Replication with State Using the Self-Organizing Map Neural Network

Geofrey Cantara Soriano[a], Yoshiyori Urano[a]

[a] Graduate School of Global Information and Telecommunication Studies, Waseda University, 1011 Nishitomida, Honjo-shi, Saitama 367-0035, Japan

geofreysoriano@suou.waseda.jp, urano@waseda.jp

Abstract— **Most architecture of mobile ad hoc network is in the form of decentralized, self-configuring and dynamic topologies. Nodes are mobile in network. The mobility of node in network is common problem in peer-to-peer technology. Object replication is one of the techniques applied in order to share objects in the mobile peer-to-peer environment. Predicting the estimated time for the node to exit is a great opportunity to improve the efficiency of search algorithm. Even if the object owner had departed from network, the shared objects are still available at all times. The goal of the technique is to maintain a number of object replicas over the time before a node exits in decentralized and unstructured environment. That is the reason why there is a need to replicate objects based on the predicted condition of nodes that are about to depart from the network, which is necessary.**

This paper proposes a modified form of random replication of data within a mobile peer-to-peer network based on predicting condition for a mobile node to replicate the object from it. It uses the unsupervised learning neural networks called the Self-Organizing Map by classifying the input attributes of each node and providing a training set - serving as a basis of identifying the nodes' current state.

Existing algorithm shows significant results such as reducing data traffic, load balancing, and decrease query latency. The preliminary results of the proposed scheme had level into the existing algorithm because of its node mobility prediction condition capability as the significant feature to replication. To test the functionality of the technique, a simulation was developed in a multi-agent based modelling environment called the NetLogo and observations are compared with the proposed scheme with the existing algorithm.

Keywords— **Mobile Ad hoc Networks, Self-Organizing Map, Neural Networks, Replication, Peer-to-Peer**

I. INTRODUCTION

The peer-to-peer (P2P) network and mobile ad hoc network (MANET) had acquired popularity in the field of research. Both systems have similar objective, and that is to establish communication between each node. In addition, the P2P and MANET are based on a form of decentralized, self-configuring and dynamic topology, as point out in [1], [4]; therefore, there is no need for central administration in the network.

The peer-to-peer system aims to share resources and information among nodes in the large number of users connected over the internet. Recently, P2P has emerged as a general structure for distributed services and applications, however, composes of deceitful and undependable nodes with proportionate tasks in the distributed systems, as mention in [1]. Further on, the increase number of researches in file sharing has conceived and popularized by applications such as the Gnutella [2] and Freenet [3]. Gnutella peers perform as both the client and the server that send its query to all its neighbour nodes [2]. Furthermore, the kind of search algorithm causes problem of flooding the network and so will take much bandwidth although reduces search latency but it increases loads on predicament nodes, as mention in [6], [7], [8].

In connection with the rapid progress in wireless communications, MANET has boosts its attraction in research field which characterized as the self-organizing mobile wireless networks [1], [4]; a collaboration of mobile nodes in which communication may not be within the direct connection range with each other and works autonomously. Moreover, MANET uses applications that should be designed to work in decentralized manner (e.g. [1], [4], [13]) and such collaboration are sharing of files and information, resource discovery, multitasking, etc. But as mention in [1], [4]; MANET is restricted in terms of node transmission range in which typically small as compared to P2P network that is extremely large.

Some of information related to the similarities and differences feature between the P2P file sharing and MANET are presented in [4], [13]. The main resemblance of the two networks that this paper focuses with, is in terms of topology aspects, which is the frequent changes in nodes' mobility. The on-off state of peers causes the unavailability of objects in the network. In systems, search flooding or broadcasting (e.g. [7], [14]) are used as means of exchanging data in file sharing applications among nodes that reduces search latency but causes traffic overloads. There are many means of reducing search latency. One method is replicating objects (e.g. [6], [7], [8]) into several numbers of hosts. In this way, it improves the performance of the system by minimizing the number of nodes to be explored before a certain query is determined.

The purpose of this paper is to establish a new method of replicating files in the network based on the predicted condition of mobile nodes by using the Self-Organizing Map

(SOM) [5] in classifying the input conditions of each node and procures a choice to replicate queried object. Unlike other replication algorithms, the proposed scheme uses nodes attributes to predetermine the availability of the node to accept replicated objects.

