the node degree distribution of the Gnutella-like graph. We compare this graph with a power-law random graph. In this graph, the ith node chooses $\frac{w}{i^{\alpha}}$ other nodes as its neighbors, where $w = 500$, $\alpha = 0.8$, and $1 \leq i \leq 10,000$. We use this setting in order to achieve an average node degree of 3.4 links per-node, as in the Gnutella-like graph. Fig 1(b) shows the node degree distribution of this graph.

Next, we use two normal random graphs, one with $p = \frac{3}{20,000}$ and the second with $p = \frac{1}{2000}$, in which a node creates a connection with a given other node with a probability of $\frac{3}{20,000}$ and $\frac{1}{2,000}$, (respectively). The resulting average node degrees are 3 and 10, (respectively). We use the first normal random graph in order to allow a fair comparison with the previous two graphs. However, since such a graph is not connected (a normal random graph is connected if and only if $p = O(\log N)$ [2]), we also use the second connected normal random graph. Fig 1(c) and Fig 1(d) show the node degree distributions of these two graphs.

Next, we use a 3-regular random graph, in which each node is connected to three other random nodes. Finally, we use a 3-Araneola overlay [8], in which roughly 90% of the nodes have a degree of 3, while the rest have a degree of 4, leading to an average node degree of roughly 3.1.

4. THE METRICS

4.1 Connectivity

Table 1 presents the connectivity of the different graphs. A k-regular random graph and a k-Araneola graph are almost always k connected [10, 8]. Therefore, such graphs achieve high dependability even with high failure-rates, (includes node and link failures). A normal random graph is connected with high probability if p is at least logarithmic in the number of nodes [2]. Therefore, the first normal random graph is disconnected (connectivity 0). The second one has a connectivity of 1. The power-law random graph and the Gnutella-like graph have a connectivity of 1, as several nodes in these graphs have a degree of 1. Such nodes are very likely to be disconnected from the overlay graph. For a given number of links, we observe that a 3-regular random graph and a 3-Araneola overlay achieve much higher connectivity than a Gnutella graph, a power-law random graph, and normal random graphs, due to their regular structure. In fact, a 3-regular random graph and a 3-Araneola overlay, in which the average node degree is roughly 3, achieve higher connectivity than a normal random graph with an average node degree of 10.

4.2 Flooding Efficiency

We now evaluate the flooding efficiency in all the graphs except the normal random graph with $p = \frac{3}{20,000}$, as this graph is not connected. For each graph, we run the flooding protocol $10,000$ times, one time from each node, and we calculated the average flooding efficiency. We report about our results in Fig. 2, and Fig. 3 shows the coverage achieved with each TTL.

In a power-law random graph and in a Gnutella-like graph, starting from a TTL of 4, the flooding efficiency, i.e., the coverage divided by the overhead, is poor. This is due to the pres-

Graph	Connectivity
3-regular random graph	3
3-Araneola overlay	3
Normal random graph($p = \frac{3}{20{,}000}$)	0
Normal random graph($p = \frac{1}{2{,}000}$)	1
Gnutella-like graph	1
Power-law random graph	1

Table 1: Connectivity: A 3-regular random graph and a 3-Araneola overlay has a connectivity of 3. The rest of the graphs have a connectivity of 1 or 0.

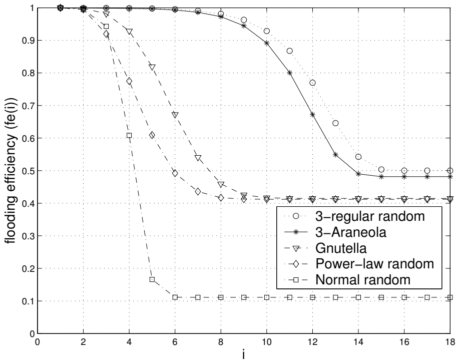


Figure 2: Flooding efficiency: for effective TTLs, a 3-Araneola overlay and a 3-regular random graph achieve a near to perfect search efficiency. Other graphs achieve much lower search efficiency.

ence of high-degree nodes in both of the graphs, which create and receive many duplicate search messages. A similar phenomenon occurs in the normal random graph with $p = \frac{1}{2{,}000}$, as the degrees in such a graph range from 1 to 23. In contrast, in low degree balanced graphs such as a 3-Araneola overlay and a 3-regular random graph, the flooding efficiency is very high. For small TTLs ($\leq$ 8), the flooding efficiency of the 3-regular random graph and the 3-Araneola overlay is very close to one. Hence, for such TTLs, flooding is as efficient as random walks. Fig. 3 shows that with a TTL of $7/8/9$, flooding over a 3-Araneola overlay and a 3-regular random graph reaches, on average, to $485/989/1957$ (roughly 4.85%/9.9%/20%) and $376/739/1424$ (roughly 3.8%/7.4%/14%) nodes, respectively. Therefore, with a 3-Araneola overlay and a 3-regular random graph, it is possible to reach any desired portion of the nodes efficiently; this is thanks to their good coverage granularity, as discussed in the next section.

