International Journal of Computer and Communication Technology

Volume 7 | Issue 1 Article 4

January 2016

ENHANCEMENT IN REVISITING DYNAMIC QUERY PROTOCOL IN UNSTRUCTURED PEER-TO-PEER NETWORK

DIPTI GOGAWALE
M.E. Computer, Pune Institute Of Computer Technology, Pune, India, sdgogawale@gmail.com

MADHURI WAKODE
M.E. Computer, Pune Institute Of Computer Technology, Pune, India, madhurichopade@gmail.com

Recommended Citation

GOGAWALE, DIPTI and WAKODE, MADHURI (2016) "ENHANCEMENT IN REVISITING DYNAMIC QUERY PROTOCOL IN UNSTRUCTURED PEER-TO-PEER NETWORK," *International Journal of Computer and Communication Technology*: Vol. 7 : Iss. 1 , Article 4.
DOI: 10.47893/IJCCT.2016.1327
Available at: https://www.interscience.in/ijcct/vol7/iss1/4

ENHANCEMENT IN REVISITING DYNAMIC QUERY PROTOCOL IN UNSTRUCTURED PEER-TO-PEER NETWORK

DIPTI GOGAWALE[1] & MADHURI WAKODE[2]

[1,2]M.E. Computer, Pune Institute Of Computer Technology, Pune, India
E-mail: sdgogawale@ gmail.com, madhurichopade@gmail.com

Abstract- The Peer to peer architecture run over internet are widely used in unstructured topology. Random walk based and control flooding resource query algorithms are widely used in peer to peer network. In this paper we consider searching of file in ad hoc network. We propose Enhanced Selective Dynamic Query algorithm. The main aim of this algorithm is to minimize traffic cost and response latency for resource query response. When user want to search a query first history table is referred. If query is present in history table next searching is not required otherwise optimal combination of integer TTL value and set of neighbor for next query round is calculated in network by using knapsack for next query. This algorithm tries to achieve best tradeoff between traffic cost and response latency.

Keywords- *Peer to Peer network, Query algorithm, Selective Dynamic Query, wireless ad hoc network.*

I. INTRODUCTION

Peer to peer is distributed application architecture in which each computer in the network can act as a client or a server for the other computer network. Based on how the nodes in the overlay network are linked to each other P2P can classify as structured or unstructured. In unstructured network peers are connected in ad hoc fashion. It is h resilient to peer failure and simple to be implemented.

In this paper we consider the problem of resource query in unstructured peer to peer by using wireless ad hoc network. Wired network is associated with network failure like links going up and down, node crashing and have many vulnerabilities facing malicious intruders. In wireless ad hoc network no pre deployed infrastructure is available. It is self organizing and self healing. Nodes communicate with each other without intervention of access point.

The problem definition is like in a network represented by G(V,E) we need to find file f with minimum traffic cost and response latency. Traffic cost is number of messages required to complete a query and response latency is time required to find out file.

Enhanced selective dynamic query Protocol is proposed in this paper. The algorithm returns the file from history table if it is previously searched otherwise it calculates group of neighbors and integer TTL having optimal combination for next query.

The rest of paper is organized as follows the related work has given in section 2.In section 3 we define problem and explain proposed model. We introduce the methodology and solution description in section 4 and at finally section 5 concludes this work.

II. RELATED WORK

Minimizing resource query time in unstructured peer to peer in an active area of research.To address this issue two widely used approaches are Controlled flooding and Random walk based.

In Controlled or Iterative flooding based algorithm each packet of an individual query round carried out an integer TTL value [2][4].

One such control flooding algorithm is Dynamic Query protocol proposed by Gnutella developer. It retrieves result at low traffic cost[9].

In random walk based algorithm as name suggest the query node is send by query packets in random fashion until the target is achieved [8] .

Biased Random walk [3]uses statistical preferences .This algorithms can reduce network traffic and enhance the system scalability but disadvantage is that it usually result in much longer search latency [7], [8]. Puttaswamy et al proposed to use index replication [8] to find “rare” objects. By using this every node stores just the metadata of its data on all of its multihop neighbors. Some researchers improve search efficiency by exploiting the geographical and temporal locality [12]. Chawathe et al. [13] direct queries to high- capacity nodes, thus increase the chance of finding the request item. Chen Tian[1] proposed SDQ. It minimizes traffic cost and response latency by using Knapsack. Enhancement of this algorithm is possible by adding Cashing.

