Efa: an Efficient Content Routing Algorithm in Large Peer-to-Peer Overlay Networks

Son Vuong Juan Li
Computer Science Department, University of British Columbia
{vuong, juanli}@cs.ubc.ca

Abstract

An important issue for peer-to-peer application is to locate content within the network. There are many existing solutions to this problem, however, each of them addresses different aspects and each has its deficiencies. In this paper, we focus on the unstructured peer-to-peer scenario and present a constrained flooding routing algorithm, Efa, which overcomes some of the deficiencies of those existing strategies. Efa performs application level broadcasting in a potentially very large peer-to-peer network overlaid on the Internet. Efa is completely decentralized and self-organized. It is a more scalable alternative to flooding, which is commonly used in unstructured peer-to-peer systems. Utilizing just a small amount of topology info, Efa is almost as simple as flooding, but it is much more efficient and scalable.

1. Introduction

One of the most challenging problems related to data-sharing peer-to-peer systems, is locating content. Content locating decides whether the system's resource can be efficiently used or not, moreover, this greatly affects the scalability of P2P systems and their other potential advantages. For any given system, the search technique depends on the needs of the application. The technique described in this paper is to improve the salability and efficiency of the content discovery under the unstructured and dynamic P2P environment through a broadcasting algorithm.

2. Discovery Mechanisms

Currently, there are two kinds of searching schemes for decentralized peer-to-peer systems [2]: *structured searching* scheme and *unstructured searching* scheme. Although structured systems such as Pastry [4], Chord [5], and CAN [3] scale well and perform efficiently, they have many limitations. Firstly, they have a high requirement on data placement and network topology, which are NOT applicable to the typical Internet environment, where users are widely distributed and come from non-cooperating organizations. Secondly, they only support search by identifiers but not richer queries. Thirdly, they just offer file level sharing. Unstructured systems like Gnutella [1] and Napster [6] do not have problems mentioned above and they are most widely used by current Internet users. Therefore, in this paper, we focus on the unstructured systems. However, in unstructured P2P systems, no clue emerges as to where content is placed, so queries have to be flooded through the network to get results. Flooding produces many duplicated messages and much network traffic, which incurs great network load. Therefore, we propose a routing algorithm aiming at suppressing flooding, that is, at reducing the number of duplicated query messages.

3. Algorithm description

In a well-connected network, several different paths may exist to connect two nodes, which is the reason for huge amounts of duplications created by flooding. If node v can anticipate that one of its neighbors u, receives query messages from another path, then v dose not forward the query to u. To achieve anticipation, we make a rule directing the nodes duplicating and forwarding messages. We keep track of 2-hops topology information to compute the forwarding set. The following definitions are used in the algorithm.

- v: Current node
- $id(v)$: Node v's unique id
- $N(v)$: Neighbor set of v
- $NN(v)$: Neighbor's neighbor set of v
- $fr(u,v)$: v is the current node, u is the node which forwards the query to v. $fr(u,v)$ is the forward reaching set of u for the current node v, that is, the immediate (no more than 2 hops away) set of nodes reached by the local flooding source u.

Proceedings of the Third International Conference on Peer-to-Peer Computing (P2P'03)
0-7695-2023-5/03 $17.00 © 2003 **IEEE**

• *routing(u,v)*: For local source *u*, current node *v*'s routing set. For example, if *u* forwards the query package to *v*, the set of nodes *v* forwards is decided by *routing(u,v)*.

The following algorithm is to compute *fr(u,v)* and *routing(u,v)*.

$fr(u,v) = N(u) \cup \{\forall v' \text{ in } NN(u), \text{ if } id(v') < id(v)\}$
$routing(u,v) = \forall v' \text{ in } N(v), \text{ if}$

1. $v' \notin fr(u,v)$ AND
2. $\{N(v') \cap fr(u,v) = \varnothing\}$ OR $\{N(v') \cap fr(u,v) = A, \forall v'' \in A, id(v'') > id(v)\}$

The algorithm below describes the routing process for the current node, *v*, when it receives a query from its neighbor, *u*.

forward(u,v)
*/*when node v receives forwarded query from its neighbor u, this algorithm decides how v forwards this query */*

If the received query has been received before
discard it
else
if u is null / v is the node which initiates the query*/*
forward the query to N(v)
else
forward the query to routing(u,v)

4. Experiments

We show preliminary simulation results of the presented broadcast algorithm, Efa. We compare the traffic cost of Efa with simple flooding over the same topology network. We set the TTL to "unlimited", to make the broadcast reach every node in the network. Both Efa and flooding are evaluated in the same way. We create two types of network topologies: the grid topology and the Barábasi-Albert [7] random topology, with different degrees. For each topology, we vary the network size. When the network becomes stable, we initiate a flooding process, starting from a randomly-chosen node. After the flooding process ends, we analyze the results to see if all the nodes are covered, and count the amount of redundant messages. Then we repeat the experiment a few times, initiating the broadcast from different sources, and compute the average value of the results.

The experimental results justify the algorithm's correctness: no matter the location of the source node, the query messages eventually reach all nodes in the network. Compared with simple flooding, Efa saves more network traffic, especially when the network is well connected. A comparison of Efa and simple flooding is shown in Figure1.

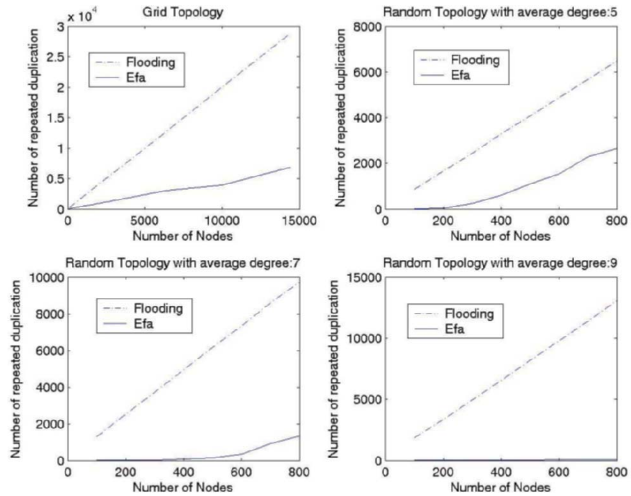

Figure 1: Comparison of Efa with flood

6. Conclusions

In this paper, we propose an application level-broadcasting algorithm, Efa, to deal with the routing problems in unstructured P2P network. Our preliminary simulation results show that the algorithm can achieve good performance and scalability.

7. References

[1] Gnutella website. http://gnutella.wego.com/.
[2] Q. Lv, P. Cao, E. Cohen, K. Li and S. Shenker "Search and Replication in Unstructured Peer-to-Peer Networks," in Sigmetrics, June 2002.
[3] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. "A Scalable Content-Addressable Network:, In ACM SIGCOMM, pages161-172, August 2001.
[4] A. Rowstron and P. Druschel. "Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems," Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001), November.
[5] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek. "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," In ACM SIGCOMM, pages 149-160,August 2001.
[6] Napster website. http://www.napster.com/.
[7] A.L. Barábasi and R. Albert, "Emergence of Scaling in Random Networks". Science, pages 509-512, October 1999.