Volume 2, Issue 2, February 2012 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Self-sustaining Broadcast for Fully Decentralized Peer-to-peer Networks

[1]V.Ravikumar
Research Scholar
Dept Of Cse ,
Bharath University

[2]Dr.M.Rajani,
DirectorResearch,
Bharath university

Abstract-Recently, there has been a growing interest in peer-to-peer networks, sparked by the popularity of file sharing applications such as Napster and Gnutella. A typical characteristic of a peer-to-peer system is that all the nodes are equal participants in the network. Gnutella is an example of a 'pure' peer-to-peer system, being fully distributed where all nodes are equal and no special nodes with facilitating or administrative roles are required. Due to its fully decentralized nature, Gnutella implements its services like searching and peer discovery via application-level broadcast. For this, messages are routed through Gnutella's overlay network by means of flooding. The high cost of flooding limits the scalability of fully distributed peer-to-peer systems like Gnutella. To overcome the problem of scalability, we propose Rumor mongering (also known as Gossip) as a cost-effective alternative to flooding to implement application layer broadcast in decentralized peer-to-peer networks. We introduce a new version of Rumor Mongering, called Deterministic Rumor Mongering, which makes use of the fact that most peer-to-peer network topologies display a power-law distribution in their node degree, to implement a more intelligent routing strategy. Using simulation, we show that Deterministic Rumor Mongering performs broadcast at a significantly lower cost than flooding, at the cost of a slightly reduced reliability and increased time of completion of the broadcast.

1. Introduction

Recently, there has been a lot of activity in the area of peer-to-peer networking, sparked by the popularity of file sharing applications such as Napster [1], Freenet [20] and Gnutella [2]. Although the exact meaning of the term "peer-to-peer" is very debatable, these systems typically depend on a number of voluntarily participating nodes contributing resources to create some form of infrastructure. The most well known service such a peer-to-peer infrastructure can provide is certainly multimedia file-sharing, but other systems with completely different applications such as SETI@home [3] are also labeled as peer-to-peer. Even among the peer-to-peer systems aimed at file sharing, there exist big differences in terms of architecture and implementation of key mechanisms such as searching. Napster, for instance, relies on a central indexing server to locate files that are shared. Gnutella implements searching in a fully decentralized and distributed manner, without any special nodes with facilitating or administrative roles. In that sense, Gnutella can be considered a more "pure" peer-to-peer system.

The advantages of a decentralized architecture are clear. The absence of a central node that represents a single point of failure, results in increased fault tolerance and robustness.

Furthermore, the only state information a Gnutella node needs to store is the address of its immediate neighbors. This allows Gnutella to deal with the very dynamic nature of a typical peer to peer network, where nodes frequently join and leave the network. The problem of convergence due slow propagation of routing information is therefore avoided. Gnutella provides two types of services via its overlay network: searching for files and peer discovery. These services are implemented through broadcasting messages across the network.

This application level broadcast is implemented with flooding as the underlying routing mechanism, where every node acts a router and forwarder for messages sent by other nodes. This obviously raises the question of cost and scalability. Individual nodes can be swamped with messages to forward and might not have enough resources such as CPU cycles, memory or bandwidth required for the task. This problem has also been observed in the real Gnutella system [4], where the fact that nodes could not keep up with the rate of messages to be forwarded, has lead to a fragmentation of the network. It is therefore important for the successful deployment of decentralized peer-to-peer networks that the problem of cost and scalability is studied and more cost-effective methods for application level broadcast are explored.As an efficient alternative to flooding to implement application-level broadcast in a fully distributed manner, we propose the use of *Rumor Mongering* (also known as Gossip) as a routing method. Rumor Mongering trades off reliability and speed for a reduction in cost. First, we study the performance and cost of Rumor Mongering, which chooses the neighbors to forward messages to uniformly at random. Then we introduce a new variant of the protocol, which we call.

Deterministic Rumor Mongering. Deterministic Rumor Mongering makes a more intelligent decision about message forwarding by making use of the fact the topology of a typical peer- to peer network displays a power-law distribution of the node degrees. By means of simulation, we show that Deterministic Rumor Mongering can significantly reduce the cost of application level broadcast compare to flooding.

The relevance of the work in this paper is not restricted to the area of file-sharing applications. The concept of Application-layer overlay networks is increasingly being proposed for a diverse range of applications [5][6][7]. An efficient broadcast method can be an important building block in these decentralized systems, especially for ad-hoc networks and other highly dynamic environments.

