Elephants Journey towards Successful Resource Discovery in Unstructured P2P Networks

Sabu M. Thampi[†], **Chandra Sekaran K**[††]
[†]L.B.S Institute of Technology for Women, Kerala-695012, India
[††]National Institute of Technology Karnataka, Surathkal, Karnataka-575025, India
smtlbs@yahoo.co.in, kch@nitk.ac.in

Abstract- **This paper presents a resource discovery scheme for decentralised unstructured P2P file sharing applications. The scheme utilises the principles underlying elephants migration from dry place to green fields for food. Hence, the proposed approach is a bio-inspired swarm intelligence based search technique. The aim of the technique is to route a query from a node to suitable peers in the network to find the required object. For this, the scheme divides the resource available areas in the network as wet areas and dry areas. The wet areas are resource fertile areas and dry areas have very limited resources. When a node is located in a dry area, its queries are propagated to wet areas for increasing the performance of resource discovery process. At the same time, the powerful nodes try to make the dry area into a resource lush one so that nodes that have moved to wet area return home for resource discovery. Simulation results show that the proposed scheme significantly increases query success rate and reduces the network traffic as the resources are effectively distributed to well-performing nodes.**

Keywords: Unstructured P2P, Bio-inspired computing, elephants migration, searching, replication.

I. Introduction

In client/server systems, resources are concentrated on a few number of servers. In order to increase the performance of such systems, complex load balancing and fault-tolerance techniques are to be employed. Limitation on network bandwidth further increases the complexity of client/server systems. Peer-to-peer (P2P) networks which distribute the processing loads and network bandwidth among participating nodes have been emerged as an alternative for client/server systems. The participating nodes in a P2P system make at least part of their resources as 'shared', allowing other contributing peers to access these resources. There are two kinds of P2P architectures: centralized and decentralized. Decentralized P2P systems are further classified as decentralized structured and decentralized unstructured. This paper focuses on decentralized unstructured P2P networks. In a decentralized unstructured P2P system there is neither a centralized index nor any strict control over file placement. Structured P2P systems strongly connect the placement of files with the structure of the overlay network. These systems maintain hash-table-like lookup/insert functions. These structures are called distributed hash tables [1]. Examples of structured P2P systems include Chord [2], Pastry [3], Tapestry [4], and CAN [5]. Gnutella [6], KaZaA [7], and Limewire [8] are examples of decentralised unstructured P2P applications. Recently, there has been an extensive amount of research on unstructured P2P networks.

There are several issues in unstructured P2P networks. Among the various problems, the two important problems are: (a) efficient resource discovery and (b) improvement of objects' availability in the system. The solution for the resource discovery problem is to use a search algorithm that finds the resources based on the local information about the network. A good search mechanism should provide several matching queries with short hop distances thus saving valuable network resources such as bandwidth. The searching process includes features such as the technique for forwarding queries, the set of nodes that receive query-related messages, the structure of these messages, local processing, stored indices and their maintenance. Designing a good search mechanism is difficult in P2P systems for several reasons, including scale of the system and unreliability of individual peers [9]. In a P2P network, the quality of query results is measured by the number of results and efficiency in object discovery measures search accuracy and the quantity of discovered objects for each request.

The success of a search technique largely relies on the availability of appropriate shared objects (files) in nearby by nodes. However, as a result of irregular node failure, unreliable network connectivity and inadequate bandwidth, efficient sharing of resources becomes more difficult in a P2P network. A prominent technique for improving availability and fault-tolerance is replication. The idea of replication is that if several copies of data subsist on autonomous nodes then the prospect of at least one copy being accessible is increased. Even if a node is not up, due to replication the files available in the node may be made available in some other nodes.

One of the well-known applications of P2P systems is file sharing. The files are shared among nodes dispersed in various locations of a P2P network. Any user can send a query to access the required file existing in any of the nodes in the

978-1-4244-4793-0/09/$25.00 ©2009 IEEE

network using a resource discovery scheme. Several categories of search techniques are mentioned in the literature. A few of the search techniques are discussed briefly in related work section.

