Push-Pull Gossiping for Information Sharing in Peer-to-Peer Communities

Mujtaba Khambatti
Arizona State University
Tempe, AZ 85287-5406
mujtaba@asu.edu

Kyung Ryu
Arizona State University
Tempe, AZ 85287-5406
kdryu@asu.edu

Partha Dasgupta
Arizona State University
Tempe, AZ 85287-5406
partha@asu.edu

Abstract

We expand the scope of peer-to-peer (P2P) systems to include the concept of "communities". Communities are analogous to interest groups and can overlap. We use communities as a more natural arrangement of distributed systems. This paper focuses on a novel, push-pull gossiping technique to improve decentralized information dissemination and search within a P2P community. To facilitate an efficient gossiping operation, a distributed discovery algorithm first identifies highly influential peers (called seers) in a community. Then, the push phase multicasts information to these seers so that peers can easily and quickly retrieve this information via a pull phase. Our experiments show that pushing gossip information to only a small number of seers allows a large percentage of peer members to obtain (pull) the information within just two hops.

Keywords: Distributed computing, peer-to-peer computing, distributed information sharing, peer communities

1. Introduction

Current peer-to-peer (P2P) systems are often targeted for information sharing [1], [2], file storage [1], [3], [4], [5], searching [6], [7], and indexing [8], [9] by using an overlay network. The ability of P2P systems to harness vast amounts of storage from a scalable collection of peers and its emphasis on de-centralization and lack of a central authority seems to empower everyday home computer users by allowing them to share a portion of the authority. These systems also have other interesting technical characteristics such as self-organization and adaptation [10].

P2P systems are distributed systems in which logically distinct computing elements called peers that have comparable roles and responsibilities, communicate information, share or consume services and resources amongst each other [11]. Throughout this paper, we use the term 'peer' to refer to a network-addressable computing element, like a desktop PC, a personal digital assistant, a networked printer, etc.

In our work, we expand the scope of P2P systems to include the concept of "communities". Communities are analogous to interest groups. They are formed implicitly based on common interests and can overlap [12]. A collection of peer nodes forms a P2P community if they share one or more common interest attributes. Interest attributes are a reflection of the activities and interests of a peer's owner. For instance, the owner can express that she is interested in French wines and house decoration. Such expressions are personal declarations. Also, her repeated web search queries to find "K-12 education in Arizona" can be used to provide implicit information about her interests. These attributes are either explicitly provided by a peer or implicitly discovered from past queries.

We use communities as a more natural arrangement of distributed systems. They are useful in structuring the information storage space, discovering resources, and pruning the search space. They also allow for better dissemination of useful information [11].

This paper focuses on a novel push-pull communication technique to improve information search within a P2P community. Initially, certain properties of the P2P community are discovered. These properties include the approximate number of members and the information about the member peers. The repeatable push-pull phases of gossiping disseminate information within a community amongst chosen peers whenever required. Both discovery and gossiping techniques are efficient, de-centralized, robust and highly scalable.

The rest of this paper is organized as follows. Section 2 describes the motivations for a new information search technique in a community-based P2P network. In Section 3, we explain how our P2P network is formed such that it exhibits characteristics of a small-world, scale-free network. Then, in Section 4, we present our distributed discovery of seers and push-pull gossiping along with experiment results. Finally, Section 5 concludes this paper with a summary.

This research is partly funded by grants from AFOSR, DARPA, Microsoft and NSF.

2. Motivation

P2P systems are considered to be completely de-centralized and can also be dynamic. This characteristic makes them very attractive as system solutions to everyday users (the "little people"), like home users, small-scale networks, and ubiquitous computing environments, who now will have the ability to choose their own policies, roles, and responsibilities and change them autonomously.

Searching for information is one of the key challenges in P2P systems. Centralized searching has the downside that the central authority controls the indexing and presentation of the information. P2P searching allows anyone to put up information in the search index and then cooperatively search the P2P space. P2P searching techniques include flooding, directed flooding, iterative deepening [6], and directed BFS [6].

We explained earlier that we consider P2P communities to be a natural way for arranging distributed systems. Consider a digital library built out of a collection of peers in which each peer owns a set of books that it is willing to share with other peers[1]. The subjects of the books owned by a peer form its set of interests. Peers are implicitly grouped into communities based on the common interests they share. Because a peer could own books spanning a variety of subjects, a peer could be a member of multiple communities.

