

Knowledge and Cache based Adaptive Query Searching in Unstructured P2P Networks

G. Ramachandran

Department of Computer Science Engineering,
Annamalai University, Chidambaram, Tamilnadu,
India,

K. Selvakumar, PhD.

Department of Computer Science Engineering,
Annamalai University, Chidambaram, Tamilnadu, India,

ABSTRACT

Efficient search is a challenging task in unstructured peer-to-peer networks. In this paper, Knowledge and Cache based Adaptive Query Searching (KCAQS) is proposed that adaptively performs a query searching through either directed flooding or biased random walk based on the number of hop counts in query message. In addition, knowledge intended forwarding is deployed for forwarding a query to the high quality peers through probabilistic knowledge predicted from the previously requested queries. Searched results are properly cached in the peers along the returning path. Synchronized caching is performed to properly update the responses of each peer to its connected corresponding high degree connectivity peer in the overlay network. Due to caching of the same files in many peers, most of the cached responses may become redundant. In order to avoid redundant data, cache consistency is sustained through the flexible polling mechanism where a proper cache update is performed through Additive Decrease Multiplicative Increase (ADMI) algorithm based on file utility. Our experimental study shows that the proposed searching scheme significantly reduces the network search traffic and communication overhead. Performance metrics such as success rate, access latency, network traffic response time and cache hit ratio are evaluated for the proposed scheme.

General Terms

Peer to Peer Networks, Search Algorithm, Network Traffic

Keywords

Unstructured Peer to Peer Networks, Blind Search, Caching Technique, Efficient Search

1. INTRODUCTION

Peer is a self-governing entity that possesses large storage capacity and high data processing capability. In a computer network, Peer to Peer (P2P) is a set of autonomous self-organizing peers that may act as a client or server. The main intention for the P2P network is that sharing of distributed resources (files, databases) among peers to eliminate the single point of failure problem due to centralized server [1]. P2P computing is emerged as a prospective powerful model to share a large amount of data with high precise resource scaling through the direct exchange of files [2] between peers in the network rather than centralized server. In a unstructured P2P network such as Gnutella [3], Free net where peers are self-organized randomly and dynamically adapts to continuous node membership changes without incurring high maintenance overhead [4]. The major issue in the unstructured P2P network is hard to locate resources or services among randomly placed peers and data items in the network. There is a need for efficient searching mechanism that incurring less access delay and low overhead to

successfully locate resources among various peers in the network. The existing P2P resource sharing systems either relies on centralized server or various flooding algorithms but does not perform efficient searching because of the ineffective techniques to search and retrieve content from a location of peers in the network [5]. In unstructured P2P networks, peers are termed as blind because they do not have capability to search and determine neighbor peers which satisfies resource queries [6].

Search algorithms are categorized into two types as uninformed (Blind) and informed (Heuristic) based on usage of location resources. In informed search algorithms, centralized server or distributed directory service which maintains localized knowledge (directional metadata) is deployed for resource discovery to take better heuristic decision. Whereas in uninformed search algorithms, query searching is a complicated task due to non availabilities of location information. Two distinct uninformed search algorithms exist named as flooding or Breadth-First-Search (BFS) and k-random walk or Depth First Search (DFS) [7]. In Breadth First Search (BFS) scheme, the location of desired resources is determined through flooding the query among the peers in the network where query searching is terminated until TTL expired or desired resources are found out. Here, searching is simple and high success rate is achieved but a large amount of processing overhead is incurred because the query is spread throughout the network. In Random Walk (RW) [8] search algorithm query message is forwarded to one of its neighbors within the single hop. In this type of searching, search cost is low but it takes long search time and less success rate because coverage range grows linearly with hop count and also walker's visits only one peer for each hop [8].

1.2 Contributions

In the proposed work, adaptive searching is performed based on the knowledge learned from the user past experience. It is strengthened using distributive caching that caches the query responses along the peers in returning path. Therefore, an effective search is obtained through the improvement in certain quality metrics termed as response time, success rate, network traffic and access latency.

i) In KCAQS, a query search is performed through either directed flooding or biased RW based on the TTL value in Query message. Hence, Queries are forwarded probabilistically to high quality peers through the knowledge intended forwarding.

ii) If TTL in query message is less than threshold value, a directed flooding forwards query to a subset of n number of neighbor peers within transmission probability p. It enhances

searching efficiency only in less TTL but causes high overhead for larger TTL.

iii) If TTL in query message is greater than threshold value, a biased Random Walk (RW) is performed that forwards query to intelligently selected neighbor peer in longer hops that ultimately reduces large overhead but would take longer response time. Therefore, it is suitable for large TTL. Hence, query searching is performed adaptively varied upon the number of hop count mentioned in generated query.

iv) Synchronized caching is performed to cache the responses of previously requested queries among the peers in returning path. Here, cached responses in low degree connectivity peers are updated into next directed high connectivity peers in an overlay network. Therefore, the query search is performed only in the overlay network which encompasses a large number of high connectivity peers.

v) Cache consistency is obtained through Flexible Polling Mechanism (FPM) where proper cache update is performed through Additive Decrease Multiplicative Increase (ADMI) algorithm based on File Utility. The responses of mostly requested queries are updated within a short interval of time (Short TTR) where as rarely requested queries are updated within a long interval (Long TTR). Hence, peer has large probability to access the response in short TTL rather than responses in long TTR.

vi) Search Efficiency of the proposed scheme is evaluated through the high success rate, less response time and reduced network traffic. Furthermore, a comparison is made between the search efficiency of KCAQS scheme with or without replica.