In the past, people discovered a variety of the interesting insect or animal behaviours in the nature. A congregate of birds sweeps across the sky. A group of ants hunt for food. A group of fish swims, turns, flees together... - we call this kind of collective motion as swarm behaviour. Recently biologists and computer scientists in the field of artificial life (*also called ALife - describes research into human-made systems that possess some of the essential properties of life*) have studied how to model biological swarms to understand how such social animals interact, achieve goals, and evolve [10,11,12]. Biologically inspired computing is an area of computer science which uses the advantageous properties of biological systems. It is the amalgamation of computational intelligence and collective intelligence. Biologically inspired mechanisms have already proved successful in achieving major advances in a wide range of problems in computing and communication systems. The bio-inspired computing is connected with several fields such as artificial neural networks, evolutionary algorithms, swarm intelligence, artificial immune systems, DNA computing and quantum computing.

This paper presents a biologically inspired swarm intelligence based resource discovery scheme for decentralised unstructured P2P networks. It utilises the principles of African elephants migration during dry seasons for food. The proposed scheme classifies a node in the network as ordinary nodes and power nodes. The presence of less popular objects in its neighbourhood makes a peer reside in the *dry region* and hence for finding out more popular resources it moves to a resource rich region called *wet region*. Due to this success rate is improved a lot. Meanwhile, powerful nodes in the network try to increase the availability of popular objects in the dry region. This leads the nodes to shift to the previous region after the nodes satisfy with the presence of sufficient quantity of popular objects. Thus, the network load is managed effectively.

The rest of the paper is organised as follows. Section II briefs the related work. Section III briefly explains the process of elephants migration. Section IV describes the proposed resource discovery scheme for decentralised unstructured P2P networks. The simulation environment is discussed in section V. Section VI presents the results and a discussion on them. Finally section 7 concludes the paper.

II. Related Work

The search schemes for unstructured P2P network are generally classified as *blind* [13, 14, 15, 16] and *informed* [17, 18]. Most of the search techniques are either random techniques or probabilistic schemes. In a blind search, nodes do not keep information about object location. In an informed search, nodes gather some metadata that assist the search operation.

Bio-inspired approaches are recently introduced in a variety of ways for different applications such as resource discovery and packet routing. A hybrid ant-inspired search algorithm (HASA) for P2P media streaming distribution in Ad Hoc networks is proposed in [19]. It utilises the merits of random walkers and ant-inspired algorithms for search in unstructured P2P networks, such as low transmitting latency and less redundant query messages.

[20] focuses on the free-rider problem in unstructured P2P networks, and proposes a new search algorithm, called "AntSearch", to reduce the redundant messages during a query flooding. Each peer maintains a pheromone value to present its success rate of past processed queries, and keeps a list of pheromone values of its immediate neighbours. These pheromone values are used to flood a query through those peers which are not likely to be free-riders. The main idea of the AntSearch is using pheromone values to identify the free-riders, prevent sending messages to those peer in order to reduce the redundant messages.

A Swarm Intelligence Technique Bees Algorithm called P2PBA (Peer to Peer file sharing - Bees Algorithm) which is based on the lines of food search behaviour of Honey Bees is proposed in [22]. The scheme optimizes the search process by selectively going to more promising honey sources and scan through a sizeable area.

[21] investigates a multi-swarm approach to the problem of Neighbour Selection (NS) in P2P networks. Particle swarm share some common characteristics with P2P in the dynamic socially environment using particle swarm optimization algorithm. Each particle encodes the upper half of the peer-connection matrix through the undirected graph, which reduces the search space dimension.

[23] proposes a P2P network based location search algorithm which can be used to establish connections in Internet Telephony. By using the location search algorithm, the caller can identify the peer it is calling. The algorithm is based on the notion of *gradient search* and is applicable to unstructured networks. It is inspired by a biological phenomenon called *haptotaxis*. The algorithm performs at par with DHT-based location search algorithms.

The search algorithm proposed in [24, 25] is termed ImmuneSearch which draws its basic inspiration from natural immune systems. It is implemented separately by each individual peer participating in the network and is completely decentralized in nature. Instead of flooding query messages, ImmuneSearch uses an immune system. It is inspired by the concept of affinity governed proliferation and mutation for message movement. In addition, a protocol is formulated to change the neighbourhoods of the peers based upon their proximity with the queried item. The topology evolution coupled with proliferation and mutation help the P2P network to develop 'memory'. Due to this, the search efficiency of the network improves as several individual peers perform search. [26] presents SemAnt, an algorithm for distributed query routing based on the Ant Colony Optimization meta-heuristic.

The technique proposed in this paper is the first resource discovery mechanism for P2P networks utilising elephants migration policy. The scheme is intended for increasing query success rate and reducing network traffic.

