A new bio-inspired location search algorithm for peer to peer network based Internet telephony

Sachin Kulkarni*, Niloy Ganguly*, Geoffrey Canright‡ and Andreas Deutsch§
*Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur, India
Email: sachindkulkarni@gmail.com, niloy@cse.iitkgp.ernet.in
‡R&D Telenor, Oslo, Norway
Email: geoffrey.canright@telenor.com
§Center for Information Services and High Performance Computing (ZIH)
TU-Dresden, Dresden 01062
Email: andreas.deutsch@tu-dresden.de

Abstract— **In this paper, we propose 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 upon the concept of *gradient search* and is applicable upon unstructured networks. It is inspired by a biological phenomenon called haptotaxis. The algorithm performs at par with DHT-based location search algorithm, however is much more robust than such structured algorithm. We also present some initial analysis to explain the reason behind the superiority of the algorithm.**

I. INTRODUCTION

A p2p based system is a network that relies on the computing resources available from the participants in the network instead of concentrating on a few number of centralized servers. This kind of network allows to build a server-less infrastructure, where participating nodes cooperate to find the desired resources. Moreover, many p2p architectures offer inherent scalability and robustness as the network reorganizes itself in case of dynamic entry and removal of nodes. However there is a cost associated with it. For example, with respect to Internet telephony this is achieved at the cost of increased latency in locating the callee in the network. Optimizing this lookup time is one of the primary challenges of p2p Internet telephony systems.

Based on the type of neighborhood relationship between the nodes, p2p systems can be classified as unstructured and structured. Unstructured p2p networks are formed by linking nodes in the network at random. Location search in the unstructured network is generally done by flooding the query over all the neighbors. Although flooding quickly finds out the desired location, it is inefficient as it produces a huge number of message packets.

Structured p2p systems like CHORD and CAN, can perform location search much more efficiently through the maintenance of Distributed Hash Table. In DHTs, the identification information of nodes (keys) is distributed among the participating nodes themselves. DHTs also ensure that lookup for a desired user in the network can be done in $log(N)$ time, where N is the number of nodes in the network. However DHTs assume that underlying networks are structured, which is detrimental to the robustness of the network, which is built with unreliable nodes. Thus an algorithm, more efficient than, however as fast as, flooding needs to be designed for the location search in unstructured networks. In this paper, we propose such an algorithm.

In this paper, we propose a new algorithm based on a *gradient search*, which is a key-based algorithm. It performs a guided search for a desired key in the entire search space. *The algorithm is motivated by a biological phenomenon called haptotaxis, hence named "hapto-search"*. We have previously developed algorithms for search, which are inspired by the natural immune system and chemotaxis [1], [2]. Both haptotaxis and chemotaxis are used in interacting cell systems to regulate cell migration.

The rest of this paper discusses the background of Internet telephony, existing p2p telephony systems such as Skype (section 2), details of the proposed algorithm (section 3), performance evaluation of the algorithm (section 4) and finally the conclusion and future work is detailed in section 5.

II. BACKGROUND AND RELATED WORK

In this section, we discuss Skype, one of the most popular p2p telephony systems and SIP (Session Initiation Protocol), the most commonly used protocol in IP telephony.

A. Skype

Skype is a very popular p2p Internet telephony system, which is based on the super node architecture of Kazaa [8]. Skype builds an overlay network consisting of two types of nodes, ordinary host and super nodes. An ordinary node is a simple peer joining the Skype network. A super node acts as an ordinary node's end point on the Skype network. Every node maintains, in the cache, a list of reachable super nodes, whom it can contact for finding the location information of the callee. Skype also resolves the problem of communicating with nodes behind NAT (Network Address Translation) or a UDP (User Datagram Protocol) restricted firewall by using a variation of

STUN (Simple Traversal of User Datagram Protocol Through NATs) [7] and TURN (Traversal Using Relay NAT) protocols.

The problem with Skype is that it does not follow open source standards. There are various studies [4], [5], which analyze Skype architectural features such as the login function, call establishment, media transfer, NAT and firewall traversal.

For location search, Skype claims to use *global indexing technology*, which builds up a multi-tired network, where supernodes communicate with each other to find the location information of ordinary nodes.

In Internet telephony location search is part of the session initiation protocol (SIP) which is discussed next.

