

A Novel Learning-based Search Algorithm for Unstructured Peer to Peer Networks

Mahdi Ghorbani^{1*}, Ali Mohammad Saghiri² and Mohammad Reza Meybodi²

1- Department of Computer & IT Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

2- Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran

Corresponding author Email: Mahdigh13@yahoo.com

ABSTRACT: In order to file sharing as a popular application of unstructured peer to peer networks, finding a certain amount of data in each node, needs performing an appropriate search method. In this paper, we propose a new version of k-random walk algorithm using learning automata. In the proposed method, the value of k for k-random walk is not selected randomly but it is selected in an adaptive manner. It is decided which k walkers are more useful to be selected in order to keep on the search according to past experience of each node. Simulation results show that the novel search algorithm improves the number of hits per query, success rate and the delay of objects discovery in comparison with the k-random walk algorithm.

Keywords: Peer to peer networks, Search, Random walk, Learning automata.

INTRODUCTION

Peer to peer networks has been growing rapidly in last few years. Many applications of peer to peer systems like file sharing are due to distributed architectures of these networks. Peer to peer networks are usually implemented with no central control in order to obtain some features such as flexibility, scalability and reliability. Peer to peer networks can be categorized into two classes: structured and unstructured. In structured peer to peer networks, placement of the contents is controlled via Distributed Hash Table (DHT)s and yield few overhead on the network. The placement of the contents in unstructured peer to peer networks is more loosely and nodes can join and leave without a strict control (Androutesellis et al. 2004; Lua et al. 2004). Therefore applying a search procedure can not locate a node directly and it is necessary to use search mechanisms. Nowadays, most of applications of peer to peer systems focus on unstructured peer to peer networks and designing an efficient search method is the most important issue. Search methods in unstructured peer to peer networks can be categorized in different ways. According to query forwarding we can divide the search methods in groups of deterministic and probabilistic (Li et al. 2005). In a deterministic approach, prior information on query path is used for routing and the query forwarding is deterministic (e.g. local indices) (Thampi et al. 2010). In the other approach, the query is routed either probabilistically or randomly. Another classification of search methods is based on node's awareness of content in the network: Blind and Informed (Thampi et al. 2010; Tsoumakos et al. 2006). In informed approaches, nodes are aware of network status and the placement of the contents by storing some metadata of information. In blind search strategies, nodes keep no information about the placement of the objects. They use flooding techniques to forward the queries. Blind search methods waste the network bandwidth and yield much overhead. According to the above classifications, random walk methods (Tsoumakos et al. 2006) are instances of probabilistic and blind search methods. One form of the random walks methods is k-random walk algorithm which involves the use of k independent random walks. When an attempt to locate an object is not successful, k nodes will be selected among of neighbors randomly and the queries forward to them. The search is successful once the object is found by any one of the individual random walks. In this method, success rate and number of hits vary due to random selection of nodes for routing queries and also the peers available for a long time could be selected more often.

In some recent studies (Dorrigiv et al. 2007; Fletcher et al. 2005; Gkantsidis et al. 2004; Tsoumakos et al. 2003a,b; Thampi et al. 2008 a,b; Kalogeraki et al. 2002), reinforcement learning techniques (Sutton et al. 1998; Mance et al. 1996) such as Q-learning (Sutton et al. 1998) are utilized to improve search features. Applying learning mechanisms in a network, each node can learn about network status and makes decision for next iteration of search based on previous nodes' feedbacks. It is expected that learning-based search approaches make better search response time, decrease load of network and affect the usage of bandwidth efficiently.

In this paper, we propose a new version of k-random walk algorithm based on learning automata in unstructured peer to peer networks to choose the appropriate neighbors in order to find the resources. Our proposed algorithm, improves the success rate and number of hits per query in comparison with k-random walk algorithm. We also evaluate our algorithm via simulation.

Learning Automata

A learning automaton (Najim et al. 1994; Abolhasani et al. 2008) is an adaptive decision-making system that can improve its performance by learning how to choose the optimal action from a set of allowed actions through repeated interactions with the random environment. Automaton randomly chooses its action based on a PDF defined over the action-set. At each iteration, the selected action is served as the input to the random environment. The environment responds the taken action in turn with a reinforcement signal. The action probability vector is updated based on the reinforcement feedback from the environment. The objective of a learning automaton is to find the optimal action from the action-set so that the average penalty received from the environment is minimized. The relationship between the learning automaton and the random environment is shown in figure 1.