The organization of this paper is as follows: in section 2, we introduce the existing owner replication and path replication as presented by other P2P file sharing applications, in section 3 we present the problem overview for this paper, section 4 introduces our own algorithm using the SOM to identify the condition of a node to provide decision in replicating objects on a specific node. Section 5 presents our simulation setup and results, while section 6 presents our conclusions and future work.

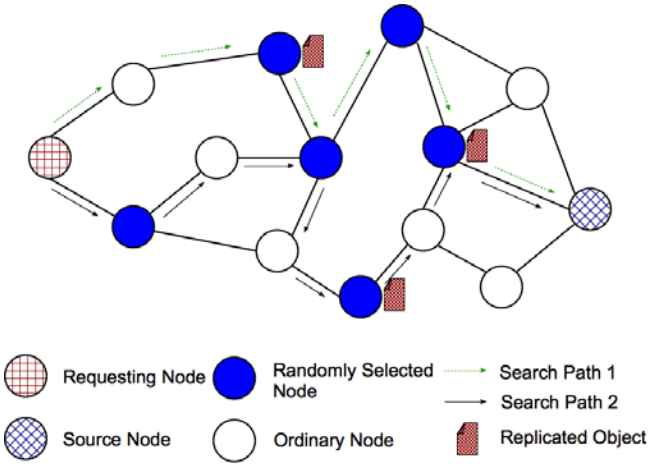

Figure 1. Illustration of Random Replication Algorithm.

II. Related Work

In this section, we review the related work on replication algorithms in P2P network. We observe some of the mechanisms in terms of placement strategies that are classified in terms of reactive and proactive replications, as in [6].

In reactive replication [6], the requesting node has the only copy of the object from the source node. This is likely termed as the "owner" replication as described in [7]. After a search success, a copy is generated into the node where the query was originated. A Gnutella system implements this kind of strategy [7]. This is primitive type of replication where search overhead is still higher. In the contrary, it gained good results in terms of minimize storage consumption for replicas.

In proactive replication [6], the object is replicated in selective nodes from the requester node to the provider node. Sometimes it is copied along the path of the successful query; it is termed as the "path replication", as in [7], [8]. Object is copied along the path from the requester node to the provider node after a succeeded search. This kind of strategy is implemented in the Freenet system [7].

Both the owner and path replications are easier to implement in the P2P networks and in [7] introduces a third type of replication algorithm known as "random replication". In random replication, the object copies are randomly placed in selective nodes along the path from the requester to the provider node of among the search walkers after the search succeeds, as in Figure 1.

In terms of allocation strategies, two types are presented in [8]. One is the Uniform replication, where objects are replicated in equal number. On the other hand, in the Proportional strategy, more popular items are easier to find since they are usually replicated. Therefore, better performance is then obtained. Better result is achieved if Square-Root replication is used, as described in [7], [8]. However, implementation for this type of replication is difficult, as indicated in [8]. Nevertheless, the path and random replication has shown close result to square-root replication (e.g. [7], [8]).

Most of the presented algorithm above are implemented in the fixed network and doesn't consider the mobility of each node (e.g. nodes leave due to link disconnection). Such limitations are being considered in this paper.

III. Problem Overview

The file-sharing system is one concept of distributing a copy of objects to another node. Each node has a collection of objects to be shared with other nodes in the network. Retrieving the location of the data is done using any of the different kinds of search strategies. A query for the certain object is initiated to perform the search among nodes. Each node is maintaining open links to each other. These links are used to forward a query message to another node until a search succeeded. The queries were processed on each node over the local collection of shared objects. The set of nodes of the message pass through until the requested file is resolved, known as the *path* – where its length is the "number of hops". The shorter the path length, the better is the performance in searching the requested object.

One way to achieve a good performance in a search on a distributed system is using the replication method. Copies of objects are placed on different locations or nodes in the network, which minimizes the search size. Reference [8] shows the effectiveness of other replication algorithm in a fixed network. On the contrary, the mobility of the node wasn't considered. In addition, the existing algorithms in [7] and [8] do not concern about the totality of replicas present in the network (where average search size is minimized on a network with a larger number of replicas). Thus, the data storage consumption is greater on those existing algorithms.