B. Role of SIP in IP telephony

In our protocol design, we intend to use p2p SIP. SIP (Session Initiation Protocol) is one of the most commonly used protocols for standard IP telephony systems. It is an application layer control protocol, that enables initiating, managing and terminating multimedia sessions (data, voice, video and audio) over the Internet. In a typical SIP-based telephony system, registration servers are employed per domain, where the users register their location information (IP address and listening port) at the time of joining the network. An aspirant caller contacts the proxy server for the location information of the callee. This proxy server may happen to know the required information or can contact the registration server of the callee's domain for it (using some non-SIP protocol). Once it knows the location details of the callee, it sends the session initiation request of the caller to the callee along with its session description information. Once the proper handshake is done, both the nodes can start the session.

SIP is based on traditional client-server architecture for Internet telphony. In order to build a p2p architecture using SIP, one has to replace the existing SIP registration and location search by some p2p protocol. There have been attempts to provide a SIP-based p2p architecture for IP telephony system. One such architecture [6] is a hybrid of structured p2p (DHT) and SIP server systems.

It this paper, we propose a location search algorithm on unstructured networks. The unstructured network will bring robustness to the system. Side by side, the search algorithm will reduce the lookup query time. The algorithm is discussed next.

III. HAPTO-SEARCH ALGORITHM

In this section, we discuss the hapto-search algorithm, its inspiration and details of how it works.

A. Design Goal and Basic Idea

Main design goal of this algorithm is to attain the speed of location search comparable to DHTs (i.e.$\log(N)$) but to make no assumption about the structure of the network. The reason against using DHTs is that they assume that the underlying networks are structured. Structured networks are more prone to failure and not as robust as unstructured networks.

The algorithm is a gradient-based search, in which every node is assigned a 32-bit key unique within our p2p network. A node distributes its key to a fixed number of peers in the network at random. When a peer wants to call another peer, it tries to search for a node, which has the key of the callee. Thus a query traverses in the network and the next hop is calculated in such a way that it ensures that the query always moves "closer" to the destination than the current position. This algorithm is inspired by a biological phenomenon called haptotaxis.

B. Our Inspiration - Haptotaxis [9], [10]

Migration of cells within our body, is a fundamental process in tissue development, inflammation, tumor metastasis and wound healing. To achieve appropriate physiological outcomes, cell movement must maintain a defined direction and speed in response to environmental stimuli. There are cell adhesion proteins at the outer surface of the cells, which bind to the adhesion ligands present in the extracellular matrix (ECM). This extracellular matrix is a substratum that surrounds the cells in a tissue. Cells tend to move in the direction where the adhesion between ECM ligands and cell receptors is more. It is believed that the magnitude of this adhesion influences cell speed and random turning behavior, whereas a gradient of adhesion affects the resultant direction of cell movement. This phenomenon of guided and gradient based movement of cells is called haptotaxis. Haptotaxis is regulated by the presence of the ECM consisting of immobilized molecules. Alternatively, cell migration towards a gradient of diffusible and soluble growth factors is known as chemotaxis. In this example, the ECM resembles a p2p network. Then, a query path in the p2p network resembles a haptotactic cell movement in the ECM.

Accordingly, the concept of haptotaxis gives an idea of guided search towards the destination so that one can always make sure that at every step, the moving entity is going "closer" to its destination.

C. Detailed algorithm

In this section, we introduce the detailed algorithm, which is a key-based algorithm. It allows nodes to retrieve the location information of each other. Each node is assigned a key, which is unique across the network. This key can be generated by hashing the userid into a 32-bit length code. Every node distributes its key to a fixed number of nodes in the network. This distribution is done at random. A node (say) A knows the location information of another node (say) B, if it has the key of B. Thus if one node wants to search the location information of the callee node, it tries to reach for the node, which has the key for this callee. Once it finds such a node, it can get the desired location information from it.

If a node doesn't have the key required, it routes the query to the neighbor, who is closest to the destination. A neighbor is said to be closer to the destination if it has a key which is closer to the destination key (in terms of Hamming distance) than the current node. The closest neighbor is selected to forward the query. This way the algorithm ensures that the query always moves "closer" to the destination as Hamming