1.3 Paper organization

This paper is structured as follows: Part 2 analyze the related works of peer-to-peer search schemes in unstructured networks. Part 3 proposes an overview of the proposed system. Part 4 explains the detailed description of the proposed scheme that comprises knowledge based adaptive searching with synchronized caching and the flexible polling cache consistency mechanism. Part 5 provides the experimental results and performance evaluation. Part 6 offers the conclusion of this work.

2. RELATED WORK

A large number of research activities have been carried out for the resource discovery mechanism in the unstructured P2P network. An extensive study is focused on existing search algorithms such as UIM, Ant search algorithm, CoQUOS where queries are forwarded using flooding mechanism and mainly utilizes factors such as a user's common interest, localities of content available among peers to improve the search quality. But, it fails to achieve optimized query searching and low access latency.

In popularity-biased random walk [9], Queries are forwarded only to peers having the highest probability biased content popularity (P_i) until TTL has expired. Popularity of a peer is computed using metropolis algorithm with a markov chain. It is a newly reconstructed topology makes use of heterogeneities available in peer popularities and index free searching scheme. In this scheme, efficient object replication and topology reconstruction are realized using square root principle to achieve efficient search performance with less search time and improves high query answering capabilities of peers in the network. A User Interest Model (UIM) [10] is fashioned as a probability theoretic framework in which the

user's common interest is designed through statistical patterns among peers that locally shared resources. Greedy file search protocol is deployed to attain faster resource discovery in which queries are forwarded to peer having highest probability. Therefore, a file with high matching degree is returned to requested users. A popular probabilistic modeling tool termed as Condition Random (CRF) is used to estimate interest difference between peers in the network. The routing tables are properly updated according to queries are perceived by each peer and reduces overall communication and processing cost. It does not relies on any predetermined semantic structure because varies dynamically according to files retrieved by peers in the network.

Ant Search algorithm [11] is a controlled flooding algorithm where the search process comprises two phases (1) Probe phase (2) Flooding phase. In probe phase, queries are flooded among peers within small TTL to acquire the statistical information about requested file. After that in flooding phase, two parameters such as k (the number of neighbors can flood the query) or TTL (an upper bound of flooding hops) must be chosen according to the information collected in probe phase. In this search, search efficiency is improved since queries are flooded only to neighbor peers having higher pheromone values which are data structure collects the information about the location of targeted file. Hence, queries are flooded only to peers having capability to provide the response for requested query. CoQUOS [12] is an efficient P2P content distribution searching scheme in which query search is performed through a continuous query model by means of public subscriber scheme. It possesses two potentials such as Cluster-resilient Random Walk (CRW) mechanism to circulate continuous queries across peer and Dynamic Probability (DP) consistently decide whether peer can register the received queries or not. During continuous query searching, the searched data item is found out through announcements made out from a new advertising peer. The success rate is effectively improved in the sense of designing effective high quantify notification.

Light flooding [13] is same as that of pure flooding, whereas a large number of redundant messages are minimized to improve the scope of searching. Two primary allusions in pure flooding insisted such that high redundant messages are possible only in high hops and high network coverage exists in lower hops rather than higher hops. Light flooding is composed of two stages. In the first stage, queries with small TTL ($M=2$) are transmitted into overlay networks until M reaches zero such that peers which receives queries are termed as seeds. At that time, a flood net is constructed using seeds as tree sub overlay as low diameter clusters based on the degree of connectivity in order to eliminate unwanted redundant messages in higher hops and then queries with TTL (N) are transmitted wholly throughout the network. Therefore, the light flooding is an efficient flooding scheme to improve the scope of searching by reducing a large number of redundant messages.

3. OVERVIEW OF THE KCAQS

This section offers an overview of the proposed scheme which adopts various meticulous strategies such as adaptive query searching based on hop count. In an unstructured P2P network, a large number of peers are randomly placed throughout the network where there is no awareness about the location of other peers in the network. Therefore, Query searching is a complicated task because queries must be forwarded to every single peer in the network. It generates high overhead and longer searching time. In our proposed

methodology, certain assured enhancements are carried out in the existing searching schemes to achieve high search efficiency with less overhead.