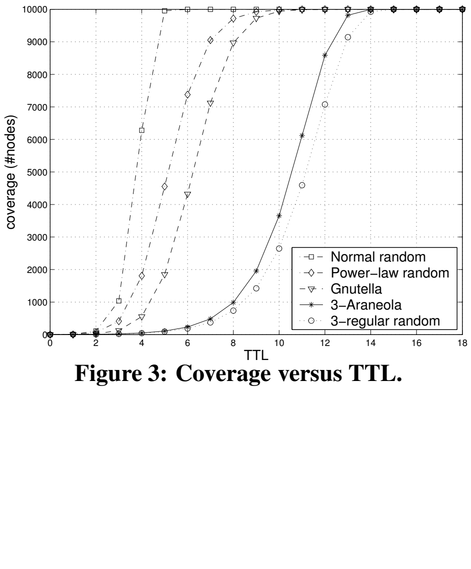


Figure 3: Coverage versus TTL.

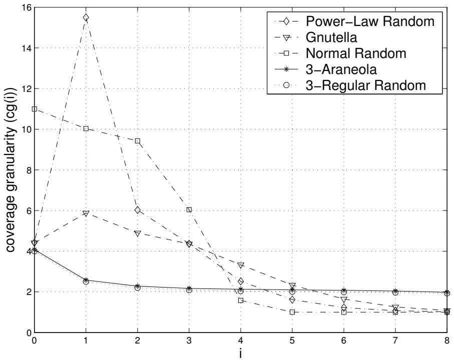


Figure 4: Coverage granularity: a 3-Araneola overlay and a 3-regular random graph achieve a good *cg* value for all TTLs. In the rest of the graphs, *cg(i)* is very high for small (effective) TTLs and low for high (ineffective) TTLs, in which the flooding efficiency is poor.

4.3 The Coverage Granularity

Recall that *cg(i)* is defined as $\frac{N_{i+1}}{N_i}$, where N_i is the expected number of nodes that receive a query that originates from a random node with a TTL of i. Fig. 4 shows *cg(i)* for the five graphs evaluated in the previous section. As the figure shows, a 3-Araneola overlay and a 3-regular random graph have a low (virtually identical) *cg(i)* value for all TTLs. In addition, in these two graphs, *cg(i)* is very similar for all the TTLs. This is due to the fact that k-regular random graphs are good expanders. Therefore, in these two graphs, one can adapt the search dependability and overhead according to the failure rate. In contrast, in the rest of the graphs, *cg(i)* is very high for small (effective) TTLs and low for high (ineffective) TTLs. In addition, in these graphs, the low coverage granularity is achieved only when the flooding efficiency is poor (see Section 4.2).

4.4 Load Balancing

It is desirable that the flooding overhead would be distributed equally among all nodes. Recall that for a random overlay, we define the load balancing (*lb*) as $\frac{d_{max}}{d_{min}}$, where d_{max} (d_{min}) is the maximal (minimal, respectively) node degree. In the normal random graph with $p = \frac{3}{20{,}000}$, we ignore nodes with degree 0, as they are not connected to the overlay. Table 2 shows the *lb* value of the different graphs. The 3-regular random graph achieves perfect load balancing, i.e., 1. Next, the 3-Araneola overlay achieves excellent load-balancing: $\frac{4}{3}$. The two normal random graphs have *lb* values of $\frac{14}{1}$ and $\frac{23}{1}$. In such graphs, assuming queries are distributed uniformly, the overhead incurred on a highly-connected node may be $O(\log N)$ times the overhead incurred on a low-connected node, as in such graphs a connected node's degree is between 1 and $O(\log N)$. In the Gnutella-like graph and the power-law random graph the load balance is even worse, as the overhead incurred on a highly-connected node can be two orders of magnitude greater than the overhead incurred on a low-connected node.