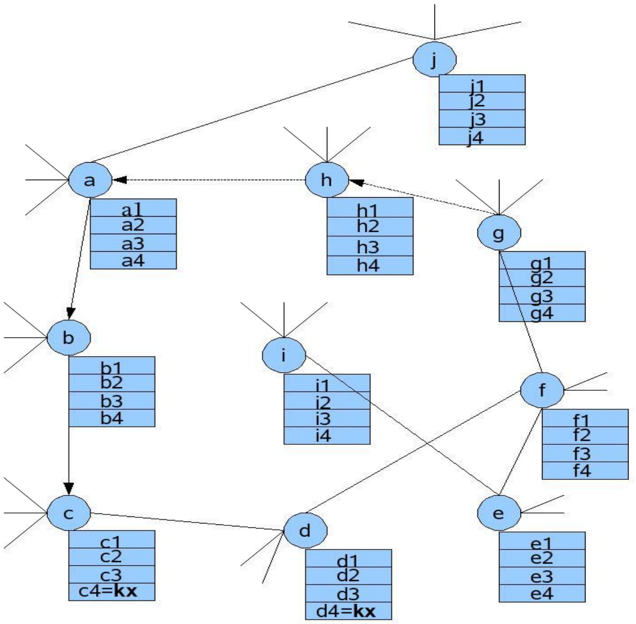

Fig. 1. Sketch of a network showing how the haptotaxis-inspired algorithm operates. Each node is hosting 4 keys received from random peers. This diagram illustrates a case where node g wants to call node x (Node x is not shown in the figure). Nodes c and d have the key of x. Node g forwards the query to a neighbor, which has a key closer to k_x (node h in this case). This way query traverses in the network and ultimately reaches node c. The path indicated by arrows shows a possible query traversal.

distance between destination key and the closest key present at the current node always decreases, if it is not the same. The search ends when the query reaches a node hosting the destination key (i.e. Hamming distance becomes zero).

It may happen that the current node itself is closer to the destination than all its neighbors. We can call this a *dead end* as there is no closer neighbor to forward the query. In such case, we continue the search by sending the query to the farthest neighbor, i.e. the node - Hamming distance of whose key from the destination key is most among all the neighbors of the current node.

Dead ends occur due to the presence of local maxima in the search space. Dead ends will slow down the search algorithm. However, occurrence of dead ends depends upon the number of times a key is replicated as well as the average network degree. The number of nodes, where the key of each node gets replicated, is to be decided based on various parameters such as average hops needed to reach the key, the number of occurrence of dead ends etc. Algorithm 1 shows the details of the hapto-search algorithm.

Figure 1 shows a part of a network, wherein each node is given 4 keys from random peers. Consider a case when the caller g tries to search the key k_x, which has the location information of x. It finds out which neighbor has a key ”closer” to k_x. Assuming node h has a key ”closer” to k_x, the caller forwards the query to h. This way the query traverses in the network and ultimately reaches node c, which holds the required key.

```
input  : Key of the destination node
output: Location information of destination node
current ←source ;
while current node doesn't have the destination key do
    for all the neighbors of current node in the network do
        shortest ← GetShortestHammingDistance (neighbor,destination);
        if neighbor's shortest distance < my shortest distance then
            Add the neighbor to the sorted list of prospects
        end
    end
    while List of prospects is not empty do
        node ← take a node from the sorted list with minimum distance;
        if node not already visited in the search process then
            current ← node;
            Break;
        end
    end
    if No such node available then
        This is the condition of dead end
        current ← choose the neighbor with the largest distance
    end
end
```

Algorithm 1: Searching a key in a peer to peer telephony network. The input to the algorithm is the key for the destination and the output is the location information of the destination. The central function GetShortestHammingDistance() which, given a current node and a destination node, finds out a key present at the current node, which has minimum Hamming distance to the destination key and returns this minimum distance.

IV. PERFORMANCE EVALUATION

This section presents the results of the performance evaluation of the hapto-search algorithm. However, we first define the metrics based upon which the evaluation is done.

A. *What to evaluate?*

We ran the algorithm on various networks (with different topologies and different average node degrees). For each network, and for different values of key replication (number of nodes, where a key is hosted), we ran the algorithm

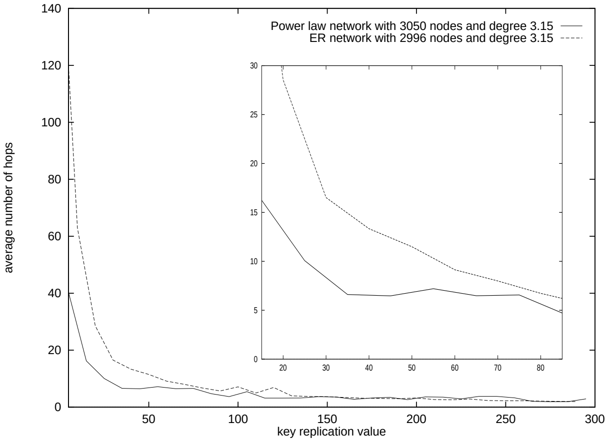

Fig. 2. Plot of avg number of hops Vs key replication for power law network and ER network. The enlarged figure shows an enlarged section of this plot for key replication value ranging between 15 and 85.

for randomly chosen 2000 source-destination pairs. We were mainly interested in the following aspects:

- whether the algorithm scales to $\log N$, where N is the number of nodes in the network (for matching performance of our algorithm with DHT architecture),
- comparing the performance of the algorithm with random search, where the next hop to forward the query is chosen randomly from the neighbors of the current node.
- We have mentioned before that dead ends essentially imply the presence of local maxima in the search space. Gradient search will become faster if the number of local maxima decreases. *Dead ends* are the result of connectivity, amount of key replication as well as the network topology. So we measure
  - at what value of average node degree and key replication, the number of dead ends in a search gets reduced to an insignificant value,
  - testing the performance on ER networks and power law networks,

B. Average hops

The most important performance criterion in a search is the average number of hops taken to transport the query to the appropriate destination. Figure 2 shows a plot of the average number of hops vs the key replication value for power law network and ER network.

As one can see from the figure, hapto-search algorithm performs better for power law network than ER network. However, there is a drastic fall in the average number of hops per search as the key replication value increases. It was observed that for the power law network, the average number of hops gets reduced to a small value 6.6 at key replication 35 whereas for ER networks, average number of hops can attain such a small value at key replication value 85.

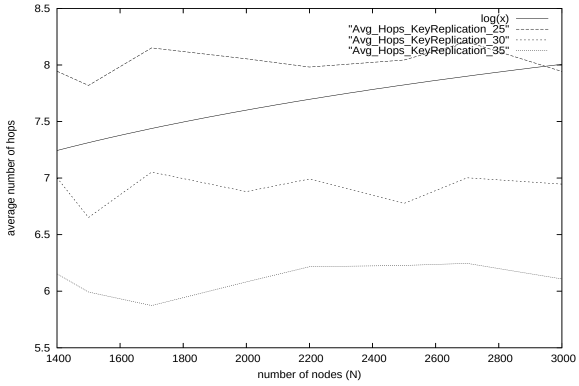

Fig. 3. Plot of avg number of hops for different key replication values (25,30 and 35) Vs number of nodes in ER networks. These plots are compared with a log(N) curve. One can see that the plot for key replication value 25 is just above the log(N) curve and the plot for key replication value 30 is just below.

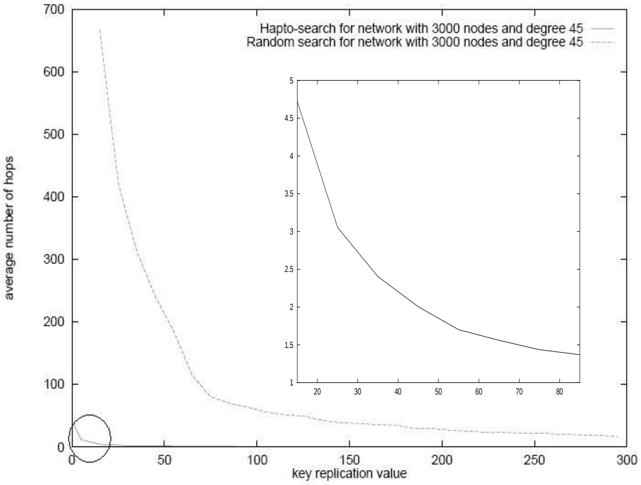

Fig. 4. Comparative plot of average hops Vs key replication for random search and hapto-search algorithm. The encircled portion shows the performance of the hapto-search algorithm. The enlarged figure shows exclusively the plot of the performance of hapto-search algorithm for the key replication value ranging between 15 and 85.

Figure 3 shows the plot of average hops Vs number of nodes in the ER network for different values of key replications. It compares the curve of average hops with the $\log(N)$ curve. It can be seen from the plot that for key replication value 25, the curve of average hops needed is just above $\log(N)$ curve and for key replication value 30, the same curve is below $\log(N)$. So we can say that the hapto-search algorithm can attain a speed compared to $\log(N)$ if the key replication value is set between 25 and 30.

C. Comparison with random search

We also compared the hapto-search algorithm to the random search algorithm, where the next hop is chosen randomly from neighbors. Figure 4 shows the comparative plot of average hops needed for random search and hapto-search algorithm. The enlarged plot shows the performance of hapto-search algorithm on the same network for key replication value ranging between 15 and 85. One can see that the hapto-search algorithm performs far better than random search. The average number of hops taken by the random search algorithm at key replication value 295 is 16.8 whereas for the hapto-search algorithm and for the same network, even at the key replication value of 5, the number of average hops needed is 11.8. This shows that the gradient-based guided search is far more efficient than the random search.