If the Computer Science and Medical communities were disjoint, then search operations for medical information by a node belonging to the Computer Science community would not produce any results. However, if the communities were linked at some point, lets say Q (Q belongs to both communities), then medical information would be found, but at a great search expense, since on the average half the Computer Science community would be searched before a node from the Medical community is found.

To mitigate such problems, we need a community-based query propagation method. Thus, to provide efficient searching, it is better to search for one (or more) target communities, irrespective of the current membership of the searching node.

Our work focuses on efficient methods to discover and then use these self-configuring communities for community-based search query propagation in a populated P2P space. In addition, we propose efficient algorithms to facilitate the management of quickly changing community structures (such as dynamic communities, failures, and mobile nodes).

[1] Assume these are non-copyrighted works.

3. P2P Network Formation

A P2P network can be thought of as a graph where the nodes represent peers and the edges represent the associations or *links* between two peers. In this section, we explain how such a network can be constructed. In the beginning, we explain what we mean by peer links and how they are setup between peers. We then provide the rules for a new peer joining a P2P network.

3.1. Peer Links

Suppose a peer, belonging to domain abc.com, claims the attribute "Computer Science" (referred to as *claimed attribute*). This peer is essentially isolated unless it *a priori* knows about the other members of the Computer Science community or the other members of the abc.com community.

Flooding and querying a central server are two solutions to the isolation problems; however, the first is expensive and the second violates the self-configuring tenet of the P2P structure. Hence, we propose the use of an overlay network based on links.

When peer 'X' is born, it needs to have one or more logical neighbors[2]. If it has three neighbors, 'A', 'B', and 'C', then we say that it has three links, X→A, X→B, and X→C. Unlike overlay networks used by some peer-to-peer implementations, our link based network is not based on peer node names, but on user selected neighbors.

Now we discuss how these links are created. Peer 'X' links to peer 'A' if 'A' is:

i) A special peer chosen by the domain for P2P links.
ii) A peer, known to 'X' that it trusts - a friend.
iii) A well-known peer that belongs to many communities 'X' is interested in.

For a novice/new peer, (i) may be most appropriate. As 'X' ages, it finds other peers and adds these links to improve search speed and information access. The linkages are similar to friendships in real life, or to http links in the Web and are directed by humans.

Here, we present two definitions from our earlier work [11] that relates to links between peers:

1. **Outlink Weight:** The weight given to each claimed attribute based on the percentage of outgoing connections from a peer that can reach, after at most one indirection, other peers claiming the same attribute.
2. **Inlink Weight:** The weight given to each claimed attribute based on the percentage of incoming connections to a peer that arrive directly from other peers claiming the same attribute.

[2] Neighbors are directly linked peers.

3.2. Rules for joining peers

Although various systems, like social networks [13, 14], the World Wide Web [15, 16], and Gnutella [17] demonstrate scale-free and small world network characteristics, there is no guarantee that a new P2P system such as a digital library would form a small-world network.

We solve this uncertainty by enforcing certain rules on new peers that want to join the P2P system. By virtue of these rules, the P2P system that forms is a small-world network and also exhibits a power-law (or scale-free) characteristic for the distribution of the number of neighbors of each peer.

We derive inspiration from the work of M. Steyvers and J. Tenenbaum [18] who reported that the network growth process known as preferential attachment presented in [19] yielded much lower values of clustering coefficient for certain types of networks. Steyvers and Tenenbaum proceeded to describe a model for the growth of semantic networks that resulted in both small world networks and scale-free structures.