III. Migration of Elephants

Elephants are incredibly social creatures that have lasting memories, and can communicate over long distances through low range sound waves. They show a range of cognitive abilities and social behaviour. The African elephant is the largest living land mammal. There are two kinds of elephants: *African* and *Asian elephants*. African elephants are larger than Asian elephants. African elephants inhabit a diverse array of habitats [27]. The elephants migrate and normally follow the same migratory routes every year. The migration occurs typically at the commencement of dry season. The animals move toward more suitable locations near rivers and water sources. The environmental conditions considerably affect the migration distances. The distance being covered during migration by African elephants is more than 100 km in dry seasons. Asian elephants residing in the dense forests of southern India, travels between 20 and 50 km during migration. However, when the rainy season appears, elephant herds go back to native regions to feed on the lush green vegetation. Thus, the migration of elephants allows time for the re-growth of plants in fatigued scraping areas [28, 29].

IV. Proposed Bio-Inspired Search Scheme

The P2P system model comprises nodes and files (objects). There are a few neighbouring nodes (n_1, n_2, $n_3 \ldots n_n$) associated with each node 'p'. The term 'file' stands for any general content in a node or peer. A file can have more than one replica in the system. The number of neighbours connected (links) to a node is called its degree. Peers with a large number of degrees receive more number of queries than peers with a small number of degrees. This is due to high availability of objects and more number of links in high degree peers. The topology of P2P networks is modeled as a network with an undirected graph G whose nodes represent hosts and edges represent internet connections between those hosts. Nodes are usually very dynamic, where some can join and leave the network in the order of seconds whereas other nodes stay for an unlimited period of time. When a user requests a file, a search for the file is initiated and other nodes in the network need to be queried if the file is not available locally. A query is composed of one or more required words. In this paper, the terms 'peer' and 'node' are used in an interchangeable manner. Every query message has a message-id, and this ID is stored in a peer receiving the query. Hence, if the queries with same message–id come to the same peer another time, the query is discarded.

The random walk proposed in [15] with a few added features is employed for discovering resources in the P2P network. For a query generated in a node N, the shared folder of the node which contains the shared objects is checked for a query match. In case the desired object is found, query is dropped; else the query is forwarded to K walkers which are neighbours of N. The life of a walker is determined by the current value of Time-To-live (TTL) parameter. TTL is the number of hops to be visited during a search operation. However, each neighbour forwards a walker to only one node and from that node onwards the query is forwarded as in the previous hop until the TTL expires or desired result is found. If the popular objects are not available in nearby nodes; the query will take maximum TTL value and thus the number of messages increase drastically creating heavy network traffic. At the same time, the success rate decreases. The querying node may unhappy with the system and decides to leave the system permanently. Hence, queries should be appropriately routed to well-performing nodes and a proper mechanism for increasing availability of popular objects in all categories of peers should be employed. The first issue discusses a suitable criterion for routing queries in case of shortage of popular objects in the current search path. The later demands a suitable replication scheme for replicating popular objects to well-performing nodes for increasing the availability and fault-tolerance. Due to the dynamic nature of nodes in an unstructured network, nodes may come up and down frequently. In order to provide efficient access to popular files for all nodes in the network, the popular objects are replicated to good nodes which increase the availability.

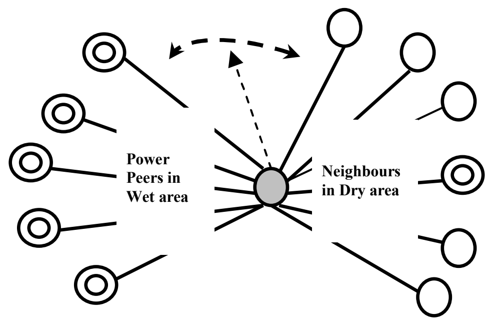

Figure 1. Dry and Wet areas in a P2P network

The migration policy of elephants influences the proposed resource discovery scheme. Each query is treated as an elephant. Each node houses a collection of objects. A group of peers hosting the resources within certain hop-limit forms an *area*. The area may be a *dry area or wet area* depending on the number of popular objects existing in the area. The service a node in the network may receive within a dry area is very limited as the area contains less number of popular objects. Hence queries have to travel long distances for successful resource discovery and most of the queries may fail. When the number of hits in the dry area reaches below a threshold, the queries are forwarded only through powerful peers situated in the wet area. The peers located in wet area are called power peers. The wet area hosts large number of popular objects, and at the same time a proper mechanism for moving popular objects from wet area to dry area is required. Nodes having several power peers as neighbours may receive more query hits since queries from power peers are only passed through other power peers. Hence, such nodes may be always situated in a wet area.