The rest of the paper is structured as follows. In section 2, we introduce our simulation platform and we discuss relevant parameters such as the metric of cost. Section 3 studies the cost of flooding-based broadcast, as it is implemented in Gnutella. In section 4 and 5, we evaluate two variants of Rumor Mongering as an alternative to flooding. Finally, section 6 concludes the paper

2. The Simulator

To study the cost of application level broadcast mechanism, we built a discrete event simulator in Java. This simulator takes as input a graph file, describing the topology of the network to be simulated. Its modular design allows us to easily simulate different kinds of broadcast routing protocols.

The simulation is based on synchronous rounds. In each round, every node reads the messages from its input queue and handles them according to the specified routing rules. This means either to forward the message to one or several of its neighbors or to discard it. Each node keeps track of the number of messages it has forwarded during each round. A simulation of a broadcast is initiated by having one node in the network send a message to all its neighbors and the simulation ends when no more messages are forwarded per round. The results of the simulation may depend on which node was chosen to start the broadcast. We therefore perform a large number of simulation runs, with the node initiating the broadcast chosen at random. As the result we take the average over all the runs.

2.3 Topology

To establish the cost of application layer broadcast for peer-to-peer networks by simulation, we need to generate a network topology with the typical characteristics of peer-to-peer networks.

One such characteristic that has been observed for peer-to-peer networks by measurement [9] and simulation [10], as for many other communication [11] and social networks [12], is a power-law distribution in their node degree2. That means that a few nodes have a very high degree and many have a low degree. The number of nodes with a certain degree decreases with the degree according to the following power law: *f d* ? *d k* . The frequency *fd* (number of nodes) of a degree *d*, is proportional to *d* to the power of a negative constant *k*.3

A model proposed by Barabási and Albert [13] allows generating random topologies with a power-law characteristic. This model suggests two possible causes for the emergence of a power law in the frequency of node degrees in network topologies: incremental growth and preferential connectivity. *Incremental growth* means that growing networks are formed by the continual addition of new nodes, and thus the gradual increase in the size of the network. *Preferential connectivity* refers to the tendency of a new node to connect to existing nodes that are highly connected or popular. Both of these are typical characteristics of peer-to-peer networks, which usually grow and evolve in such an ad-hoc fashion. We therefore believe that the Barabásilbert Model serves as a good basis to generate random topologies with the typical characteristics of peer-to-peer networks.

3. The Cost of Flooding-based Broadcast

In this section, we try to find the cost of flooding-based broadcast as it is implemented in Gnutella. Under the assumption that the initial TTL value is high enough for the broadcast to cover the entire network, the cost can be calculated relatively easily. During a broadcast, every node receives the message at least once. The first time the message is received, the receiving node forwards it to all its neighbors, except the one from where the message came from. Subsequent arrivals of the same message are ignored. Therefore, the number of messages that each node has to forward per broadcast is limited to the number of its neighbors, minus one. Only the node initiating the broadcast sends the message to all its neighbors, which results in one additional message to be forwarded. Hence, the cost *c* of a broadcast as defined in the previous section can be calculated simply as a function of the number of nodes *N* and the average node degree *d*, i.e the average number of neighbors per node. Flooding guarantees that all *N* nodes of the network are being reached by a broadcast, which means $r = N$.

4. Rumor Mongering

In this section, we discuss Rumor Mongering as an alternative routing protocol to implement broadcast. Using simulation, we compare its performance with broadcast based on flooding.

4.1 The Protocol

Rumor Mongering or Gossip protocols are a class of probabilistic protocols for message routing.They are also called *epidemiological* protocols, since messages are spread in a network much like a disease in a susceptible population i.e. the neighbors to which messages are forwarded by each node are chosen randomly.

Gossip protocol have been used for a wide range of applications, such as database consistency management reliable multicast failure detection garbage collection and broadcasting on small networks In most of these applications, it is assumed that every node can directly communicate with every other node or that the global topology of the network is known. This allows a specific type of graph to be superimposed on that network, on which the Gossip protocol operates. Some of the primary research in this area is concerned with finding an optimal topology graph to be superimposed in order to minimize the cost or maximize the reliability of the protocol.

In the context of decentralized peer-to-peer networks, it is not possible to superimpose an arbitrary graph on the network, since peer-to-peer networks are created in an ad-hoc fashion and new nodes can connect to any node on the network they want. Therefore, the exact global topology is not known and the individual nodes have only a very local view of the network. Little work has been done to study the performance of Gossip protocols in the context of ad-hoc peer-to- peer networks.