Mobility of nodes is important feature in search and replication of objects, since we can determine if the node is about to depart the network. That study comes in our proposed scheme, the ability to predict if the nodes are departing the network based on its current state. Moreover, the capability to minimize the storage space utilization without degrading the performance of the search method is been considered.

IV. Proposed Algorithm

The Random Replication with State Prediction Algorithm (RRSP) works on the principle of detecting the condition of the nodes that are moving out from the network. Hence, in the algorithm, it randomly copied the shared objects after a

successful query based on random replication scheme. In random replication, when the query node requests for an object and once it is located; the replication strategy is on the go by randomly selecting nodes along the search paths, as illustrated in Figure 1. The proposed algorithm uses the nodes' condition as the basis for decision to replicate objects into the nodes. The decision making for each node is manoeuvre by gathering the input condition and feed in into the SOM Neural Network technique. The nodes variable conditions are discussed, and presented as to how they are used in the development of the scheme.

A. Self-Organizing Map (SOM)

The SOM is a neural network type introduced by Teuvo Kohonen. It is used to classify input data into groups. The data is trained using the unsupervised learning where number inputs and outputs are specified, SOM is further explained in [5].

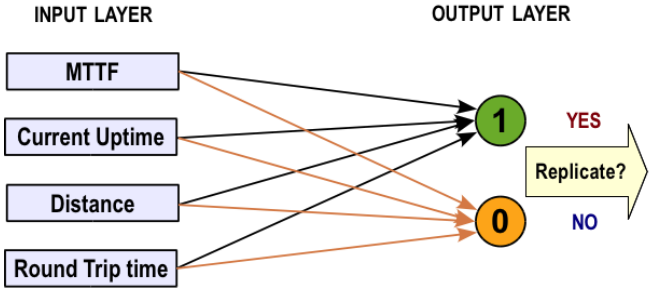

Figure 2. Processing of SOM technique in decision making for object replication

In our algorithm, the SOM usage is to classify the input state of each node whether it is going out from the network or not. Thus, a decision on replicating objects into nodes is based on SOM analysis on specific inputs of a particular node. Figure 2 illustrates our algorithm for SOM processing of inputs which are situated inside each node. Consequently, it is a 4-dimensional input vectors representing the nodes attributes. The nodes' inputs are mean time to failure (MTTF), current uptime, distance and round trip time (RTT), which are further explained on the later part of this section. A set of training samples, as patterns, is fed into SOM learning algorithm as the basis for each node's input variables to gain a group or classified category. Only two categories as output of SOM are set to simplify decision-making on a node, and so, these are either "0" or "1". The number "0" is for nodes about to leave and "1" for those who have enough time to stay in the network. Figure 3 shows the diagram on how the proposed scheme works with nodes checking its state using the SOM process.

When the search for data succeeds, the random replication is on progress. It checks the category of each node along the path by using the SOM algorithm. Once a nodes' category is specified, the copy of the requested object is placed for those nodes whose category is set to "1". On the other hand, those possesses the "0" category the replication scheme ignores the replication of object on that node, as illustrated in Figure 2, and proceed for the next node to be evaluated. The method keeps on going until the requesting node in the random replication is reached.

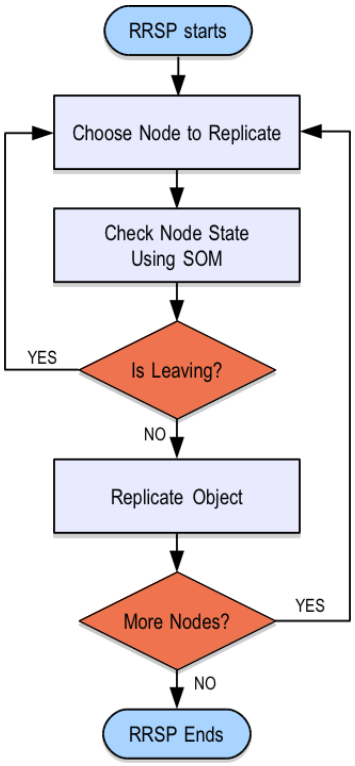

Figure 3. Diagram for the proposed scheme with SOM identifying the condition of nodes

B. Mean Time to Failure (MTTF)

This RRSP uses the information of each nodes' session time. To get the MTTF, each node remembers both of the last join time and the last leave time and computes MTTF and then contribute as one of the variables in predicting the nodes condition during the SOM analysis. Each node stores its MTTF to be used for future references.