III. PROBLEM DEFINITION

Peer to Peer system S consist of S = S1, S2, S3,...,Sn where S is set of Peer in system.
S=(Rl, Hes ,Dl, Hne, nTTL,Pes, Dne / Ɵ) (1)

where
Rl ->Result Need
Hes ->Horizon Estimated
,Dl ->Degree Estimated
Hne ->Estimation of Next Horizon
nTTL, Dne ->Next QueryTTL
selectQuery(Dne ,nTTL){Select proper set of neighbor}
Pes ->Estimation Popularity

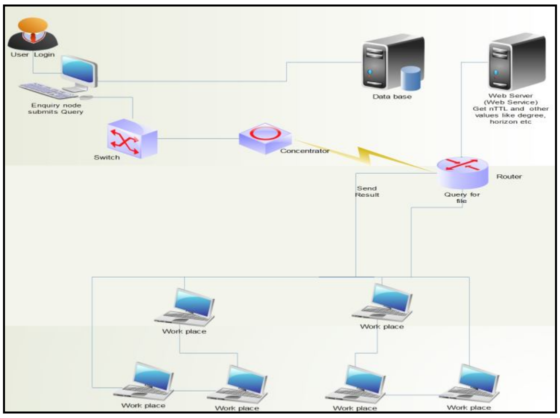


Fig 1.ESDQ in Unstructured P2P Networks

We are going to Implement Following algorithms in Our Project.

1)Iterative phase algorithm:
Depending on Hne of next query and the total residual degrees of all unused neighbors, SDQ calculates a proper nTTL value and the number of required degree; then it select subset of the neighbors according to the number of required degrees; query packets with this nTTLvalue are then propagated via neighbors. If proper results are returned then the iteration process stops other-wise, a new query round is initiated.

2) Popularity and horizon estimation:
SDQ is less likely to have a big overshooting with a relatively smaller perspective TTL value.
H_{es} is estimation horizon of touched nodes
P_{es} is estimation popularity
$P_{es} = R_{es}/H_{es}$ (2)

3) Selecting Next TTL
The multi objective optimization problem is hard to find an optimal solution. We can exploit the integrity of nTTL values. D_l is the key here. For each possible nTTL value, we can use (4) to calculate the number of neighbors' degrees d which should be covered.

4) Calculating Next Query Set:
Select optimal subset of residual neighbors to reach D_{ne} can be solved as follows. Let n be residual neighbor, i is neighbor index [1]
$A=\{a_i ,1\leq i \leq n\}$ (3)
X_i =1 if neighbor i is chosen for next query otherwise it is 0.

By using Knapsack try to achieve target as follows
$\min(\sum^{n}_{i=1} a_i x_i)$ (4)
Knapsack have high computational complexity .We introduce Search History pattern Information, based on this information.

5) History Table:
We are enhancing SDQ using history table. It is a data structure which maintain list of files which are recently requested with their location .When user request a file or made some query, algorithm first consults with history table. If the file name is in history table user request is directed to corresponding destination address field otherwise it calculates optimal combination of nTTL value and set of neighbors. This algorithm minimizes traffic cost and Response Latency by using History table.

The database present in a such format that we can use Least Recently used page replacement algorithm.

The least recently used File(LRU) replacement algorithm, though similar in name to NRU (Not Recently used), differs in the fact that LRU keeps track of File search over a short period of time, while NRU looks at the usage in the last clock. LRU works on the idea that files that have been most heavily searched in the past few queries are most likely to be used heavily in the next few queries too. LRU can provide near-optimal performance [14].

IV. METHODOLOGY

If query which user wants to search is already present in History table then it directly picked up otherwise it calculate optimal combination of an integer TTL value and set of neighbor for next query round[1]. As for each request we determine nTTL value for next query around and we calculate degree of next node .TTL value if the Time to Live for a particular request that is to say If the request doesn't get reply within a given time then that request is discarded by a particular peer ..Hence our System is NP complete because if node doesn't get response within given time same request can be regenerated.

There are a few implementation methods to implement history table it tries to reduce the cost yet keep as much of the performance as possible.

The most expensive method is the linked list method, which contains all the File Names in a list in memory. At the back of this list is the least recently used File, and at the front is the most recently used File. The cost of this implementation depends on items in the list will have to be moved about every memory reference, which is time-consuming process.

Another method that requires hardware support is as follows: suppose the hardware has a 64-bit counter that is incremented at every Search query. Whenever a File is searched, it gains a value equal to the counter at the time of File access. Whenever a File needs to be replaced from database, the algorithm selects the file with the lowest counter and swaps it out.

Thus In our Practice we have to consider above approach to keep counter with each File, Initially the counter is set to 0 , whenever a file is searched in Given time t ,its counter needs to be implemented .Thus after specific time interval t which is initially set ,The file with lowest counter needs to be deleted from the given table .

