

Two Levels TTL for Unstructured P2P Network using Adaptive Probabilistic Search

Yash Pal Singh, Rakesh Rathi, Jyoti Gajrani, Vinesh Jain

Abstract— P2P networks are playing an important role in current scenario of unstructured networks. P2P network supports various applications and taking the advantage over the centralized search system. Centralized search systems suffer from the problems of single point of failure, low availability, denial of service attacks. Searching of the required data is a vital issue in the P2P network. Many methods have been implemented for searching in P2P network such as Flooding, Random Walk, Expanding Ring or Iterative Deepening, K-Walker Random Walk, Two Level K Walker Random Walk, etc. These methods are based on property of randomness in the network. Some of these generate large traffic while others take long searching time. A probabilistic approach with Two Level K Walker Random Walk for searching has been implemented in this paper and comparative study has been done with other algorithms.

Index Terms— peer to peer, APS, random walk, K-walker, dynamic search, peersim, probabilistic, flooding.

1 INTRODUCTION

A P2P network is a collection of distributed, heterogeneous, autonomous and highly dynamic peers. Participant peer shares a part of its own resources such as processing power, storage capacity software and files. P2P networks are dynamic in nature. Various types of P2P networks are the Purely Decentralized Systems, Partially Centralized Systems, and Hybrid Decentralized Systems. According to network structure P2P network is classified as Unstructured, Structured and Loosely Structured based on their data location with respect to overlay topology. In the first case data content are totally unrelated to overlay topology. In Structured P2P network data contents are precisely defined with respect to overlay topology. In this type of network each node has the idea about data content so searching can be done easily. In last one searching can be done on the basis of routing hints. Structured P2P networks are not suitable for highly transient node population where node can leave and join any time. In this paper a probabilistic approach with two levels TTL has been implemented for searching in unstructured P2P network. In this case we maintain a database and each node has the same probability of its neighbors in the database. First time we perform K-Walker Random walk because every neighbor has the same probability. After termination of the search we increase the probability of the neighbor by some amount on successful path and decrease the probability on unsuccessful path. Next time when any node wants to make query, it will choose the node with highest probability from the database and send a query message to this node. This continues until the desired content is found or TTL expires. In the case of Two Level TTL, those nodes where the search is unsuccessful a new TTL1 will be initialize which will be less than the last TTL, and at every unsuccessful node the query message will be exploded in to K number of threads. This process will be continuing like K Walker Random Walk until the desired content found or TTL1 expires. We proposed a new search algorithm which takes the advantage of these two searching techniques [1]. Implementation of this algorithm and comparative study has been done with other algorithms in the paper.

2 RELATED WORK

Various search protocols have been implemented for Unstructured P2P network. Basic searching techniques are blind search and knowledge base search. Flooding and Random Walk protocols use blind search and Adaptive probabilistic search uses knowledge based search. Mostly protocols are working for file sharing applications on the Internet. Most common examples are Napster, Gnutella, Kazaa and BitTorrent. Gnutella is based on flooding which is used for file sharing application. BitTorrent is also used for Flooding [5]: In the case of flooding each querying node sends query message to its entire neighborhood. These neighbors also forward this query message to their corresponding neighbors until search is successful or TTL expires. If desired content is very far from querying node then number of message generated for this query will be very large. Iterative Deepening [5]: The idea of iterative deepening is taken from artificial intelligence and used in P2P searching, where the querying node issues BFS searches (in sequence) with increasing depth limits. It terminates the query either when maximum depth limit (D) has reached or result is obtained. Same sequence of depth limits is used by all nodes and same time period W between consecutive BFS searches. Local Indices [5]: In this technique a system wide policy specifies the depths at which the query should be processed. All nodes at depths not listed in the policy simply forward the query to the next depth. Routing indices [5]: Routing indices guide the entire search process like intelligent search. Intelligent search uses information about past queries which have answered by neighbors, While Routing indices stores information about the documents topic and also the number of documents stored in its neighbors. It concentrates on content queries, queries based on the file content rather than file name or file identifier. Dynamic Search [10]: It maintains user define threshold value. If the number of hop count is less than threshold, flooding will be performed otherwise K walker random walk will be performed. This will generate lesser amount of traffic than flooding. K-walker random walk and related schemes [5]: In the standard random walk algorithm, query message(also