We extend the domain of their model to a P2P network involving peers and links. The rules for a new peer 'X' to

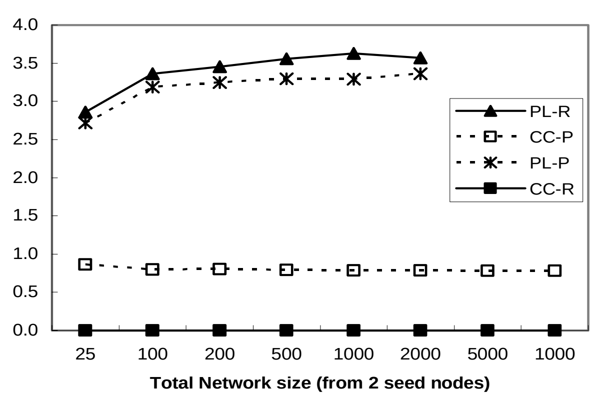

Figure 1. The graph shows values of clustering coefficient (CC) and characteristic path length (PL) for various network sizes generated from an initial seed of 2 nodes. The suffix 'P' indicates P2P networks and the suffix 'R' indicates Random networks.

join a P2P system are:

1. Peer 'X' selects a single peer 'A' from a list of known peers (see Section 3.A) that are currently members of the P2P system, such that 'A' is one of the more well-connected peers, i.e. it has on the average more links to other peers within the P2P system than the other peers from the list.
2. Peer 'X' creates links to N neighbors of 'A' such that the neighbors of 'A' that have more links to other peers are chosen with a higher probability than other neighbors.

Initially, a very small number N+1 (we ran successful simulations with 2 peers, 5 peers, and 10 peers) of fully

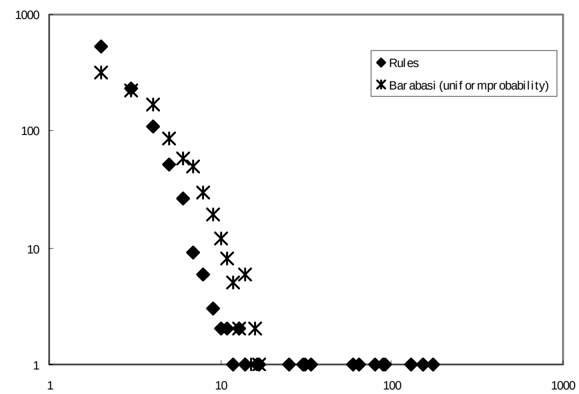

Figure 2. The graph shows the power-law distribution of the frequency (X-axis) Vs degree (Y-axis) plot in a network size of 1000 nodes formed with 2 seed nodes. We compare our "Rules" method with the well-known "Barabasi" technique that uses preferential attachment. The power-law exponents are –1.13 using Rules, and –2.88 using Barabasi's technique.

connected peers is chosen. These peers are the starting seed of the P2P system and would be set up in an explicit manner by a group or an individual person. Any peer that wants to join the P2P system henceforth needs to follow the rules described above.

This algorithm allows a P2P system to form and then grow in a scalable manner. Our simulations showed (see fig.1) that the resulting networks generated had a low characteristic path length comparable to a corresponding random network; however, the value of the clustering coefficient remained nearly as high as regularly connected networks (which have clustering coefficient equal to 1).

4. Peer-to-Peer Communication

After we run our algorithms detailed in [11] that uses interest attributes for assisting in the formation of communities, peers will discover the communities that they participate in. However, since interest attributes are constantly changing values, the formation of communities needs to occur on a regular basis to keep the P2P system up-to-date and to keep the peers subscribed to the most suitable, existing communities. Then, again, a periodic increase in communication messages might not be suitable for low bandwidth networks, as regular communication will be affected by this increased traffic.

Therefore, instead, we opt for *Distributed Discovery* followed by push-pull *P2P Gossiping*.

4.1. Distributed Discovery and Push-Pull

Our push-pull approach reduces the number of messages that are sent within the P2P network. Prior approaches for information dissemination or information retrieval within a P2P network have tried to send out messages through all or selected neighbors and up to a certain depth. Unlike these approaches, our technique involves a discovery phase to gather data on peers that would be interested in receiving certain information. Thereafter, the push phase of P2P Gossiping involves a multicast of that information to specially selected peers (called seers) based on the discovered data. As long as the discovered data is available and recent, the push phase can be repeated with new information numerous times. Whenever required, a peer will retrieve the information from a nearby seer via a pull phase.

We show that our Distributed Discovery is a low overhead, simple protocol to identify seers and terminates easily. Our algorithm for the push phase makes gossip information available to a large percentage of interested peers within a very short number of links. In addition, these two techniques facilitate the management of quickly changing community structures via undirected intra-community communication.