- On underlying Gnutella, the overlay network is constructed with peers of high degree connectivity using power law distribution.
- Neighbor's peers of low connectivity are assembled in a certain group and soon after that they will update its resources and location information to any of the neighbor peer with high connectivity in the overlay network.
- Instead of forwarding query to every peer in the network, queries are only routed to high connectivity peers where low connectivity peers update their location information. It reduces searching cost. Most probably, high connectivity peers have high bandwidth, therefore, it can handle large query traffic.
- In order to achieve an efficient message forwarding mechanism in an unstructured P2P environment, the proposed knowledge based adaptive query search that forwards query using knowledge intended searching based on TTL (the number of hop counts) in two stages such as flooding or Breadth First Searching (BFS) and Random Walk or Depth First Searching (DFS) where an appropriate decision is taken based on the affiliation between Hop Count (HC) of query message and decision threshold (T).
- In the proposed scheme, tradeoff is readily made where caching of frequently requested data items causes an additional computation and storage cost. But, it reduces the more expensive bandwidth utilization and recovers the network from high traffic.
- In synchronized caching, cached responses in low connectivity peers are properly updated in to the cache of its connected high degree connectivity peer. Cached information is properly updated for every time interval TTL based on utility of file.
- Knowledge intended searching with synchronized caching accomplished fast optimized searching improves searching efficiency with high success rate, less response time and reduced network traffic.

4. PROPOSED KCAQS SCHEME

Our proposed scheme is efficiently designed to make more suitable for power law deployed network which is an unstructured P2P system composed of peers $\{P_0; P_1 \dots P_n\}$ and those peers are connected to each other through logical links $\{L_0, L_1 \dots L_n\}$ in a decentralized manner. Two neighbor peers P_i and P_j are connected through a link $L = (P_i; P_j)$ [14]. Here, topology is prominently maintained through ping and pong message even under the dynamic admission and removal of peers from a network. Each peer in a network advertises its resources with other peers as announcement messages composed of metadata (a list of keywords) which describes the content of its resources. In addition to that, peer maintains a cache to store the responses of previously requested queries.

4.1 Construction of Overlay Network using PLD

Gnutella network has Power Law Distribution (PLD) [15] that allows holding less number of peers with a high degree of connectivity whereas many peers are available with a low degree of connectivity. It shows that on underlying Gnutella, a flat overlay network is constructed with peers of high degree connectivity. As soon as, low connectivity peers with in certain radius are assembled into a group and that they will update its resource and location information to the neighbor peers with high connectivity in the overlay network. It paved the way for effective query searching where queries are merely flood only in established overlay networks rather than throughout the peers in the network. The generating function of PLD with 'n' number of peers of 'd' degree vertex connectivity is given by

$$G[n] = \sum P_d (n^d) \quad 0 \leq d \leq \infty$$

Where P_d is the probability for the vertex on the graph has a degree d and d is the maximum degree of the peer with high connectivity. In case of any failure or breakdown in high connectivity peer as soon as responsibilities of high connectivity peer is handed over to the next direct high connectivity peer in that group. On that account, Generating function of n number of peers is resolute as follows

$$G'[n] = \sum P_d (G[n]) \quad 0 \leq d \leq \infty$$

4.2 Knowledge based Adaptive Query Searching

In order to achieve better scalability, KAQS [16] is intentionally performed only on overlay network. Furthermore, our proposed approach utilizes the feedback learned from the past experience to probabilistically guide future query processing. Every peer has to construct the probabilistic function using knowledge learned from past experience regarding to query hit, searching time and local topological information. Henceforth, In order to forward queries, an appropriate decision is taken intelligently from the constructed probabilistic function[17]. In knowledge intended searching, queries are successfully forwarded wholly throughout peers in the network but only for a small number of peers in each hop. It improves search efficiency with high success rate, less response time and reduced network traffic. The query search is performed adaptively in two phases called as threshold phase (Directed Flooding) and ultimate phase (Biased Random Walk) depending upon the TTL (number of hop count) represented in forwarding query message. Accordingly, an appropriate decision taken based on the affiliation between Hop Count (HC) of query message and decision threshold (T). In the threshold phase (a short term or local space) the value of hop count is small and then flooding is performed on the basis of dynamic tuning forwarding probabilities. Secondly in the ultimate phase (Long-term space), a random walk search is consistently preserve the searching performance gained in the threshold phase.

i) Local search: when $HC \leq T$

If hop count is less than the decision threshold, directed flooding is carried out to flood among the peers in the network. Directed flooding is defined as peer forwards the query message probabilistically to a subset of neighbors that

has the ability to provide more high-quality search results than other neighbor peer in the network. At this point, a subset of neighbors to forward a query is determined from the product of degree of connectivity d and transmission probability p .

$$\text{Subset of neighbors to flood Query} = d * p$$

If $p=1$, then normal flooding takes place. Otherwise, directed flooding takes place that forwards the query message to a subset of neighbors ($d * p$)

ii) Ultimate Search: when $HC > T$

If hop count is greater than the decision threshold means then biased random walk strategy is used to flood the query among peers in the network. Biased random walk resembles that instead of forwarding queries to randomly chosen neighbors, each peer forwards the queries through an appropriate decision taken through knowledge learned from the previous experience to a selected neighbor has high probability to provide high quality.