To implement this approach consider following History Table

TABLE 1: HISTORY TABLE

Consider In above table we have maintained counters for each file for a particular time t, we consider time

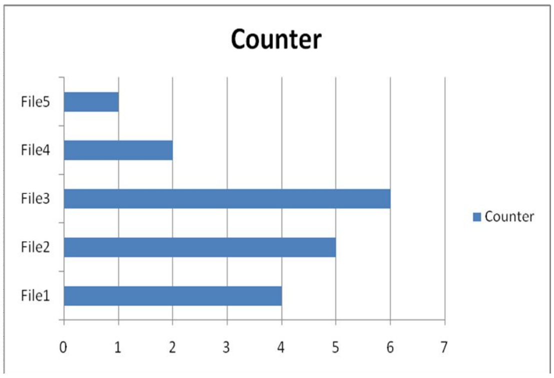
Sr No	FileName	File Info	Destination Node	Counter
1	Demo.txt	Corel Draw Demo	4	4
2	Demo.txt	Visual Studio installation demo	3	5
3	Demo.txt	SQL Server installation demo	2	6
4	Manual.pdf	Java User Manual	4	2
5	Image1.jpg	College image	1	1

in days, Let t be 15 days or more . For Given time period t, this counters indicate that file is searched c times ,where c is the counter for the file .

File 1: 4 Counter for given time t
File2: 5 Counter for given time t
File3: 6 counter for given time t
File4: 2 counter for given time t
File5:1 counter for given time t

Now, As per Least recently used (LRU) algorithm ,we observe that File5 has lowest counter than any other file in given time t, Hence we choose to remove this file from Database . This algorithm needs to be implemented only when we are about to store files and counters on timely basis.

We can show above analysis in following Gantt chart



This paper used to promote better performance of existing SDQ.

V. CONCLUSION

In this paper we are enhancing SDQ using History Table. If query which user want to search is already present in History table then user request is directed to corresponding destination address field otherwise it calculates optimal combination of an integer TTL value and set of neighbor for next query round. This algorithm tries to improve performance of Dynamic Query Protocol in Unstructured Peer to Peer Network by minimizing traffic cost and response latency and give robust performance.

REFERENCES

[1] Chen Tian,Hongbo Jiang,Xue Liu,Wenyu Liu“Revisiting Dynamic Query Protocol in Unstructured Peer-to-Peer Networks”,Jan 2012.

[2] N.Chang and M. Liu, “Revisiting the TTL-Based ControlledFlooding Search: Optimality and Randomization,” Proc. ACM MobiCom, 2004.

[3] R.Beraldi,“Biased Random Walks in Uniform Wireless Networks,”IEEE Trans. Mobile Computing, vol. 8, no. 4, pp. 500- 513,Apr. 2009.

[4] N. Chang and M. Liu, “Optimal Controlled Flooding Search in aLarge Wireless Network,” Proc. IEEE Third Int’l Symp. Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2005.

[5] E. Cohen and S. Shenker, “Replication Strategies in UnstructuredPeer-to-Peer Networks,” Proc. ACM SIGCOMM, 2002.

[6] O.S. Community, http://gnutella.wego.com/, 2010.[10] A. Crespo and H. Garcia-Molina, “Routing Indices for Peer-to-Peer Systems,” Proc. IEEE 22nd Int’l Conf. Distributed Computing Systems (ICDCS), 2002.

[7] C. Gkantsidis, M. Mihail, and A. Saberi, “Hybrid Search Schemes or Unstructured Peer-to-Peer Networks,” Proc. IEEE INFOCOM,2005.

[8] Q.Lv,P.Cao,E.Cohen,K.Li, and S.Shenker ,”Search and Replication in Unstructured Peer to Peer Network” ,Int’l conf,2002.

[9] D.Stutzbach and R.Rejaie ,”Characterizing the Two Tier Gnutella ”,Proc .ACM ,Conf 2005

[10] H.Jiang and S.Jin,“Exploiting Dynamic Querying Like FloodingTechniques in Unstructured Peer-to-Peer Networks,” Proc. IEEE 13th Int’l Conf. Network Protocols (ICNP), 2005.

[11] I.Stoica,R.Morris,D.Karger “Chord: A Scalable Peer –to-Peer Lookup Service for Internet Applications” Proc ACM, 2001.

[12] H.Cai and J. Wang, “Exploiting Geographical and Temporal Locality to Boost Search Efficiency in Peer to Peer System”,IEEE Trans,Oct 2006.

[13] Y.Chawathe ,S.Ratnasamy “Making Gnutella like P2P system scalable”,Proc ACM 2003

[14] Kedzierski,Moreto,Cazorla,Valero“Adapting cache partitioning algorithms to pseudo-LRU replacement policies” ,IEEE 2010

◈◈◈