4.2. Distributed Discovery of Seers

Peers are classified based on their involvement value [11] with respect to the community. The involvement of a peer in a community is the average of all outlink weights (see section 3.A) corresponding to the common interest attributes shared by members of that community. In other words, for peer 'X', involvement is proportional to the number of peers from the neighborhood[3] of 'X' to which there are outgoing links.

Peers are called seers if they have a higher value of involvement due to their links (direct or indirect) to more peers claiming the same attribute(s). Thus, information stored on the seers will be available to more peers within the community.

Now, we propose a simple Distributed Discovery algorithm to find the seers that will be used for the push phase. The algorithm will gather involvement values from most peer members of the community in an efficient manner.

Lamport's distributed snapshot algorithm [20] is not relevant to solve this problem because of two reasons:

1. The Lamport algorithm makes an assumption of FIFO channels. This might not be true in our P2P network due to the use of overlay links.
2. Termination detection of the Lamport algorithm requires knowledge of the total number of nodes in the distributed system, and these nodes have to remain alive throughout the execution of the algorithm. This is not practical in a dynamic P2P network where peers can appear and disappear at random.

Therefore we provide a solution to the Distributed Seer Discovery problem. Consider a peer that is a member of community 'Ca' by virtue of claiming an interest attribute 'a'. We employ a vector (fig. 3) that an initiating peer sends to every peer in its neighborhood claiming the same attribute 'a'. Any peer can be the initiating peer. Eventually, the vector would have traveled to all peer members of the same community. Every peer receiving the vector would append its information to the vector and send it forward with the same criteria used by the initiating peer. A unique vector identifier alerts peers to drop duplicates. The peers, that have no neighbors yet to receive the vector, construct an end message with the vector and send it to the initiating peer whose identity can be obtained from the first element of the vector.

Vector ID	Peer ID	Peer Involvement	…	…	…

Figure 3. The initiator sends a vector with the above fields. Receiving peers append their information and send the vector onwards or back to the initiator.

The initiating peer waits a certain amount of time and receives end messages. The union of all end messages will provide information on the members of community

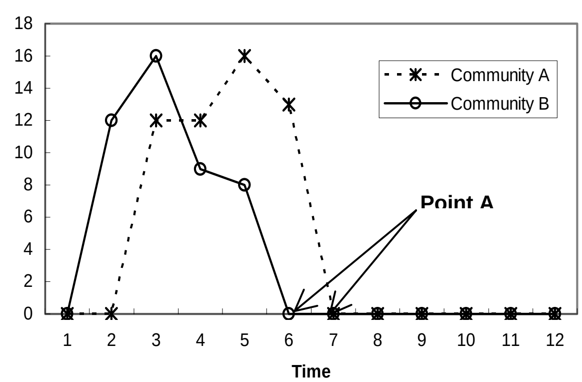

Figure 4. The graph shows the number of end messages received by the initiator of two separate communities. The network size was 1000 nodes.

[3] The neighborhood of a peer includes neighboring peers and each of their neighbors.

'P' to the initiator. Since all peers know the identity of the initiator, they can obtain this information in a deterministic time if needed. This leaves one question unanswered: how to set the waiting period for the initiating peer.

In our experiments, we found that the initiating peer received end messages with the frequency graph shown in fig. 4. Therefore, by making the initiating peer wait 't' cycles after the frequency drops off to zero (Point A), most of the end messages can be collected. Here, 't' is calculated as the time it takes for the frequency to reach its peak from zero. If more messages arrive during this wait time, Point A is reset and the value 't' is recalculated for the new peak frequency.

Initiator:

```
Start-prog
Create vector 'v' for community 'P'
Insert my information
Send 'v' to direct neighbors claiming 'P'
Send 'v' to 2nd-deg neighbors claiming 'P'

/* Wait till freq peaks and then drops */
Wait to receive end messages
End-prog
```

Receiving Peer:

```
Start-prog
Receive vector 'v'

If I have already received 'v'
   Send NACK to sender of the 'v'
   End-prog
Else
   Send ACK to sender of the 'v'
End-if

Insert my information
List neighbors (up to 2-deg) claiming 'P'
Remove sender of 'v' from list

If list has peer identities
   Foreach peer in list
      Send 'v' to peer
      Receive acknowledgement from peer
   End-for
Else
/* This means I am at the end */
   Create end message with 'v'
   Send to Initiator
   End-prog
End-if

If NACK received from all peers
/* This means I am at the end */
   Create end message with 'v'
   Send to Initiator
End-if
End-prog
```

Figure 5. Pseudo-code of the Distributed Discovery algorithm. The first code is executed by the initiator of the algorithm. All other peers execute the second code on receipt of the vector.