called walker) is forwarded to one randomly selected neighbor which again randomly chooses one of its neighbors and forwards the query message to selected neighbor and procedure continues until the data is found. One walker is used in standard random walk algorithm and this will reduce the message overhead greatly but causes longer searching delay. In the k-walker random walk algorithm, k walkers are deployed by the querying node which means querying node (source node) forwards k copies of the original message to k neighbors (randomly selected) and at the intermediate node it will forward single copy of the message to one randomly selected neighbor until the search is successful or TTL expires. Each walker takes its own random walk. Each walker talks with the querying node periodically to decide whether it should terminate or not. Soft states are used to forward different walkers for the same query to different neighbors. This algorithm tries to reduce the routing delay. On an average the total number of nodes reached by k random walkers in H hops is the same as the number of nodes reached by one walker in kH hops and this causes routing delay to be k times smaller. A similar scheme is the two-level random walk [7]. In this scheme in level 1, the querying node deploys k1 random walkers from the querying node with the TTL1 and perform random walk at intermediate nodes. At the nodes where TTL1 expires and search is unsuccessful level2 will start, each walker forges k2 walkers with the TTL2. Query is processed by all walkers in the path. For same number of walkers ($k=k_1+k_2$), this scheme has longer searching delays than the k-walker random walk but generates less duplicate messages. Another similar approach called "modified random BFS", where query is forwarded only to a randomly selected subset of neighbors and when query message is received, each neighbor forwards the query to a randomly selected subset of its neighbors (excluding the querying node). Until the query stop condition is satisfied, same method continues. This approach may visit more nodes and has a higher query success rate than the k-walker random walk. Adaptive Probabilistic Search [5][6]: It assumes that objects and their copies in the network follows a replication distribution for storage. The number of query requests for each object follows a query distribution. This process does not affect object placement and the P2P overlay topology. APS is based on k-walker random walk and is probabilistic forwarding. The querying node deploys k walkers simultaneously and on receiving the query, each node looks up its local storage for the desired object. The walker stops successfully once the object is found otherwise it continues. Query is forwarded to the best neighbor that has the highest probability value. The probability values are computed based on the results of the past queries and are also updated based on the result of the current query. This query processing continues until all k walkers terminate either success or fail (the TTL limit is reached).

2 TWO LEVELS TTL FOR UNSTRUCTURED P2P NETWORK USING ADAPTIVE PROBABILISTIC SEARCH [1]

Performance of APS algorithm can be increase by using Two-Level Random Walk instead of One-Level Random Walk (K Walker Random Walk). In the case of Two Level Random Walk we generate K1 threads which will be less than K which we have used in K Walker Random Walk. At the edge nodes where TTL1 expires and the search is unsuccessful then second level will start and these K1 threads will exploit in K2 threads subsequently a new TTL2 will be initialized which will be less than TTL1. In this case we are generating fewer threads so chances of collision will decrease than other searching algorithms. Algorithm for the proposed technique is as follows:

2.1 Algorithm

```

K1 number of walkers in first level
K2 number of walkers in second level
Set  $K_2 < K_1$ 
TTL1 Life or number of hops travel by each walker in level1
TTL2 Life or number of hops travel by each walker in level2
Set  $TTL_2 < TTL_1$ 
level - variable for level number
ttl variable to hold TTL1 and TTL2 in each Level
K3 variable to hold K1 walkers and K2 walker in each level
tcount - counter for TTL1 and TTL2 value
Select a querying node
tcount=1
level = 1
While (level <= 2)
{
While (tcount <= ttl)
{
Select a neighbouring node by applying APS;
Process the node;
If Object Not Found then
tcount++;
Continue;
Else
Come out of the loop (exit)
}
level ++
}

```

In the above algorithm a variable level will define the phases for querying message in case TTL1 and TTL2 level is initializing to 1 in first phase.

3 SIMULATION AND EXPERIMENT

Peersim 1.0.5 simulator is used to implement the algorithm which supports 1000000 nodes. To perform simulation random graph model is used. Component of this graph is connected by Wire K out class of Peersim. We have taken number of simulation cycles as 200 and measuring the time in terms of cycles. We have taken value of TTL1=10 means message will travel at most 10 hops while in Two Level K

walker Random Walk and APS with Two Level K walker Random Walk TTL1=7 and TTL2=3 . An average in degree of the node is 10. For giving the input configure parameters set in configuration file. A database has been maintained to store probabilities value for neighbours. Initially all the probability values are the same in the database and updating according to the searching result. We are incrementing the probability value in database if hit is encountered and decrementing the value if search is unsuccessful.

3.1 Discussion

Each node has probability value in its database of neighbours for the requested object .If there are Q such object and N neighbours, the space required will be $O(Q \times N)$. For a typical network node, this amount of space is not a trouble. The nodes having limited amount of storage capacities, index values for objects not requested for long time can be erased. This can be achieved by assigning a time-to-expire value on each newly-created or updated index. Each query message or update message carries path information, storing a maximum of TTL peer addresses. Alternatively, each node can associate the search and requester node IDs with the preceding peer in the path of the walker. Updates then follow the reverse path back to the requester. This information expires after a certain amount of time.

The total number of messages exchanged by APS method to expire in the worst case will be $(2 \times k \times TTL)$ where all the walkers (k walkers) travel TTL hops and then invoke the update method. So the method has the same complexity with its random counterpart. The update messages along the reverse path are the only extra messages that arise in APS. Here the two index update policies are used.