4.3 Synchronized Caching

Our proposed approach is designed to significantly reduce the network search traffic without degrading query success rate. To improve the searching efficiency, synchronized caching [18] is well organized with a dynamic search approach which can selectively forwards query to the peers which has high probability to provide desired cache results. In this caching, each peer in the returning path can intermittently store the responses for previously requested queries that provide enough guidance for future message routing. Cached information is coordinated distributive among multiple peers in the network. An entry in a cache is composed of 2 fields: cached information and a timestamp. The main advantage of this cache arrangement is that to reduce query latency by the way of retrieving responses for previously requested queries directly from cache possessed matched responses among peers. There is no need to survive for searching to locate its original location whereas location of a data item is easily retrieved without convoluted searching and reduces network traffic. In addition to, cached information in low degree peer must be properly updated to its connected peer with high degree connectivity because query forwarding is carried out only in the overlay network [16].

4.4 Cache Consistency based on File Utility

Caching responses of previously requested queries among peers is a prominent technique to improve the performance of data accessibility in a P2P network which in turn reduces the access latency and bandwidth usage. When multiple peers cache the same files, then the cached files may become redundant. In order to avoid redundant files in cache, proper cache update is performed to maintain the cache consistency. In our proposed approach, cache consistency is maintained through Flexible polling mechanism (DPM) that utilizes Additive Decrease Multiplicative Increase (ADMI) algorithm [19]. This algorithm is mainly used to perform proper cache update where a file update rate predominantly depends on the file query rate. If certain files are frequently requested, then it is considered as high utility file. Obviously, the query rate of the high utility file is increasing correspondingly; therefore, file must be updated for every short interval time. On the other hand, if certain files are sparsely requested, then it is considered as the less utility file. It resembles that the query

rate of the less utility file is decreases correspondingly; therefore, the file can be updated even for every long interval time. The file update rate can be dynamically varied according to the file query rate. A suitable term to represent the time interval between update is represented as Time-To-Refresh (TTR) value which denotes the next time instant a peer has to maintain updated cached information. The TTR value is varied dynamically depending upon the file query rate.

If the file is frequently requested between two successive polls, then TTR value is reduced by multiplicative factor (a short interval of time).

$$TTR(n) = TTR(n-1)/\beta$$

Here, β is multiplicative constant. The TTR for n th event is determined from its TTR value of previous $(n-1)$ th event.

If the file is rarely requested between two successive polls, then TTR value is increased by additive factor.

$$TTR(n) = TTR(n-1) + \alpha$$

Here, α is additive constant. Then TTR for n th event is determined from its TTR value of a previous $(n-1)$ th event. The frequently requested file provides high utilization whereas rarely requested files are removed from the cache to maintain cache consistency

4.5 Replica strategies

The main reliability requirement for a P2P system is to maintain consistency between frequently updated or infrequently updated files and their corresponding replicas. The replication is mainly carried out in frequently visited node to provide high utilization of replicas whereas underutilized replicas are removed from the cache. File replication and update are performed by each node autonomously based on actual Query rate and update rate. File replication is also efficiently accomplished in high connectivity peer to obtain a high hit ratio with less average delay. In the proposed approach, the constructed power law network is more resilient for random node failure. Even, there are possibilities for a high connectivity peers to leave from the network due to the dynamic nature of the P2P network. Therefore, a replica of cached information in high connectivity peers must be maintained into the next high connectivity peer in that group. In case of departure of the high connectivity peers, communication in the whole network is completely disrupted. In case of any failure caused in high connectivity peer due to heavy load or dynamic nature, a back up of that node is provided.

4.6 Proposed algorithm

i) Query processing

// Source node S , Requested Query (Q), Requested resource (R), TTL, Decision threshold (T) is 5, Source peer (S), Destination (D), d is degree of connectivity, p is transmission probability.

Process KCAQS {Query (S , D , HC);

Generate $S \rightarrow$ Query(S , D , HC);

Check value TTL in Query message

If (TTL \leq T) // TTL=2;

{

FORWARD Query (Q) → Subset of neighbors;

Subset of neighbors = d*p; // Queries are forwarded to peers having high probability to provide high quality results within two hops.

Return (Require file(S);

}

else

If (TTL > T) // HC =7

{

HC=HC+1;

FORWARD Query (Q)(Randomly chosen neighbors; // Queries are forwarded to randomly chosen peers.

Return (Required file(S);

}

Query searching ends;

ii) Flexible Poll mechanism for cache consistency

T is maximum Threshold for file query rate.

TTR is Time To Refresh

If (File Query rate>T)

{

TTR (n) = $TTR (n-1) / (1 - \text{File Query rate})$ // Update is performed for a short interval of time.

(

}

else

If (File Query rate < T)

{

TTR (n) = TTR (n-1) + (// Update is performed for a long interval of time.

}

5. PERFORMANCE EVALUATION

5.1 P2P Environmental set up

An unstructured P2P network is constructed with 10,000 peers in a two stage power law distribution and each peer possess 24 neighbor peers. Experimental analysis is carried out to evaluate the effectiveness of the proposed mechanism. Each peer generates a query without any dependency to search resources in other peer. Generating queries are randomly distributed among peers in the network and each query has intent to receive 100 results for each. Moreover, the selection of TTL value strongly depends on the number of peers in the network. If the network contains 10,000 peers, the TTL value is set to 3 (TTL = 3). A large TTL value covers most of the peers but it is undesirable in the flooding scheme because it can generate a large number of query messages [20]. The low TTL value incurs less overhead but it does not covers the most of the peers in the network.