4.3. P2P Gossiping

For undirected intra-community information dissemination, we propose P2P Gossiping. P2P Gossiping is a push-pull approach that is resilient and does not critically depend on any single peer or message. It involves communication (gossiping) between seers to achieve an objective similar to the case of rumor spreading [21]. However, the two major differences between our method and rumor spreading is that each peer does not have an a priori knowledge of the number of peers that exist, and peers are not at random.

As indicated earlier, our algorithm utilizes the seers within a community to carry the information or updates so that it will be available to most peer members.

We propose a solution to the P2P Gossiping technique that involves the initiator of the distributed discovery algorithm from Section 4.B. At the end of the Distributed Discovery algorithm, the initiator will know certain properties of the community, such as the values of involvement that was inserted by each peer into the vector. The initiator will then create a set of seers by picking the peers that have the highest involvement values. By the definition of involvement, the peers in this set will have more community members in their neighborhood when compared to the neighborhoods of peers that are not in the set.

We show that by correctly choosing the size of the seers set, a majority of community members will lie within the neighborhood of at least one peer from the set. Therefore information that is sent to the initiator can be multicast to the members of the seers set so that it is available (if required) to the majority of community members within their neighborhoods (i.e.1 or 2 peers away).

We conducted experiments on 3 different P2P networks to determine the behavior of our push algorithm. The initiator selected X% of the peer members with the highest involvement values as seers and sent some information to those seers (push) with the hope that this information will be available to Y% of the remaining community members within 2 hops. In the graph below, we show the relationship between X and Y.

On the average, we found that pushing information out to only 10% of specially selected community members (seers) makes that information accessible to approximately 80% (or more) of the remaining members within their own neighborhoods.

Finally, in order to obtain the information, a peer will have to pull it from one of the seers that contain the information. We see in fig. 6 below that a majority of peers will find a seer within just 2 links away. The push phase will be initiated whenever there are enough updates on available information, whereas the pull phase will be started by new information requests from any peer.

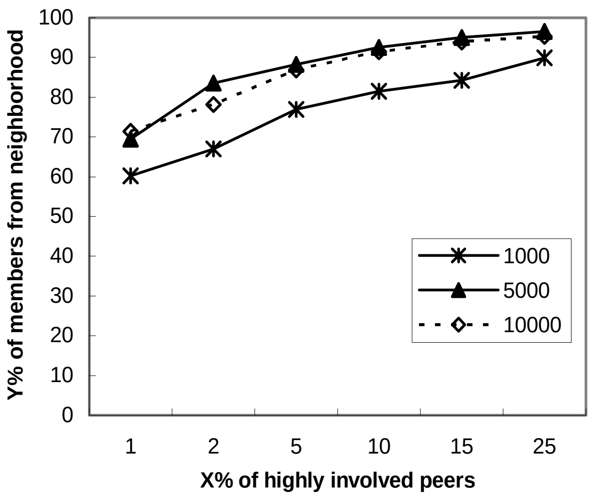

Figure 6. The graph shows the relation between X and Y for 3 different network sizes (1000, 5000, and 10000 nodes). The values are an average for tests conducted on all the communities within those networks.

5. Conclusions

A distributed system, such as a P2P network, can be naturally organized into communities. P2P communities are dynamic and implicit structures comprising of peers that share interest attributes, and they are useful for efficient search or information dissemination operations within the network.

In this paper, we provide a set of rules for peers to join a P2P network such that the network will always exhibit certain properties like small world behavior and power-law distribution. We use these rules to build a P2P network simulator so that our simulations mirror real peers as closely as possible.

Our earlier work has already shown how communities can be formed and discovered through an exchange of interest attributes amongst peers [11]. Since peers are dynamic entities of a P2P network, changes like new or deleted interest attributes need to be communicated with as many interested peers as possible in an efficient manner. This form of communication is not intended for any particular peer but is meant for all interested peers in the network.