Beside the paths of all k walkers, indices are updated so that better next hop choices are made with higher probability. Learning features are also included in both positive and negative feedback from the walkers in both update approaches. In the optimistic approach, each node on the walker's path increases index values by more than the subtracted amount (positive feedback). If the walker is unsuccessful, each node on the walker's path decreases the relative probability of its next hop for the requested object concurrently with the search. So, if the probability of a node initially for a certain object was P, it becomes higher than P if the object was discovered through (or at) that node and lesser than P if the walker failed. Conversely, if many of our walkers hit their targets on average, the optimistic approach should be considered. This is the only invariant we require from our update process. This algorithm takes the benefits of Two Level K walker Random walk in following manner. This algorithm reduces redundancy by decreasing the average number of times a node is searched. In the one-level k-walk algorithm k random threads are generated from the source and they are likely to suffer "thread collisions" especially near the source and this result in having redundant hits in the same nodes (nodes being searched multiple times). Whereas Two-level algorithm sends smaller number of threads from the source node and it will give a smaller probability of thread collisions near the source node. Each of these K1 threads will then

explode into K2 threads once it is sufficiently away from the source and from the other threads. So the same number of search threads can be generated ($k=K1 \times K2$) and with a larger number of nodes searched and a minimizing probability of redundant searches to the same nodes using the same number of total search steps.

4 CONCLUSION

Various searching techniques in unstructured p2p networks are studied and their comparative study is done. A new search technique Two Levels TTL for Unstructured P2P Network using Adaptive Probabilistic Search which helps in further enhancing the performance of APS is implemented. Further work involves reducing the size of the query message which will reduce the bandwidth. APS is the efficient technique among all which is further improved by Two Levels TTL for Unstructured P2P Network using Adaptive Probabilistic Search technique. This searching algorithm uses two-level k-walker random walk with APS instead of k-walker random walk. Advantages of two level walk will further reduce collision of nodes and also helps in searching the distant nodes in the network but it may increase the response time slightly.

REFERENCES

- [1] Dr YashPal Singh, Rakesh Rathi, Jyoti and Vinesh Jain, 'Analysis of Searching Techniques and Design of Improved Search Algorithm for Unstructured Peer-to-Peer Networks', (IJCN) International Journal of Computer and Network Security, Vol.2, No.4, April 2011.
- [2] Stephanos Androutsellis-Theotokis, 'A Survey of Peer-To-Peer File Sharing Technologies', White Paper, ELTRUN, Athens University of Economics and Business, Greece, 2002.
- [3] V. Vishnumurthy and P. Francis. On heterogeneous overlay construction and random node selection in unstructured P2P networks. In Proc. IEEE Infocom, 2006.
- [4] MA. Jovanovic, Modelling large-scale peer-to-peer networks and a case study of Gnutella. aster's thesis, Department of Electrical and Computer Engineering and Computer Science, University of Cincinnati, June 2000.
- [5] Xiuqi Li and Jie Wu, Searching Techniques in Peer-to-Peer Networks, Department of Computer Science and Engineering, Florida Atlantic University.
- [6] V. Vishnumurthy and P. Francis. A comparison of structured and unstructured P2P approaches to heterogeneous random peer selection. In Proc. Usenix Annual Technical Conference, 2007.
- [7] D. Tsoumakos and N. Roussopoulos. Adaptive Probabilistic Search (APS) for Peer-to-Peer Networks. Technical Report CS-TR-4451, Un. of Maryland, 2003.
- [8] Inad Jawhar and Jie Wu, A Two-Level Random Walk Search Protocol for Peer-to-Peer Networks, Department of Computer Science and Engineering, Florida Atlantic University.
- [9] Beverly Yang Hector Garcia-Molina, Improving Search in Peer-to-Peer Networks, Computer Science Department, Stanford University.
- [10] Tsungnan Lin, Pochiang Lin, Hsinping Wang Dynamic Search Algorithm in Unstructured Peer-to-Peer Networks IEEE Transactions On Parallel And Distributed Systems, Vol. 20, No. 5, May 2009.
- [11] Gian Paolo Jesi. PEERSIM/HOWIO: build a new protocol for the peersim simulation framework
- [12] <http://peersim.sourceforge.net/tutorial1/tutorial1.html>, November 2004

TABLE 1
COMPARISON ON SIMULATION BASIS

Parameter /Protocol	Network Size	TTL1/TTL2	Walkers	In Degree	Seen	Messages	Hits	Threshold
Flooding	1000	10	-	10	999	9807	14	-
Random Walk	1000	10	1	10	10	10	1	-
K walker								-
Random walk	1000	10	5	10	104	105	3	
Dynamic								
Protocol	1000	10	3	10	820	8789	22	5
APS	1000	10	3	10	80	100	10	-
Two Level K								
Walker Random	1000	7	3	10	90	122	12	-
Walk		3						
APS Two Level K								
Walker Random	1000	7	3	10	85	130	17	-
Walk		3						