C. Current Uptime

Nodes' current uptime, as one of the input attributes in SOM process, is also considered in evaluating the predicted condition. It measures the total time the node currently online. The current uptime is compared to MTTF if the node had reached the current MTTF and it is a relative input in the SOM process.

D. Distance

In a wireless networks, each mobile nodes called initial nodes send a ping message such as "hello" to its neighbouring nodes and be able to estimate the distance, as well as, establish communication between them. The distance is determined by using the Friis transmission equation between two points in the radio link, which is also presented in [9]. The given equation is:

$$P_r = P_t G_r G_t \lambda^2 / (4\pi d)^2$$

where P_r and P_t are the received and transmitted power, respectively. The G_r and G_t are antenna gain of the receiver and transmitter, correspondingly, λ is the wavelength and d is the distance. This equation is applicable for wireless communication, although we didn't implement in our simulation, we assume this equation is the same way of attaining the distance between two nodes in MANET.

In our simulation, the distance was obtained between nodes by getting the link length as provided by the simulator from the initial node to the endpoint of the link or to the neighbour node.

E. Round Trip Time (RTT)

The RTT is the return time measured by sending a packet from the local node to the remote node. Hence, this paper is using a simulation to determine the performance of the proposed algorithm. The formula below is used for the estimation of RTT in wireless LAN, which is also presented in [10].

$$RTT = \frac{2 \times distance}{c} \quad where\ c \approx 3 \times 10^8 \frac{m}{s}$$

The *c* here is the speed of light, while the distance is the length of local node to the remote node. The average RTT among the links or neighbours of a certain node is feed into the SOM as one of its inputs.

V. SIMULATION OVERVIEW

In this section we discussed the simulation settings and results, in order to evaluate the performance of our proposed algorithm within the MANET, which is implemented in an unstructured topology.

A. The Simulator

The algorithm simulations were performed in the NetLogo [11] simulation environment. It is a multi-agent programmable environment, which allows the modelling of simple development of complex mobile agent systems, as stated in [11].

To use the SOM technique for the assessment of the node condition detection, we use Java libraries from Encog [12]. Encog is an Artificial Intelligence framework containing a multiple variety of neural network programs and codes, and machine learning context. It supports the classification of neural network trained input samples as it is related to pattern recognition, as in [12]. In our proposed algorithm, there is a set of training data together with the specific nodes current values and feed into the SOM strategy to classify into groups to determine the nodes current values where it fit in.

The integration of Encog programming codes into NetLogo is through NetLogo's extensions facility. The extension facility was used to widen our program by writing new commands and reporters in Java and include the Encog libraries for SOM technique and utilize into the simulator as if they were built-in instructions. The custom commands or reporters we developed were called from the simulation to obtain a prediction for the condition of a node and these were used in our proposed scheme.

B. Simulation Setup

The experimental network can accommodate a maximum of 100 nodes. Every node establishes a link upon joining to a maximum of 5 neighbours within the network. A unique id is assigned to each node.

The exponential distribution is used in the first joining time of node, while a weibull distribution is assigned to reinstate and leave duration of the node. In addition, each node has the capacity to store 10-replicated objects. The join and leave time are meticulously monitored for each node as usage variable for calculating MTTF, while updating the uptime frequently every 5 seconds.

The nodes' distance is checked frequently to verify its current gap to neighbouring nodes. A specified limit is set for all, whereas, a link is deleted if limit is reached. A drifting node, as in [9], is detached from the network once all links were removed.