5.2 Performance Metrics

In the unstructured network, the aspect of coverage, success rate and flooding is the best efficient search algorithm regardless of crushing search cost. Besides that, if searching

cost is considered as an important measure, RW be the suitable searching algorithm in an unstructured system. Because of these circumstances, there is a need to evolve a new appropriate evaluation metric called as Search Efficiency (SE)[21] in order to demonstrate the effectiveness and efficiency of the proposed system. To measure the search efficiency of the proposed algorithm, we use the metrics network traffic, average query response time and query success rate.

5.3 Search Efficiency (SE)

Search Efficiency (SE) is an objective metric that affords general supervision to characterize the efficiency of the system. In the proposed scheme, Searching Efficiency (SE) is determined from three distinct factors such as Query Efficiency (QE), Responsiveness and Reliability [19].

$SE = \frac{\sum \text{Query Hits [t]}}{\sum \text{Query Messages [t]}} * 100\% * (\text{Success rate} / \text{Query Message} * R)$

5.3.1 Query Efficiency (QE)

Query Efficiency is evaluated from a number of Query Hits (QH) divided by the number of Query Messages (QM). In the best searching algorithm possess a large number of QH occurs even with few query messages.

$\text{Query Efficiency} = \frac{\sum \text{Query Hits [t]}}{\sum \text{Query Messages [t]}} * 100\% / \text{Query Message} * R$

Where QH is the query hits at any peers in hop h, T is TTL varies from (1 -7), QM is the total number of query messages generated during the query, R is the replication ratio of the queried object.

i) Network traffic

In an unstructured P2P networks, network traffic overhead is raised due to the total number of queries forwarded or received by peer during the search. Hence, average network traffic is the total amount of traffic generated during the whole number of queries processed in the network divided by the total number of queries totally generated by peers in the network [21].

$\text{Average network traffic} = (T_i / (Q_i))$

Where T_i is the total amount of traffic generated by queries and Q_i is the total number of queries generated by peer i.

5.3.2 Responsiveness

The main intention of efficient searching is to find out the desired data objects as soon as possible. Hence, a high responsiveness is required for good searching performance. Response Time (RT) is defined as the time taken to forward the query reply to the requested peer. In the other sense, RT states that the least number of hops exist to forward query reply to requested peer. In the proposed scheme, required data item is found with in h hops, then Query Hit is represented as $QH (t=h) = Q$. Responsiveness is inversely proportional to response time. The fast optimized search needs high responsiveness with less response time [21].

$\text{Query Efficiency} = \frac{\sum \text{Query Hits [t]}}{\sum \text{Query Messages [t]}} * 100\% / \text{Query Message} * R$

Where h is the query hit at Hop h.

1.1.1 5.3.3 Reliability

Reliability is measured by a success rate that considers the satisfaction level of user experience or perspective even when

two searches get the same number of searching cost and Query Hits.

i) Success Rate

The success rate of a P2P search is explicitly relies on how query processing efficiently perform requested queries.

$$\text{Success Rate} = \frac{\sum Si}{\sum Qi}$$

Where Si is the number of successful searches done by peer i.

Qi is the number of queries generated by peer i.

ii) Cache hit ratio

Cache hit ratio (CHR) is defined as the ratio of requested queries satisfied through cached information in a cache rather than broadcasting of query message. It is represented as a rate of the number of messages routed through cached content out of 100 messages.

$$\text{Cache Hit Ratio} = \frac{\sum Ci}{\sum Qi}$$

Where Ci is the number of successful searches routed through cache

Qi is the number of queries generated by peer I.

5.4 Effect of hop count

The performance of the proposed scheme is the effect of hop count on success rate, response time and traffic overhead. The proposed KCAQS scheme enrich searching and retrieval process i) To enhance searching process, queries are forwarded through knowledge intended searching where search is performed by either suing flooding or random walk which is strongly depends upon the hop count value. ii) To reduce network traffic and response time, Information retrieval process is enhanced through synchronized caching approach, where low connectivity peers updates their cached responses to high connectivity peers because Query searching is carried out only in a overlay network. To avoid redundant data, cache consistency is achieved by Flexible Poll Mechanism (FPM) with Additive Decrease Multiplicative Increase algorithm. Therefore, this considerable enhancement in Query searching is used to attain a high success rate with reduced network traffic and response time.

i) Success Rate

High Query success rate is obtained when responses for requested queries are successfully delivered to requested peer. If hop count value increases means then obviously query success rate also increases in the sense of large area coverage and widening searching. In the proposed scheme, success rate is mainly improved due to knowledge intended Query forwarding with adaptive searching based on hop count.