A power peer is a powerful node in the sense that it has high degree; large number of popular objects, and high bandwidth. A bunch of power peers form a wet area. These power peers together participate in the search process once a node doesn't receive adequate hits in a dry area. A node maintains a list of power peers in a data table. This table is called *power peer table*. During a search process, for each query hit, a querying node checks the category of node from which a success has occurred. If the node is a power peer, and its identity doesn't exist in the table, an entry for the power peer is created. Thus, the number of entries in the data table increases with number of query hits through different power peers in the network. Finally, a node may have a collection of power peers located in the network. The IDs of power peers along with their bandwidth, and degree are recorded in the table. The table also maintains data about number of hits in each power peer for the queries originated from the node.

For each neighbour, a node 'N' in the network maintains a variable which contains the number of hits through the neighbour for the queries submitted to it for a certain period of time 't'. The hit rate is also recorded. Hit rate is computed as the ratio between number of hits to number of queries inputted. The average of hit rate for all neighbours for time 't' is computed and if it is less than a threshold (δ), the node is said to be located in a dry area. After that, the node tries to move from dry area to wet area due to low hit rate it receives. The future queries are forwarded only through the power peers chosen from its power peer table. For that, the values of number of hits (h_p) through the power peer for the queries originated from the node, degree (d_p), and bandwidth (b_p) are utilised. For each power peer in the table, a *utility value* (u_s) is computed as in equation (1). K power peers with high utility values are chosen for routing K walker messages. The utility value for a power peer is updated based on the result of every query message forwarded to it. The utility value increases with high success rate, association with more neighbours, and high bandwidth.

$$u_s = (w_1 * h_p + w_2 * d_p + w_3 * b_p) * 100 \text{ ....}(1),$$

where $w_1 + w_2 + w_3 = 1$, w_1, w_2 and w_3 are the assigned weights for normalising the values.

A power peer forwards a query message to another power peer only. Thus, the queries are passed through a resource fertile wet area. The walkers are terminated when either result is found or TTL is exhausted. The processing of queries in a wet area may provide improved success rate with less search path length as the power peers hold more number of popular objects.

Shifting queries from dry area to wet increases the query load among power peers. Hence, appropriate measures are required to handle this situation. In the first mechanism, when CPU load on a power peer reaches certain threshold, the overloaded power peer forwards the incoming queries to other power peers with next higher utility values in the power peer table. The second mechanism is to increase the availability of popular objects in the dry area. This is done using a suitable replication technique. A revised replication scheme (Q-replication) proposed in [30, 31] is used for autonomously replicating objects to well-performing nodes in dry as well as wet area. The autonomous replication scheme employs Q-learning concepts for replication. In the context of replication, a well-performing node is one which possesses high bandwidth, high degree and large available free storage. Hence, even if a node shows low performance in resource discovery, due to the presence of said parameters it is selected as a candidate node for hosting a replica. Due to replication a dry area gradually becomes green.

In the autonomous replication scheme, every peer maintains a Q-table which contains IDs of neighbours and peers within n-hop distances. The past performance of peers listed in the Q-table is represented as a value. This value is called Q-value. However, in the proposed technique a little deviation from the original proposal as in [30, 31] is required in order to consider the dry and wet areas. When a node shifts to wet area, the neighbours of the node don't further directly receive any queries from the node. With the purpose of increasing availability of objects, neighbours possessing good values for bandwidth, degree and free storage are the good candidates for hosting new replicas. So, other power peers listed in the power peer table can accommodate a few selected neighbours of the node N in their replication Q-table. The higher the values of bandwidth, degree, and the amount of free storage available in the shared folder of every neighbour (f_p), higher the chances of getting accommodation in Q-table of a power peer. For this, every neighbour of the node transfers its bandwidth and free storage level to the node based on its request. The node computes another utility value (u_p) as in equation (2). The neighbours of node N possessing utility values greater than a threshold are accommodated into the Q-table of power peers listed in the power peer table. We assume that every power peer permits an ordinary node to assign its well-performing neighbours in their replication Q-tables.