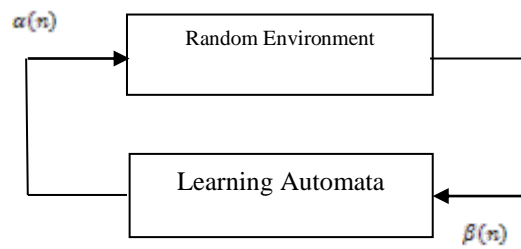


Figure 1. The relationship between the learning automaton and its random environment (Najim et al. 1994)

One form of learning automaton is KSALA (Abolhasani et al. 2008) which k actions are selected instead of one action. The response of environment is received in different methods. In “majority of polls” method, the favorable response is when the effect of $k/2 + 1$ selected actions in the environment have the same response, otherwise, the response is unfavorable.

Let a and b denote the reward and penalty parameters and determine the amount of increases and decreases of the action probabilities, respectively. Let r be the number of actions that can be taken by the learning automaton. The KSALA can be described by $\{\alpha, \beta, p, T\}$ where $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ denotes the set of actions, $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ is the set of inputs, $p \equiv \{p_1, p_2, \dots, p_r\}$ denotes the probability vector of each choice and $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ is the learning algorithm. $\alpha^h(s) \equiv \{\alpha_1, \dots, \alpha_{r_h}\}$ denotes the set of actions for choosing h^{th} action in s^{th} step where $\alpha^1(s) \equiv \alpha$. In step s , the h^{th} selected action is deleted from the set of all actions. Therefore it is not selected for next actions. Let α_i is the h^{th} selected action in n^{th} step, then the probability vector is rewarded by equation 1 and penalty is updated as given in equation 2. According to the following equations, all the members of probability vector set are updated k times.

$$\begin{cases} p_i^{h+1}(n) = p_i^h(n) + a[1 - p_i^h(n)] \\ p_j^{h+1}(n) = (1 - a)p_j^h(n) \quad \forall j \neq i \end{cases} \quad (1)$$

$$\begin{cases} p_i^{h+1}(n) = (1 - b) + p_i^h(n) \\ p_j^{h+1}(n) = (b/r - 1) + (1 - b)p_j^h(n) \quad \forall j \neq i \end{cases} \quad (2)$$

PROPOSED ALGORITHM

We focus on unstructured peer to peer networks and apply KSALA in order to train nodes during search procedure. First, we describe the data structure of the proposed algorithm and then represent the search algorithm.

Design of search algorithm

As mentioned before, probability values are so important to make decision in KSALA and are updated according to rewards/penalties feedback from the environment. In our distributed proposed algorithm, each set of k selected neighbors utilizes KSALA. It means that there is a table with k nodes and k probability values of locating objects based on past iterations. According to feedback from environment, all the probability values of the current node are updated. Here, we describe the search algorithm in details. At the first step with no past iteration, for each node k neighbors are selected randomly with the probability value of $1/k$. Every time a node forwards a query to the neighbor, a new row is added to the table. If the object is not found, locating the object will continue by selecting k neighbors with the highest probability values among all neighbors. Selected neighbors propagate query messages in order to locate the object. After receiving the feedbacks from environment, if $k/2$ of selected neighbors find the object, probability values of selected neighbors would be increased, otherwise updating the punishment of probability values is done. It is performed by KSALA algorithm. Search is terminated when the object is found or TTL is finished. To update the probability values in order to evaluate rewards/penalties in our search algorithm we apply equation 1 and 2 in the earlier section. Figure 2 and figure 3 describe updating the probability vector and proposed search algorithm, respectively:

/N: Number of all the neighbors, K: Number of selected neighbors/

IF the feedback of $[(k/2)+1]$ neighbors is "HIT" THEN
 The neighbors with "Hit" feedback updated by EQ (1) and the others updated by EQ (2)
 ELSE
 all k selected neighbors updated by EQ (2)

Figure 2. Updating P_{vectors} according to feedbacks

/ CN: Current node for search, Q: Query Keyword, K: The number of selected neighbors, P_{vector} : The set of probability values/