In this paper we proposed a novel form of undirected intra-community communication using two phases, push and pull, that are preceded by a Distributed Discovery operation. Combined with the distributed discovery of seers, both phases help with information dissemination within the community. The Distributed Discovery algorithm involves gathering information on peer members of a community. We provided a dynamic scheme to determine a termination point for this algorithm.

With the information gleaned from the distributed discovery, peers can execute our gossiping protocol using the push-pull phases. We ran experiments to show that pushing the information to only a small number of specially chosen peers allowed a large percentage of peer members of a community to obtain (pull) the gossip information from within their neighborhoods.

6. References

[1] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, “Freenet: A distributed anonymous information storage and retrieval system,” in *Workshop on Design Issues in Anonymity and Unobservability*, Berkeley, CA, July 2000, pp. 311–320.

[2] K. Ranganathan, A. Iamnitchi, and I. Foster, “Improving Data Availability through Dynamic Model-Driven Replication in Large Peer-to-Peer Communities,” in *Global and Peer-to-Peer Computing on Large Scale Distributed Systems Workshop*, Berlin, Germany, May 2002.

[3] Napster, Available: http://www.napster.com/

[4] Gnutella, Available: http://www.gnutelliums.com/

[5] P. Druschel and A. Rowstron, “Past: Persistent and anonymous storage in a peer-to-peer networking environment,” in *Proc. the 8th IEEE Work. Hot Topics in Operating Systems* (HotOS), Germany, May 2001, pp. 65–70.

[6] B. Yang and H. Garcia-Molina, “Efficient Search in Peer-to-Peer Networks,” in *Proc. Intl. Conf. On Distributed Computing Systems* (ICDCS), Vienna, Austria, 2002.

[7] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman, “Search in power-law networks,” in *Physical Review E*, vol.64(4), 046135, 2001.

[8] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, “Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications,” in *Proc. ACM SIGCOMM*, 2001, pp. 149-160.

[9] A. Rowstron and P. Druschel, “Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems,” in *Proc. 18th IFIP/ACM Int’l. Conf. Distributed Systems Platforms* (Middleware), 2001.

[10] G.W. Flake, S. Lawrence, C.L. Giles, and F.M. Coetzee, “Self-Organization and Identification of

Web Communities," in *IEEE Computer*, vol. 35(3), pp. 66-71, March 2002.

[11] M. Khambatti, K. Ryu, and P. Dasgupta, "Efficient Discovery of Implicitly Formed Peer-to-Peer Communities," in *Int'l. Journal of Parallel and Distributed Systems and Networks*, vol. 5(4), 2002.

[12] M. Khambatti, K. Ryu, and P. Dasgupta, "Peer-to-Peer Communities: Formation and Discovery," in *14th IASTED Int'l. Conf. Parallel and Distributed Computing Systems* (PDCS), Cambridge, MA, USA, November 2002.

[13] J. Guare, *Six Degrees of Separation: A Play*, New York: Vintage Books, 1990.

[14] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," in *Nature*, vol. 393, pp. 440-442, 1998.

[15] R. Albert, H. Jeong, and A. Barabasi, "Diameter of World-Wide Web," in *Nature*, vol. 410, pp. 130--131, Sept. 1999.

[16] A. Barabási, R. Albert, H. Jeong, and G. Bianconi, "Power-law distribution of the World Wide Web," in *Science*, vol. 287, 2115a, 2000.

[17] M. Jovanovic, F. S. Annexstein, and K.A. Berman, "Scalability Issues in Large Peer-to-Peer Networks-A Case Study of Gnutella," University of Cincinnati, Technical Report, 2001.

[18] M. Steyvers and J. B. Tenenbaum, "The large-scale structure of semantic networks: statistical analyses and a model of semantic growth," unpublished.

[19] A. Barabási and R. Albert, "Emergence of scaling in random networks," in *Science*, vol. 286, pp. 509-512, 1999.

[20] L. Lamport and M. Chandy, "Distributed Snapshots: Determining Global States of a Distributed System," in *ACM Trans. Computer Systems*, vol. 3(1), pp. 63-75, Feb. 1985.

[21] R. Karp, S. Shenker, C. Schindelhauer, and B. Vocking, "Randomized rumor spreading," in *41st Symp. Foundation on Computer Science* (FOCS), Redondo Beach, CA, November 2000.