$$u_p = (w_1 * f_p + w_2 * d_p + w_3 * b_p) * 100 \text{ ....}(2)$$

Next step is to assign the chosen neighbours to the power peers in the table. This is done by the node itself using the utility values of power peers - u_s. The node N computes the ratio (n_p) between number of chosen neighbours and number of power peers in the power peer table to the nearest integer. This value represents the number of neighbours to be assigned to the Q-table of each power peer. The u_s and u_p values are sorted and the node N compares both values. Power peers having higher utility values of u_s assigns n_p neighbours with higher utility values for u_p in its Q-table. The number of power peers or chosen neighbours may be odd or even. Hence, it is not mandatory that all power peers will receive the same number of neighbours of node N during this process. All the new members which are entered into the Q-table of each power peer receive an initial Q-value of 100. According to the Q-values, the new entrants receive replicas of popular objects from power peers. Thus, the availability of popular objects is augmented.

Once a dry area is crowded with sufficient number of popular objects, the area gradually becomes wet. Hence, the neighbours that are shifted to the power peers can attain the previous status to receive incoming queries from N. A few steps are to be done for completing this process. Two values are associated with this operation – an availability threshold value (λ) and hit rate threshold value (H_{thrld}). The ratio (n_{avbl}) of number of files replicated to the neighbour of node N listed in the Q-table of each power peer (N_{repl}) and the sum of number of files actually present at the time of shifting to Q-table of power peer and N_{repl} is computed. If this ratio is greater than or equal to the availability threshold, the node N is informed and it records the status of the neighbour. Thus, when x% (say 80%) of neighbours possess high availability, N starts sending its query requests to the neighbours. The neighbours for whom a green signal is obtained from a power peer in the power peer table are only considered for sending queries. The routing through power peers in the power peer table is temporarily suspended. Moreover, the neighbours of a node act as neighbours for other members in the network. Hence they may also receive replicas from those member nodes. These replicas are taken into account while computing the replication ratio. As said previously, the hit ratio is (H_t) also computed for each neighbour for a certain period. The power peer which holds the neighbour in its Q-table periodically collects the hit ratio from N and if the hit rate is greater than or equal to H_{thrld}, the power peer assumes that node N is situated in a wet area. However, the power peer continues to serve the node by replicating popular contents to neighbours in the Q-table. But, the neighbours in the Q-table which are having a n_{avbl} less than half of the threshold value up to the period are removed from the Q-table due to their low performance in replicating objects autonomously. This also saves computational resources of power peers.

The major steps in the proposed search technique are portrayed as an algorithm below:

If $(AvgHit \quad NOT \quad < \delta)$ then
    Node N Continues searching through its neighbours.
else
 a. Node is in dry area – DRY.
 b. Node moves to wet area – WET.
 c. Starts searching through selected power peers in power peer table.
 d. If query load on a power peer exceeds to certain level, redirects the incoming queries to suitable power peer in the power peer table based on its utility value.
 e. Power peers replicate popular objects to DRY area.
 f. If $(n_{avbl} \geq \lambda)$ for x% of neighbours of N then
    i. Node starts sending queries to neighbours in DRY area.
    ii. Compute hit rate - H_t for period ‘t’.
    iii. If $(H_t \geq H_{thrld})$ then
       a. Area DRY is declared as new WET area
       b. Suspend the sending of queries to power peers located in WET area (power peer table).
       c. Node N continues searching through its neighbours as previously.

V. Experimental Setup

The proposed resource discovery technique is simulated using random graphs that have 10000 nodes. The nodes can join the network and establish random connections to existing nodes. Each node carries a number of files. The average degree of a node in the network is 3.5. There are 1000 objects distributed to various nodes, the objects are replicated to different sites using autonomous replication algorithm. The quantities of objects maintained in the system are sufficient to analyse the performance since autonomous replication scheme effectively propagates the objects to various sites. The objects are word, PDF and text files available as course materials on various subjects such as computer science, electronics, physics, mechanics, electrical etc. Thirty thousand keywords are chosen from the course material files and these words are randomly selected as query keywords by all the nodes during searching. Two types of query search are employed: file name based and keyword based. In the file name based search only the objects names in the shared storage space of each node is searched. The objects containing the keywords are looked up in the keyword-based search. In the simulation scenario, all the queries contain keywords alone.