1. User submits a query
2. Search query node for Q
3. IF Q is not in CN
 - 3.1. Select K neighbors with the highest P_{vector} among all neighbors
 - Generate K query messages
 - Search starts with k neighbors
4. IF a hit occurs, sent back results on the reverse path
5. All nodes on the path update appropriate P_{vectors} with KSALA

Figure 3. Proposed search algorithm

Simulation

We use OverSim to simulate our search algorithm and evaluate our experiments. First we describe our network model and then set the required parameters of simulation. We construct our network using a random graph with 1000 nodes for different simulations. The average out-degree of each node is 3. There are 100 distinct objects distributed in various nodes randomly. The maximum time to deliver the messages of the queries is called TTL and is assigned 6. The node failure rate is 20% in our simulations. The maximum number of the selected walkers is 15. We performed several simulations to obtain the efficiency of our proposed algorithm. We compare our search algorithm with the k-random walks algorithm. The following graphs show our results of simulations with regard that k is the number of the walkers in the algorithms. In our simulations, walkers vary from 1 to 15.

In figure 4, it is shown that the success rate of k-random walks algorithm due to random selection of walkers with no feedback from environment, is low but in our proposed algorithm after the 5th selection of walker it is growing up to 85%. Selecting neighbors with high priority based on past iterations is the main reason for high success rate for our algorithm. The average success rate of search for our algorithm is about 65%.

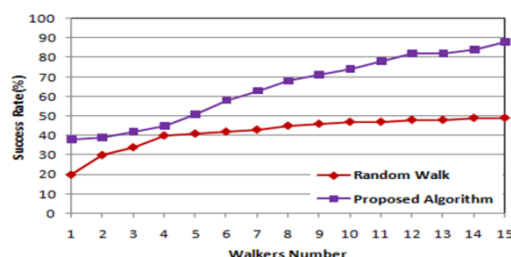


Figure 4. Comparison of success rate of two algorithms

The number of discovered objects per query is presented in figure 5. Our proposed algorithm has more precise results than k-random walk. Since in our algorithm the selected walkers always have the highest probability value, it will increase the chance of discovering objects when the query is propagated

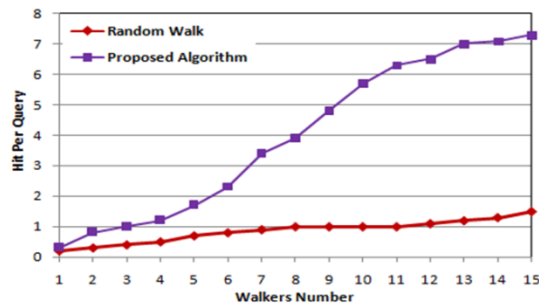


Figure 5. Hit per query under two algorithm

Figure 6 shows the number of produced messages per query under two algorithms. In k-random walks algorithm, about 70% of walkers waste TTL messages. In comparison with this algorithm our proposed algorithm produces lower messages per query. It is due to intelligent selection of walkers during search and accurate TTL consumption in order to deliver messages.

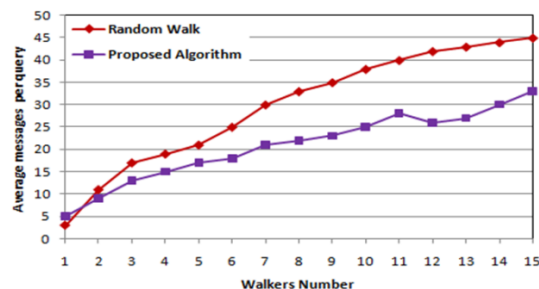


Figure 6. Overhead comparison of two algorithms

Figure 7 compares the delay of two algorithms to discover the objects. The delay is based on average number of hops visited for a search. Less hop numbers means less delay and more speed to discover objects. Our algorithm performs better than k-random walk algorithm. Although the learning phase of our algorithm and updating P_{vector} is accurate but sometimes is slow