4.2 Simulation Results

We expect Gossip protocols to implement broadcast at a lower cost than flooding. In Gossip, each node forwards a message only to a

randomly selected subset of its neighbors, as opposed to all of them in flooding.
The reduced cost in number of messages forwarded comes at a certain cost. First of all, due to its probabilistic nature, a Gossip protocol cannot guarantee that all the nodes are reached by a broadcast. Secondly, Gossip requires more time to complete a broadcast than flooding. In flooding, a message is routed along all the possible paths in the network in parallel, including the shortest ones. Gossip cannot guarantee that the messages are routed along the optimal path, resulting in an increased time of completion of the broadcast. Table 2 shows how flooding and Rumor Mongering compare in terms of these three parameters: cost, reach and required time steps to complete the broadcast. The results are obtained by simulation, averaging over 1000 runs.
The table shows the mean values, the minimum, the maximum and the standard deviation. For flooding, all the parameters are constant, except for the required time steps, since this value depends on where in the topology a broadcast was initiated. The simulation was done for a Barabási topology of 1000 nodes with an average node degree of 6. We chose the parameters *F* and *B* for the Rumor Mongering protocol to be 2.

	Flooding				Rumor Mongering (F=2, B=2)			
	mean	min	max	std dev	mean	min	max	std dev
cost *c*	4.99	4.99	4.99	0	2.78	2.73	2.82	0.01
reach	1000	1000	1000	0	918.44	894	945	8.52
time	8.94	8	10	0.36	21.33	19	30	1.32

Table 1: Flooding vs. Rumor Mongering for a Barabási Topology (N=1000, d=6)

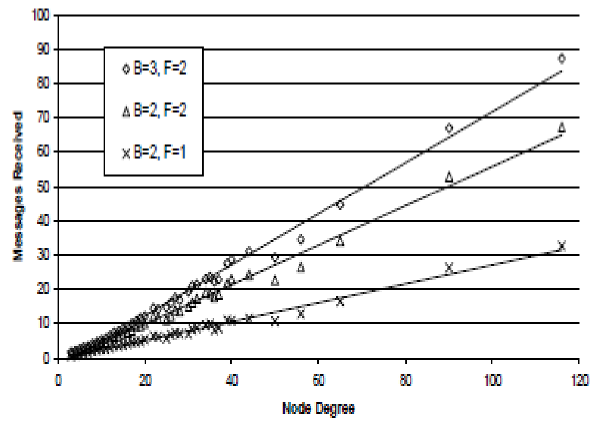

Figure 1: Average Number of redundant message received by nodes in a single broadcast implemented with Rumor Mongering, as a function of the node degree.

B	F	Probabilistic Rumor Mongering			Deterministic Rumor Mongering		
		reach	time	cost *c*	reach	time	cost *c*
2	1	67.7%	23.2	2.00	96.10%	21.1	2.00
3	1	86.9%	17.1	2.65	99.00%	16.4	2.60
2	2	91.7%	21.6	2.78	99.80%	18.4	2.82
2	3	97.1%	19.9	3.17	100.00%	18.1	3.30
3	2	97.7%	15.3	3.41	100.00%	14.2	3.42
3	3	99.2%	14.4	3.73	100.00%	14.0	3.76

Table 2: Performance of Deterministic Rumor Mongering compared to Probabilistic Rumor Mongering

For a given set of parameters B and F, Discrete Rumor Mongering achieves a significant higher reach than the probabilistic version of Rumor Mongering. It also slightly reduces the time required for a broadcast, whereas the cost is virtually the same for both versions. For example, for (*B*=2, *F*=1), Discrete Rumor Mongering reaches 96% of the network compared to 67% of its probabilistic counterpart. The question is how this relates to our main goal, the reduction of cost of broadcasting.
The increase in reach at a constant cost per node translate indirectly into a reduction in cost, considering the fact that increasing the reach of Rumor Mongering by increasing the parameters *B* and *F*, results in a in a higher cost per node reached

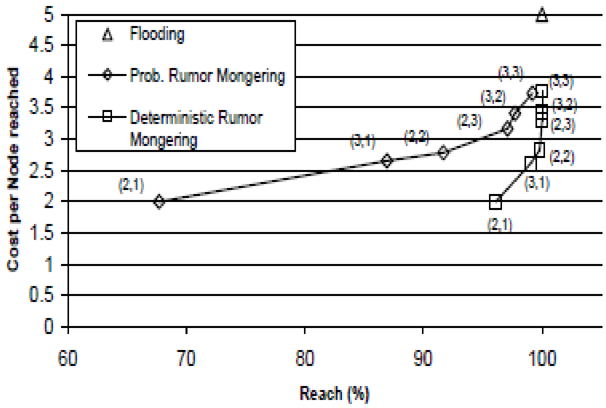

Figure 2: Cost of Broadcasting versus reach, for different sets of protocol parameters B and F