Table 1. Simulation Parameters

Parameters	Default Values
Topology	Random
Network type	Unstructured
No. of nodes	10000
Average Degree	3.5
TTL	06
Number of Walkers, k	06
Number of objects	1000
Replication Method	Q-replication
Peers	Ordinary peers, power peers
Power peer	Node degree ≥7; available storage ≥ 30% of the shared storage space; Number of shared objects in a node ≥ 15

Each node generates 100 queries and one query is propagated every 20 seconds on average. However, each node enters the query generation phase in a randomly selected time slot. Hence, the flood of query message production is regulated. Eighty percent of the nodes are up at the time of performing simulation. Fifty percent of ‘Down’ nodes selected randomly change their status to ‘UP’ after every 50,000 queries are propagated and, at the same time, the same amounts of UP nodes obtain the DOWN status. Separate rules for forming dry and wet areas are not employed. Based on hit rate a node moves to wet region. The hit rate threshold δ is preset as 0.3 and the value of H_{thrld} is 0.6. Availability threshold λ is fixed as 0.4.

Table 1 lists the various simulation parameters and their default values. The default TTL value is preset as six. There are ordinary nodes and power nodes. The power peers are selected based on node degree, number of objects being hosted and available storage. The minimum degree of a power peer is preset as seven. Initially twenty percent of the nodes in the network are assigned the power peer status. The maximum cpu_load that can be processed is represented as number of messages a node can handle at a time. This value remains the same for each power peer until the end of simulation.

The simulation tool has been developed using Java language. The tool runs in a Windows operating system environment. The software, which are used for developing the simulation software are NetBeans, J2SE Development Kit 5.0 and WampServer. NetBeans [32] is a free, open-source Integrated Development Environment, which supports development of all Java application types. WampServer [33] is an open source project and Windows web development environment. It allows creating various applications with Apache, PHP and the MySQL database. WampServer also includes PHPMyAdmin and SQLiteManager for managing databases. The simulations are conducted in systems with Intel Xeon (Quad Core) processor,12 MB L2 Cache, 1333 MHz FSB, 4 GB, and 146GB SAS HDD(15K RPM). All the necessary software mentioned above are installed in the systems for conducting various simulation experiments.

VI. Results and Discussion

Experiments are conducted several times with the objective to measure query success rate and messages per query. The details of experiments conducted are discussed below.

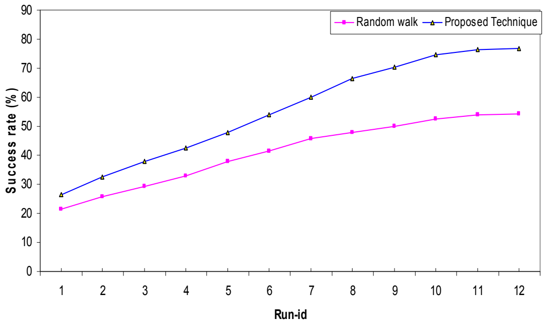

Figure 2. Query hit rate for random walk and proposed scheme

Query Success rate- random walk vs. Proposed scheme: The aim of conducting this experiment is to measure the success rate for queries generated by different nodes in the network. The experiments for the proposed technique are conducted as said environment and for random walk a different environment is used. For testing the performance of random walk search technique, no means of replication is employed. Moreover, all nodes have equal status. Experiments are conducted several times and the results are plotted as a graph shown in figure 2. From the graph we can conclude that the proposed search technique performs far better than random walk in terms of success rate. The success rate of the proposed scheme increases with time and some times later the rate of success goes steadily. This is due to the effective distribution of popular objects to well-performing nodes by replication. So, more regions gradually become fertile causing increased success rate.

Average % of query success rate through ordinary nodes and power peers: Aim of this experiment is to measure contribution of neighbours of peers and other power peers in the network towards query success rate for the proposed technique. Initially, most of the queries are failed. Gradually the situation changes. As more nodes are moved to wet area, the successes through power peers are increased. This is implicit from the results shown in figure 3 for run-IDs 2 to 6. From 7^{th} simulation run onwards, neighbours play important role in increasing the query hits. This keeps on increasing; finally the major contributors are neighbours. The proposed scheme moves the neighbours in the dry areas to wet areas and thus due to replication they receive more amount of popular objects. Moreover, the amount of failed queries is significantly reduced. Because all categories of nodes are contributing to the increased success rate, query load among nodes are properly balanced.