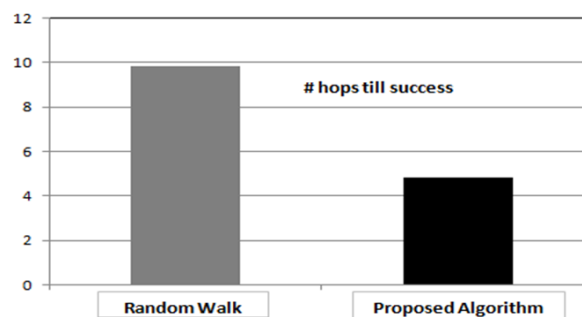


Figure 7. Delay search

CONCLUSION

In this paper we introduced an adaptive version of k-random walks algorithm for peer to peer networks utilizing a form of learning automata. Our proposed algorithm is based on adaptive select of walkers according to each node's feedback. The proposed search algorithm considers the feedbacks from the environment and then makes decision about suitable nodes in order to participate in search. By applying KSALA for each set of k nodes in the network, all the neighbors with the highest probability value of successful search in the past iterations are selected adaptively to continue the search. We compared our algorithm's performance with k-

random walks algorithm via simulations. The simulation results show that our algorithm can effectively improve the success rate, the number of hits per query, delay and produced messages per query in comparison with k-random walk.

REFERENCES

- Androutesellis S, Spinellis D. 2004. A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys* 36(4): 335-371.
- Abolhasani S M, Meybodi M R. 2008 LADIT: Learning Automata Based Protocol for Routing in Sensor Networks. Paper presented at the third Conference on Sensor Networks, Yazd, pp. 20-33.
- Dorrigiv R, L'opez-Ortiz A, Pralat P. 2007 Search algorithms for unstructured peer-to-peer networks. Paper presented at the Proc. Of 32nd IEEE Conference on Local Computer Networks, p 343.
- Fletcher G H, Sheth H A, Borner K. 2005 Unstructured peer-to-peer networks: topological properties and search performance. Paper presented at the Lecture Notes in Computer Science- Agent and Peer-to-Peer Computing, Springer Berlin/ Heidelberg, p 14.
- Gkantsidis C, Mihail M, Saberi A. 2004. Random walks in peer-to-peer networks. Paper presented at the first INFOCOM 2004, Hong Kong, , p120.
- Kalogeraki V, Gunopulos D, Zeinalipour-Yazti D. 2002 A local search mechanism for peer-to-peer networks. Paper presented at the 11th Conf. Information and knowledge management, pp. 300-307.
- Li X, Wu J. 2005. Searching techniques in peer-to-peer network. *Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, CRC Press, Boca Raton.
- Lua E K, Crowcroft J, Pias M, Sharma R, Lim S. 2004 A survey and comparison of peer-to-peer overlay network scheme. Paper presented at the IEEE Communication Survey and Tutorial.
- Mance E, Stephanie S H. 1998 Reinforcement learning: A tutorial. Paper presented at the in Proceeding of the Wright Laboratory.
- Najim K, Poznyak A S. 1994 Learning automata: theory and application. Paper presented at the in Proceeding of the Tarrytown, New York, Elsevier Science Publishing Ltd.
- Sutton R S, Barto A G. 1998. Reinforcement learning: introduction. Paper presented at the in Proceeding of the MIT Press.
- Thampi S M, Sekaran C K. 2010. Survey of search and replication schemes in unstructured p2p networks. *Network Protocols and Algorithms* 2(1):93-131.
- Thampi S M, Sekaran C K. 2008 Collaborative load-balancing scheme for improving search performance in unstructured p2p networks. Paper presented at the Proc. Of the first Int. Conf. Contemporary Computing, p 161.
- Thampi S M, Sekaran C K. 2008. An efficient distributed search technique for unstructured peer-to-peer networks. *Int. Jou. Computer and Network Security* 8(1):128-135.
- Tsoumakos D, Roussopoulos N. 2006. Analysis and comparison of p2p search methods. Paper presented at the First Int. Conf. Scalable Information Systems, Article no. 25.
- Tsoumakos D, Rossopoulos N. 2003. Probabilistic knowledge discovery and management for p2p networks. *P2P Journal*, 12(2):129-136.
- Tsoumakos D, Roussopoulos N. 2003. Adaptive probabilistic search for peer-to-peer networks. Paper presented at the Third Int. Conf. P2P Computing, p102.