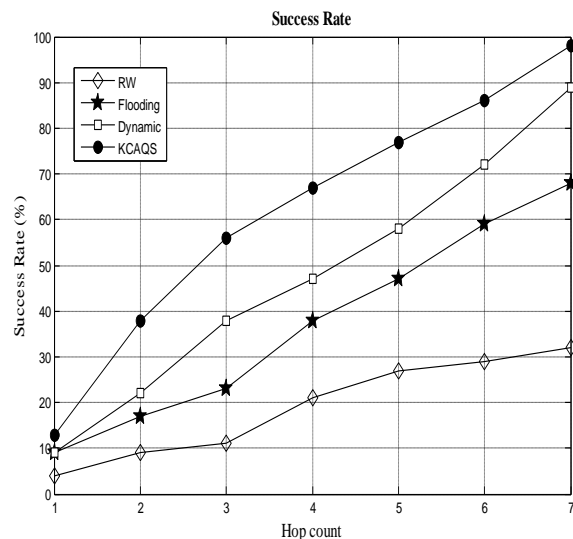


Fig 1. Query Success Rate

The Fig 1 depicts that proposed KCAQS achieves high success rate than already searching schemes such as flooding, RW and dynamic searching. In flooding, a appreciable success rate is attained than random walk because queries are completely flooded throughout the network. Recently evolved dynamic searching scheme achieves less 78% success rate where as the proposed KCAQS can improve the query success rate up to 92% because of knowledge intended query forwarding with adaptive query searching based on hop count.

ii) Average Network Traffic

Generally in P2P searching, high traffic overhead is incurred due to the large number of data and control messages are propagated to transmit query. In the proposed scheme, volumes of traffic per query decreases due to two enrich tasks such as i) knowledge intended searching with Adaptive searching either flooding (for smaller Hop Count) or random walk (large Hop Count). ii) Query results for already requested queries are easily retrieved from a cache of high connectivity peer which has responses from bypassing previously requested queries iii) In case of maximum load condition occurs on high connectivity peer means then load balancing is achieved due to that the directed next high connectivity peer in a group is already available to maintain the replica of cached information in high connectivity peer.

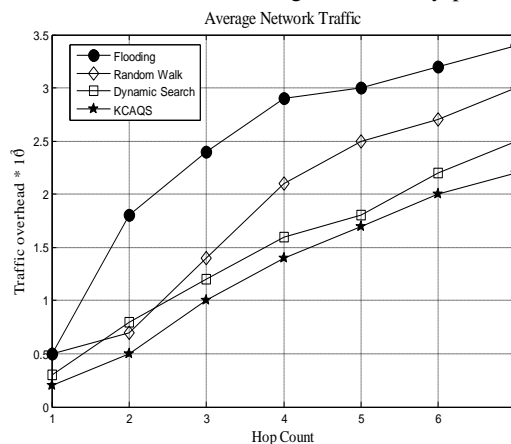


Fig 2. Average Network Traffic

Fig 2 depicts that flooding incurs high network traffic due to flooding of queries throughout peers in the network. Whereas when compared to flooding, random walk and dynamic searching schemes achieves less network traffic. Furthermore, our proposed scheme incurs considerably 40% less network traffic than other searching schemes. Overall, the probability of finding a resource is strongly depends on the chosen TTL value that also wider the spreading of the query among most of the peers in the network with reduced network traffic.

iii) Response time

Response time clearly represents that number of hops to forward query reply to the requested peer. In the proposed KCAQS scheme, response time for requested query is considerably reduced through caching of responses for already requested queries in high connectivity peers. It is easy to retrieve requested data item, when requested query is already previously requested from other peers in network. The number of queries requested from peers increases, then number of cached data item also increases. Hence, obviously the increasing number of cached data item finely meets the queries requested from the peers when the number of queries requested increases.

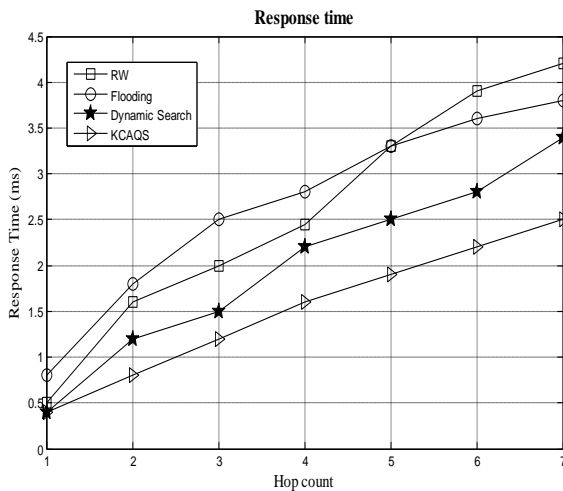


Fig 3. Response Time

Fig 3 shows that random walks takes longer search or response time because queries are only forwarded to randomly selected peers available in higher hops. Whereas flooding and dynamic searching takes less response time than random walk. However, our proposed scheme achieves considerable reduction in response time with 20 percent lesser hops to forward reply than existing searching scheme because most of the requested query is satisfied by caching.

iv) Searching Efficiency (SE)

Searching efficiency is the unified performance metric that is effectively magnified with respect to average network traffic, Response time and success rate. Therefore, this metric comprises to measure the overall efficiency of the proposed P2P searching scheme. Searching Efficiency of the proposed scheme is compared between searching with a replica and without replica.