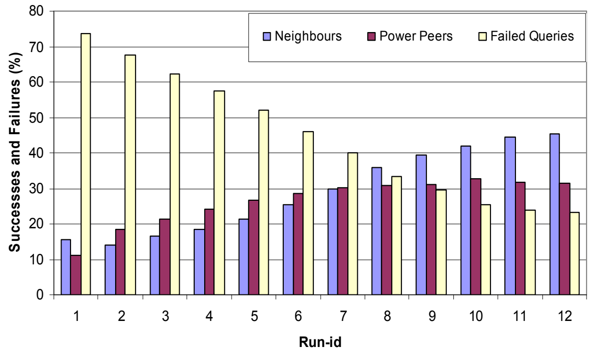

Figure 3. Success rate through different types of nodes

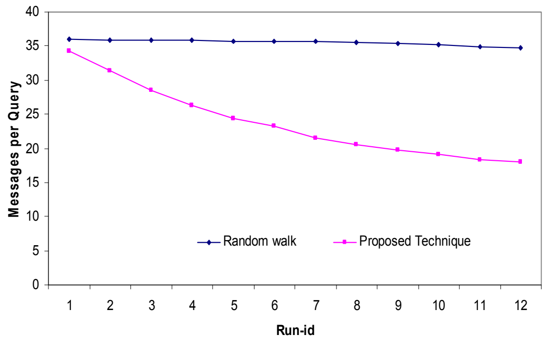

Figure 4. Messages per query

Messages per Query: To study how the proposed scheme affects the network traffic, the amount of generated for each query are monitored. Random walk produces about the same quantity of messages for each query being generated. However, the proposed scheme initially takes more messages per query and it gradually decreases as time increases. This is due to the high availability of popular objects in near by areas and large switching of dry areas into wet areas.

VII. Conclusions

In this paper, a bio-inspired resource discovery scheme for unstructured P2P network is proposed. The scheme utilises the migration policy of elephants in search of food during dry seasons. Each node in the network is classified as ordinary nodes and power peers. A node maintains a list of neighbouring nodes as well as power peers for routing queries. If enough resources are not available, the queries are propagated through the power peers located in the resource fertile area. At the same time, dry areas are filled with popular resources by means of an autonomous replication scheme. Nodes possessing certain features are only chosen for hosting the replicas. A variety of simulation experiments are conducted and the results show that the proposed technique significantly increases the query success rate and creates less network traffic.

References

[1] Mohamed Hefeeda, "Peer-to-Peer Systems", School of Computing Science, Simon Fraser University, Surrey, Canada, 2004.

[2] Stoica, Morris, R. Karger, D. Kaashoek, M and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *Proc. of SIGCOMM,* 2001.

[3] Pastry, Rowston and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," *Proc. of IFIP / ACM Middleware, Heidelberg, Germany,* 2001.

[4] Ben Y. Zhao, Ling Huang, Jeremy Stribling, Sean C. Rhea, Anthony D. Joseph, "Tapestry: A Resilient Global-Scale Overlay for Service Deployment," IEEE Journal on Selected Areas in Communications, Vol. 22, NO. 1, pp. 41-52, January 2004.

[5] S. Ratnasamy, P. Francis, M. Handley and R .Karp, "A scalable content-addressable network," *Proc. of SIGCOMM*, 2001.

[6] G.H. Page. http://www.gnutella.wego.com.

[7] K.H. Page. http://www.KaZaA.com.

[8] Limewire, http://www.limewire.com.

[9] Kavitha Ranganathan, Adriana Iamnitchi, and Ian Foster, "Improving Data Availability through Dynamic Model-Driven Replication in Large Peer-to-Peer Communities," *Proc. of 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid*, pp. 376- 376, May 2002.

[10] Kennedy, J. and Eberhart, R. C, "Particle swarm optimization," *Proc. of IEEE int'l conf. on neural networks,* vol. IV, pp. 1942-1948.

[11] Eberhart, R. C. and Kennedy. J, "A new optimizer using particle swarm theory," *Proc. of the Sixth int'l international symposium on micro machine and human science,* pp. 39-43, 1995.

[12] Yang Liu and Kevin M. Passino, "Swarm Intelligence: Literature Overview," Dept. of Electrical Engineering, The Ohio State University, March 30, 2000.

[13] Vana Kalogeraki, Dimitrios Gunopulos, and D. Zeinalipour-Yazti, "A local search mechanism for peer-to-peer networks," *Proc. of the Eleventh International Conference on Information and knowledge Management,* pp. 300 - 307, 2002.

[14] Xiuqi Li and Jie Wu, "Searching Techniques in Peer-to-Peer Networks," available: www.cse.fau.edu/~jie/research/publication/publication_file/P2Psearching.pdf.