D. Dead end calculation

A search reaches the dead end when none of the keys stored at neighbors is closer to the destination than keys stored at the current node. Dead ends force to restart the search process by sending the query to the farthest node. So dead ends in the search should be reduced to an insignificant value. They can be reduced by the optimum selection of the average degree of the network and the key replication value. Figure 5 shows the plot of the total number of dead ends encountered in 2000 searches vs the key replication value for ER network and power law network.

As one can see in Figure 5, the number of dead ends is reduced drastically when the key replication value is increased. It was observed that for ER networks with average degree 3.15, the number of dead ends reduces to 279 (in 2000 searches) for a key replication value of 280. As the network degree increases, this key replication value, where dead ends are reduced to a small value, reduces further. However it is observed that even for very high key replication value, the number of dead ends never becomes zero.

However, we can conclude that the new, innovative metrics of *dead end* defined by us directly maps with the hop-count results shown in Figure 2. A more detailed and rigorous relation (hop count and dead end) definition is part of the future work.

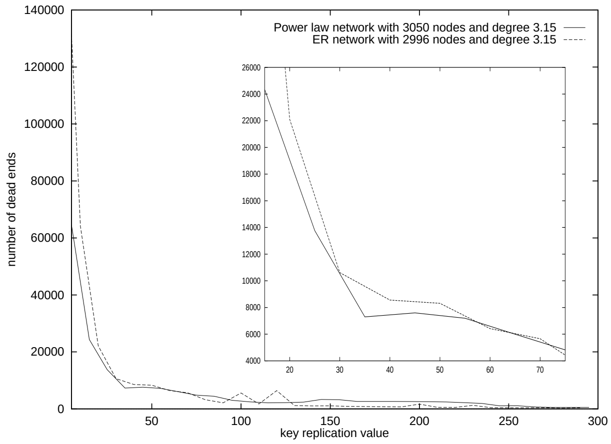

Fig. 5. Plot of number of dead ends Vs key replication value for ER network and power law network. The enlarged figure shows an enlarged section of this plot with key replication value ranging between 15 and 85.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a bio-inspired algorithm for the location search in a peer to peer Internet telephony system. It is a key-based search algorithm, which ensures that the query always moves ”closer” and ultimately reaches the destination. We proved that the algorithm is far more efficient than the random search and at suitable key replication values and average network degree, it performs as efficient as DHT-based search.

However, these promising results need to be more thoroughly tested. The algorithm’s efficiency needs to be tested on the complete test bed, which we will develop in the recent future. Also the algorithm needs to be tested in face of the dynamics of the p2p network. Side by side refinement of the algorithm also needs to be undertaken. Ideas of refinement can be derived from studying the haptotaxis process in further details. For example, now the keys are distributed at random among the peers. We plan to derive ideas from haptotaxis to make this key distribution non-random in order to improve the performance of search algorithm.

REFERENCES

[1] N. Ganguly, G. Canright, and A. Deutsch *Design of an Efficient Search Algorithm for p2p Networks Using the Concepts from Natural Immune Systems*, In proceedings of PPSN VIII: The 8th International Conference on Parallel Problem Solving from Nature, 18-22 September 2004, Birmingham, UK.

[2] G. Canright, A. Deutsch, and T. Urnes *Chemotaxis-Inspired Load Balancing*, In the proceedings of the European Conference on Complex Systems, November 2005.

[3] N. Ganguly, and A. Deutsch *Developing Efficient Search Algorithms for p2p Networks Using Proliferation and Mutation*, In the proceedings of International Conference on Artificial Immune Systems, 13-16 September 2004, Catania, Italy.

[4] T. Urnes *An Analysis of the Skype Peer-to-Peer Protocol* Internal publication, Telenor R&D, May 2006.

[5] S. A. Baset, H. Schulzrinne *An Analysis of the Skype Peer-to-Peer Internet Telephony*, Internal Report, September 15,2004.

[6] K. Singh, H. Schulzrinne *Peer-to-Peer Internet Telephony Using SIP*, In the proceedings of the international workshop on Network and operating systems support for digital audio and video, Washington, USA, 2005, 63-68.

[7] J. Rosenberg, J. Weinberger, C. Huitema and R. Mahy. *STUN: Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)*, RFC 3489, IETF, Mar. 2003.

[8] *http://www.kazaa.com*

[9] Dickinson RB, Tranquillo RT *A Stochastic Model for Adhesion-mediated Cell Random Motility and Haptotaxis*, J. Math. Biol. 1993;31(6):563-600.

[10] A. Perumpanani, D. Simmons, A. Gearing, K. Miller, G. Ward, J. Norbury, M. Schneemann, J. Sherratt *Extracellular Matrix-mediated Chemotaxis Can Impede Cell Migration*, Proc. R. Lond. B (1998) 265, 2347-2352