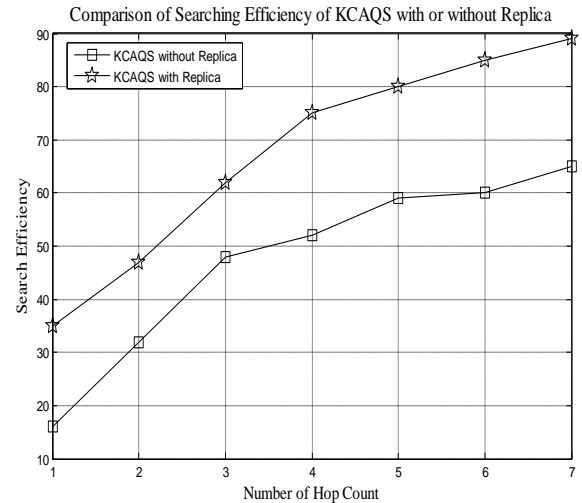


Fig 4. Search Efficiency

Fig 4 shows that the search efficiency of proposed scheme with a replica is greater than without replica. In the proposed scheme without replica, searching efficiency is not appreciable where maximum load is occurred due to the maintenance of cached responses of all peers in its corresponding high connectivity peers in the overlay networks. On the other hand, proposed scheme with replica manages the heavy load in high connectivity peers due to the maintaining of original replica of cached responses in high connectivity peers into the cache of next directed high connectivity peer. Therefore, incase of failure of high connectivity peer due to the dynamic nature of P2P network, the next directed high connectivity peer takes the responsibility. However, in our proposed scheme with replica, searching efficiency is completely improved due to enhanced metrics such as response time, average network traffic with high load balancing and success rate.

5.4 Effect of number of queries

i) Cache hit ratio

In our proposed scheme, most of the queries are satisfied through less search space because cached responses in all low connectivity peers are properly updated to the high connectivity peers. If number of requested queries increases, then the cached responses in each peer also increases. Therefore, due to the occurrence of large number of cached responses in less searching space, high cache hit ratio is achieved.

Fig 5 shows that the proposed scheme achieves a high Cache Hit Ratio (CHR) because responses of previously requested queries are cumulatively maintained in high connectivity peers. Hence, obviously most of the requests are satisfied by high connectivity peer itself in the overlay network and it leads to high cache hit ratio in the P2P searching schemes.

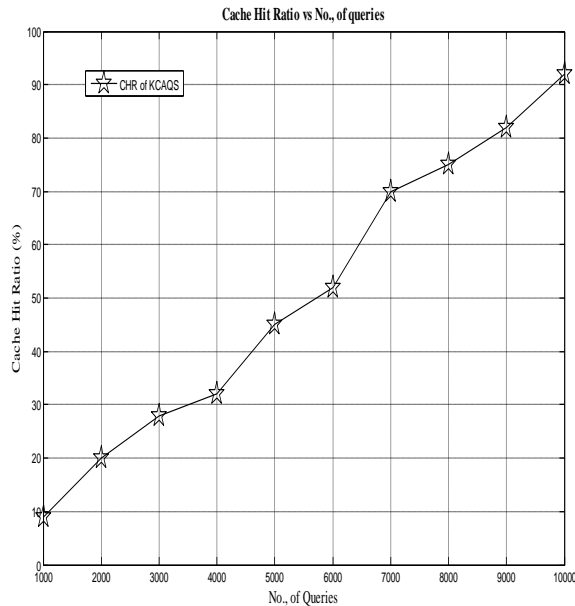


Fig 5. Cache Hit Ratio

6. CONCLUSION

In this paper we proposed a search algorithm, called as Knowledge and Cache based Adaptive Query Searching (KCAQS). It is an efficient and adaptive search mechanism which is derived from flooding and random walk. The proposed scheme can perform efficient knowledge intended forwarding where queries are only forwarded to selected neighbors that have ability to provide high quality results. Hence, search performance is improved through efficient cache based search mechanism in less search space of overlay networks. Moreover, the proposed scheme is strengthened using synchronized caching which caches the frequently requested responses in high connectivity peers of less searching space. In order to avoid stale data, cache consistency is achieved through Flexible Polling Mechanism (FPM) oriented Additive Decrease Multiplicative Increase (ADMI) algorithm. The proposed search mechanism has improved search efficiency with high success rate, less response time and reduced network traffic.