So far, all our simulations were done for networks of a constant size of 1000 nodes. To study the effect of the network size on the performance of Deterministic Rumor Mongering, we ran our simulation for the three different network sizes of 100, 1000 and 10'000 nodes. All these networks have an average node degree of 6 and the protocol parameters were chosen as follows:

B=2, *F*=1. The results shown in table 5 indicate that the network size has little or no effect on our cost parameter *c* and the proportion of the network that is reached. Only the number of time steps required to complete the broadcast increases slightly with the total number of nodes.

5. Conclusions

For fully decentralized peer-to-peer networks such as Gnutella, application layer broadcast is often an important building block that is required to implement services such as searching. In this paper, we studied the cost of Gnutella's flooding-based broadcast mechanism. We defined a metric of cost and derived a simple formula to calculate the cost, based solely on the size of the network and the average node degree. Based on this, we made an estimation of the average bandwidth that is required per node for a few typical scenarios. This showed, as expected, that flooding-based broadcast becomes prohibitively expensive, even for relatively small networks. As an alternative to flooding, we evaluated a probabilistic protocol called Rumor Mongering. Our simulation results show that Rumor Mongering is more scalable, since it can significantly reduce the cost of broadcast compared to flooding. It trades off reduction in cost against reduced reach and increased time to complete the broadcast. By choosing the parameters *B* and *F* of the Rumor Mongering protocol, the level of this trade-off can be controlled almost arbitrarily within a certain range.

We further introduced a new protocol called Deterministic Rumor Mongering. This new broadcast routing protocol achieves a much better performance than the probabilistic version of Rumor Mongering by making use of the fact that typical peer-to-peer topologies show power law characteristics in their node degree. We showed that for a typical scenario, Discrete Rumor Mongering can reduce the cost of broadcasting by around 60% compared to flooding, while still reaching more than 96% of the nodes in the network.

6. References

[1] The Napster home page, http://www.napster.com
[2] The Gnutella home page, http://gnutella.wego.com
[3] The SETI@home home page, http://setiathome.ssl.berkeley.edu
[4] Gnutella: To the Bandwidth Barrier and Beyond, http://www.clip2.com/gnutella.html
[5] David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek, Robert Morris, "Resilient Overlay Networks", Proc. 18th ACM SOSP, Banff, Canada, October 2001
[6] Y. Chu, S. Rao, and H. Zhang. "A Case For EndSystem Multicast". In Proceedings of ACM Sigmetrics, Santa Clara, CA, June 2000
[7] S. Ardon, P. Gunningberg, Y. Ismailov,B. Landfeldt, M. Portmann, A. Seneviratne, B. Thai, "Mobile Aware Server Architecture: A Distributed Proxy Architecture for Content Adaptation and Service Enhancement", INET2001, Stockholm, June 2001

[8] Gnutella Protocol Specification, http://www.clip2.com/GnutellaProtocol04.pdf
[9] Gnutella: To the Bandwidth Barrier and Beyond, http://www.clip2.com/gnutella.html
[10] T. Hong, in “Peer-to-peer: Harnessing the Benefits of a Disruptive Technology”, edited by Andy Oram (O’Reilly, Sebastopol, CA, 2001) Chap. 14, pp. 203-241
[11] Michalis Faloutsos, Petros Faloutsos, Christos Faloutsos, “On Power-Law Relationships of the Internet Topology”, ACM SIGCOMM'99, Boston, 1999.
[12] W. Aiello, F. Chung, and L. Lu. “A Random Graph Model for Massive Graphs”. In 32nd Annual Symposium in Theory of Computing, 2000.
[13] A.L. Barábasi and R. Albert, “Emergence of Scaling in Random Networks”. *Science*, pages 509- 512, October 1999.
[14] Alberto Medina, Anukool Lakhina, Ibrahim Matta, and John Byers, “BRITE: An Approach to Universal Topology Generation”. In Proceedings of the International Workshop on Modeling,
Analysis and Simulation of Computer and Telecommunications Systems- MASCOTS '01, Cincinnati, Ohio, August 2001.
[15] A. Demers, D. Greene, A. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. “Epidemic algorithms for replicated database maintenance”. In Proc. ACM Symp. on the Principles of Distr. Computing, pages 1--12, August 1987.