5. THE JOIN COST

The results of Section 4 have shown that the 3-regular random graph and the 3-Araneola overlay are the best overlays among the tested graphs. We now examine the cost/feasibility

Graph	$lb = \frac{d_{max}}{d_{min}}$
3-regular random graph	1
3-Araneola overlay	4/3
Normal random graph($p = \frac{3}{20,000}$)	14/1
Normal random graph($p = \frac{1}{2,000}$)	23/1
Gnutella-like graph	103/1
Power-law random graph	502/1

Table 2: Load balancing: a 3-regular random graph achieves perfect load balancing of 1. A 3-Araneola overlay achieves a good load balancing of $\frac{4}{3}$. The rest of the graphs achieves poor load balancing.

of distributed constructions of the tested graphs. Specifically, we examine the join overhead in each of the graphs. We evaluate this overhead in two ways. We first assume the existence of a membership service that maintains at each node a small number of random node identities. Examples to such scalable membership services can be found in [4, 8]. Next, we evaluate the join overhead without relying on the existence of a membership service. In this case, we assume that a joining node knows the identity of some other node that is currently in the system. We assume, however, in this case that a random walk of $O(\log N)$ steps from a given node reaches a random node. Law et al. [6] have shown that this assumption is true for expander graphs. Note that a scalable membership service amortizes the logarithmic cost of knowing a random node by aggregating membership information, and hence it is more efficient than a random walk for retrieving random node identities.

Table 3 shows the join cost for each graph in both cases. In a 3-Araneola overlay, a join operation requires sending $3k = 9$ messages, assuming the existence of a membership service. In the absence of such a service, connecting to a random node requires sending $O(3 + \log N) = O(\log N)$ messages. In a Gnutella graph, the overheads are similar to the overheads above. However, for a high degree node, i.e., one that has 100 or more links, the leave overhead is very high, as such a node is connected to many other nodes.

In a normal random graph and in a power-law random graph, given a membership service, the join cost is the node's degree. In a connected normal random graph this degree is logarithmic in the number of nodes in the system, and in a power-law random graph a node's degree can be $O(N)$. Therefore, assuming the existence of a membership service, the joining overhead in a (connected) normal random graph and in a power-law random graph is $O(\log N)$ and $O(N)$, respectively. In the absence of a membership service, these overheads need to be multiple by $O(\log N)$, the overhead for retrieving a random node. Finally, in a k-regular random graph, since no distributed constructions of such a graph are known, a join operation requires the reconstruction of the entire graph, leading to a prohibitive overhead of $\Omega(N)$ messages.

Graph	The join cost (with membership service)
3-Araneola overlay	9
Gnutella-like graph	constant
Normal random graph	$O(\log N)$
Power-law random graph	$O(N)$
3-regular random graph	$\Omega(N)$

Graph	The join cost (without membership service)
3-Araneola overlay	$O(\log N)$
Gnutella-like graph	$O(\log N)$
Normal random graph	$O(\log^2 N)$
Power-law random graph	$\Omega(N)$
3-regular random graph	$\Omega(N)$

Table 3: The join cost: A 3-Araneola overlay achieves the lowest join cost.

6. CONCLUSIONS

We have defined metrics for evaluating unstructured overlays for peer-to-peer lookup systems. These metrics capture the search dependability and efficiency, the granularity at which one can control the tradeoff between the two, and also the fairness. According to these metrics, we evaluated various overlays. Our results show that a 3-regular random graph and a 3-Araneola overlay achieve the best results in term of all four metrics. Moreover, using such overlays eliminates the main drawback due to which unstructured overlays were abandoned, namely the search inefficiency. In fact, with such overlays, one can reach up to 20% of the nodes with almost perfect search efficiency.

As opposed to a 3-regular random graph, a 3-Araneola overlay supports dynamic user behavior. In such an overlay, each single join or leave operation is handled locally, and incurs the sending of only 9 messages on average (or $O(\log N)$ messages in the absence of a membership service). Therefore, we conclude that a 3-Araneola overlay is an excellent solution for a flooding-based peer-to-peer lookup system.

7. REFERENCES

[1] E. Adar and B. A. Huberman. Free riding on gnutella. *First Monday*, Sept. 2000.

[2] B. Bollobas. Random graphs.

[3] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making gnutella-like p2p systems scalable. In *ACM SIGCOMM*, August 2003.

[4] P. T. Eugster, R. Guerraoui, S. B. Handurukande, A. M. Kermarrec, and P. Kouznetsov. Lightweight probabilistic broadcast. *ACM Trans. Comput. Syst.*, 21(4):341–374, November 2003.

[5] K. Gummadi, R. Dunn, S. Saroiu, S. Gribble, H. Levy, and J. Zahorjan. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In *SOSP*, 2003.

[6] C. Law and K. Siu. Distributed construction of random expander networks. In *IEEE Infocom*, April 2003.

[7] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Search and replication in unstructured peer-to-peer networks. In *Proceedings of the 16th international conference on Supercomputing*, pages 84–95. ACM Press, 2002.

[8] R. Melamed and I. Keidar. Araneola: A scalable reliable multicast system for dynamic environments. In *IEEE NCA*, 2004.

[9] 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.

[10] N. Wormald. Models of random regular graphs. *Surveys in Combinatorics*, 276:239–298, 1999.