[15] Lv. C, Cao. P, Cohen. E, Li. K and Shenker. S, "Search and replication in unstructured peer-to-peer networks," *Proc.16th Int. Conf. Supercomputing*, pp. 84-95, 2002.

[16] B. Yang, and H. Garcia-Molina, "Improving search in peer-to-peer networks", *Proc. of the 22nd IEEE International Conference on Distributed Computing (IEEE ICDCS'02),* 2002.

[17] D. Tsoumakos, and N. Roussopoulos, "Adaptive Probabilistic Search for Peer-to-Peer Networks," *Proc.3rd Int. Conf. P2P Computing*, pp. 102 – 109, 2003.

[18] Sabu M. Thampi, Chandra Sekaran K, "Q-learning based collaborative load balancing using distributed search for unstructured P2P networks," *Proc. of 33rd IEEE Conference on Local Computer Networks*, LCN 2008, 14-17, pp.797 – 802, 2008.

[19] Dong-Hong Zuo, Xu Du, Zong-Kai Yang, "Hybrid Search Algorithms for P2P Media Streaming Distribution in Ad Hoc Networks," Lecture Notes In Computer Science; vol. 4490, *Proc. of the 7th international conference on Computational Science*, part IV, pp. 873 – 876, 2007.

[20] Chi-Jen Wu; Kai-Hsiang Yang; Jan-Ming Ho, "AntSearch: An Ant Search Algorithm in Unstructured Peer-to-Peer Networks," *Proc. 11th IEEE Symposium on Computers and Communications*, pp. 429 – 434, June 2006.

[21] Ajith Abraham, Hongbo Liu, Youakim Badr, Crina Grosan, "A multi-swarm approach for neighbour selection in peer-to-peer networks," *Proc. of the 5th international conference on Soft computing as transdisciplinary science and technology*, pp. 178-184, 2008.

[22] K. Dhurandher, S. Misra, P. Pruthi, S. Singhal, S. Aggarwal and I. Woungang, "A Swarm Intelligence-based P2P file sharing protocol using Bee Algorithm," *Proc. of the 7th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-09)*, May 10-13, 2009.

[23] Sachin Kulkarni, Niloy Ganguly, Geoffrey Canright and Andreas Deutsch, "A new bio-inspired location search algorithm for peer to peer network based Internet telephony," *Proc. of the 1st international conference on Bio inspired models of network, information and computing systems*, Article No. 33, 2006.

[24] Niloy Ganguly, Geoff Canright, and Andreas Deutsch, "Design of an Efficient Search Algorithm for P2P Networks Using Concepts from Natural Immune Systems," *Parallel Problem Solving from Nature - PPSN VIII*, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Vol. 3242/2004, pp.491-500, 2004.

[25] Tathagata Das, Subrata Nandi, Andreas Deutsc and Niloy Ganguly, "Bio-inspired Search and Distributed Memory Formation on Power-law Networks," available: www.facweb.iitkgp.ernet.in/~niloy/PAPER/PPSN08.pdf.

[26] Elke Michlmayr, "Ant Algorithms for Search in Unstructured Peer-to-Peer Networks," *Proc. of the 22nd International Conference on Data Engineering Workshops*, page: 142, 2006.

[27] "Elephant Information Repository: Elephant Resources," elephant.elehost.com/, June 2009.

[28] "ELEPHANTS - Habitat & Distribution," www.seaworld.org/.../elephants/habitat-&-distribution.htm, June 2009.

[29] Raman Sukumar, "The Living Elephants Evolutionary Ecology, Behaviour, and Conservation," Oxford University Press, 2003.

[30] Sabu M. Thampi and Chandra Sekaran K, "Autonomous Data Replication Using Q-Learning for Unstructured P2P Networks," *Proc. of the sixth IEEE Int'l Symposium on Network Computing and Applications (IEEE NCA07)*, Cambridge, MA USA, ISBN: 0-7695-2922-4, pp. 311-317, July 12- 14, 2007.

[31] Sabu M. Thampi and Chandra Sekaran K, "Review of Replication Schemes for Unstructured P2P Networks", *Proc. of IEEE Int'l Advance Computing Conference IEEE (IACC'09),* Thapar University, Patiala, ISBN: 978-981-08-2465-5, pp. 794-800, 2009.

[32] www.netbeans.org (August 9, 2008).

[33] www.wampserver.com (August 9, 2008).