7. REFERENCES

- [1] Dejan S. Milojcic, Vana Kalogeraki, Rajan Lukose, Kiran Nagaraja, Jim Pruyne, Bruno Richard, Sami Rollins , Zhichen Xu, 2003. Peer-to-Peer Computing, Technical Report, Hewlett-Packard.
- [2] Choon Hoong Ding, Sarana Nutanong, and Rajkumar Buyya, 2003. Peer-to-Peer Networks for Content Sharing, Technical Report on Grid Computing and Distributed Systems Laboratory, University of Melbourne, Australia.
- [3] Yatin Chawathe, Sylvia Ratnasamy, Lee Breslau, 2003. Making Gnutella like P2P Systems Scalable, Proceedings of the International conference on Applications, technologies, architectures, and protocols for computer communications, New York, NY, USA.
- [4] S Schmid, R Wattenhofer, 2007. Structuring unstructured peer-to-peer networks, Proceedings of the 14th Annual IEEE International Conference on High Performance Computing (HiPC).
- [5] William Acosta Surendar Chandra, 2005. Unstructured Peer-to-Peer Networks - Next Generation of Performance and Reliability, Proceedings of the Twenty-fourth Annual Joint Conference of the IEEE Computer and Communications Societies.
- [6] Özgür Ulusoy, 2007. Research Issues in Peer-to-Peer Data Management, Proceedings of 22nd International symposium on computer and information sciences.
- [7] Qin Lv, Pei Cao, Edith Cohen, Kai Li ,Scott Shenker, 2002. Search and Replication in Unstructured Peer-to-Peer Networks, Proceedings of the 16th international conference on Supercomputing.
- [8] Christos Gkantsidis, Milena Mihail, and Amin Saberi, 2004. Random Walks in Peer-to-Peer Networks, Proceedings of the Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies.
- [9] Ming Zhong, Kai Shen, 2009. Popularity Biased Random Walks for PeertoPeer Search under the Square Root Principle, In the proceedings of 5th International Workshop on Peer to Peer Systems.
- [10] Gang Chen, Chor Ping Low, and Zhonghua Yang, 2008. Enhancing Search Performance in Unstructured P2P Networks Based on Users' Common Interest, IEEE Transactions on Parallel and Distributed Systems, Volume 19, Issue: 6 , pp: 821 - 836.
- [11] Kai-Hsiang Yang, Chi-Jen Wu and Jan-Ming Ho, 2006. Ant Search: An Ant Search Algorithm in Unstructured Peer-to- Peer Networks, Proceedings of 11th IEEE Symposium on Computers and Communications.
- [12] Lakshmi Ramaswamy and Jianxia Chen, 2008. The CoQUOS Approach to Continuous Queries in Unstructured Overlays, IEEE Transactions on Parallel and Distributed Systems, Volume 19, Issue: 6 , pp: 821 - 836.
- [13] Song Jiang, Lei Guo, Xiaodong Zhang, and Haodong Wang, 2008. Light Flood: Minimizing Redundant Messages and Maximizing the Scope of Peer-to-Peer Search, IEEE Transactions on Parallel and Distributed Systems, Volume 19, Issue: 5, pp: 601-614.
- [14] Vicent Cholvi, Pascal Felber and Ernst Biersack, 2004. Efficient search in unstructured peer-to-peer networks, Proceedings of the sixteenth annual ACM symposium on Parallelism in algorithms and architectures.
- [15] Lada A. Adamic, Rajan M. Lukose, Amit R. Puniyani and Bernardo A. Huberman, 2001. Search in power-law networks, International Journal on Networking and Internet Architecture, Volume: 64, Issue: 4.
- [16] Tsungnan Lin, Pochiang Lin, Hsinping Wang, and Chiahung Chen, 2008. Dynamic Search Algorithm in Unstructured Peer-to-Peer Networks, IEEE Transactions on Parallel and Distributed Systems, Volume 20, Issue: 5, pp: 654 – 666.
- [17] Dimitrios Tsoumakos, Nick Roussopoulos, 2003. Adaptive Probabilistic Search (APS) for Peer-to-Peer Networks, Proceedings of the Third International Conference on Peer-to-Peer Computing.
- [18] Chen Wang, Li Xiao, Yunhao Liu, and Pei Zheng, 2006. DiCAS: An Efficient Distributed Caching Mechanism

for P2P Systems, IEEE Transactions on Parallel and Distributed Systems, Volume 7, Issue: 10, pp: 1097-1109.

- [19] Haiying Shen, 2010. IRM: Integrated File Replication and Consistency Maintenance in P2P Systems, IEEE Transactions On Parallel And Distributed Systems, Volume 21, Issue 5, pp:100-113.
- [20] Matei Ripeanu, Ian Foster, 2001. Mapping the Gnutella Network: Macroscopic Properties of Large-Scale Peer-to-Peer Systems, Proceedings of the Revised Papers from the First International Workshop on Peer-to-Peer Systems.
- [21] Hsinping Wang' and Tsungnan Lin, 2005. On Efficiency in Searching Networks, Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies.

Mr. G. Ramachandran received the B. E degree in Computer Science and Engineering from Annamalai University in the year 1997. He received the M.E degree in Computer Science and Engineering from Annamalai University, Annamalai nagar in the year 2005. He has been with Annamalai University, since 2000. He is doing his Ph.D in Computer Science and Engineering at Annamalai University. He published 4 papers in international conferences and journals. His research interest includes Computer Networks, Network Security, Mobile Ad hoc Networks and Network Simulator.

Dr. K. Selvakumar received the B.E degree in Electronics and Communication Engineering from Kongu Engineering College in 1989. He received the M.E degree in Communication Systems from Regional Engineering College in the year 1997. He has been with Annamalai University, since 1999. He completed his Ph.D degree in Computer Science and Engineering at Annamalai University, in the year 2008. He published 30 papers in international conferences and journals. His research interest includes Computer Networks, Cryptography and Network Security, Wireless Networks, Mobile Ad hoc Networks and Network Simulator.