To evaluate the performance of our proposed algorithm, another replication strategy (e.g. random replication without added features) was simulated. A search query generator follows the weibull distribution and starts on a randomly selected active node in the network. The data queried is also randomly selected in all the shared object of all known nodes. For search method, we deploy 3-random walkers with state checking in every step in this small network. Thus, we do not concern ourselves in randomly selection for the list of nodes nor its total size where we have to execute the object replication process.

C. Preliminary Results

In preparation for strenuous observation, we allow the simulation to run for enough "tuning up" process for 20,000 seconds and observed and took the records until it reached 40,000 seconds of operation.

TABLE 1. ACCURACY PERCENTAGE OF NODES PREDICTION STATE

	Specified Limit for Distance			
	12	**16**	**20**	**22**
Total Samples	3495	3202	3234	3603
Correctly Predicted (%)	53.91%	55.15%	52.41%	51.85%

For our proposed replication scheme, we examined:

- *Accuracy percentage of node's current state:* The accuracy of SOM process to predict the current status of each node whether it is about to leave the network.
- *Relevant relation of nodes' session time and distance to SOM process*: The influence of each node's uptime and average distance to its neighbours to the SOM technique in processing the prediction for the nodes' availability.
- *Queries finished per distribution of hops*: The cumulative distribution of number of hops by queries finished under each replication algorithm.
- *Allocation of replica objects in varying number of nodes*: The number of replica (counts the original one and its copies) distributed into a changing size of available nodes.

Mobility of nodes comes with a tough method in predicting its departure from the network. The accuracy of detecting the nodes condition to depart from the network with several specified limit in the average distance between the node and its neighbours is shown in Table 1. In the given number of samples, the result shows that above half of samples were predicted correctly whether the nodes are about to go in on/off state. The first attempt towards the use of SOM in detecting nodes' mobility in the network is not impressive. However, it is a good start in determining the nodes' condition as a primary thing to consider in an unstable network with varying nodes' availability.

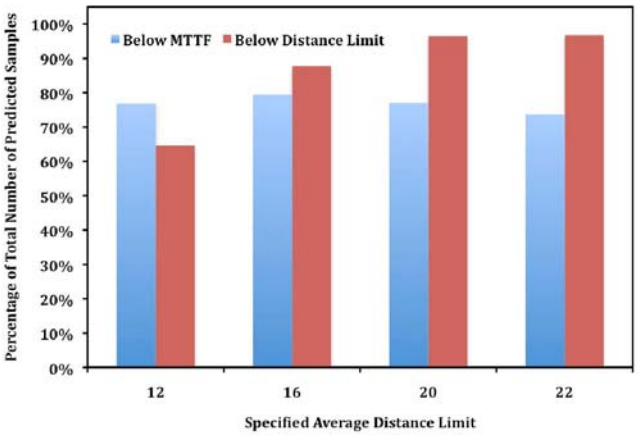

Figure 4. Interconnection of total nodes' session time and average distance in the SOM technique prediction for a specified distance limit.

To show how SOM input attributes influence the prediction ability in our proposed algorithm, the Figure 4 illustrates the percentage of samples in relation with MTTF to nodes' uptime and distance against the specified limit. The MTTF value indicates as the basis to determine the length of uptime each node spent. As the figure indicates, most of the nodes' uptimes were spent below the generated MTTF for each SOM prediction activity and little likelihood spent beyond the MTTF value.

Moreover, the distance has great effect to the nodes prediction state since in the simulation a cut-off limit length is set for the gap or distance of the two nodes. On contrary, there are some links which remained intact. This is due to the time that the next verification of gap between newly reinstated neighbouring nodes that have not yet instigated. The result shows that above 85% of generated average distance is below the range of the specified maximum length. The analysis on both nodes' session time and distance could significantly influence the way that the SOM prediction analyse the nodes' availability.

Reducing traffic comes from reducing the number of hops it takes for the query to find the object. In addition, it minimizes the search latency. Figure 5 shows the cumulative distribution of hops for all queries under the two-replication algorithm presented in this paper. In here, the samples used are those in a specified distance limit of 20, since it is nearer to the average of all samples as illustrated in Table 1. Furthermore, Figure 6 uses the samples generated in distance limit of 20 that make use of 50 objects.

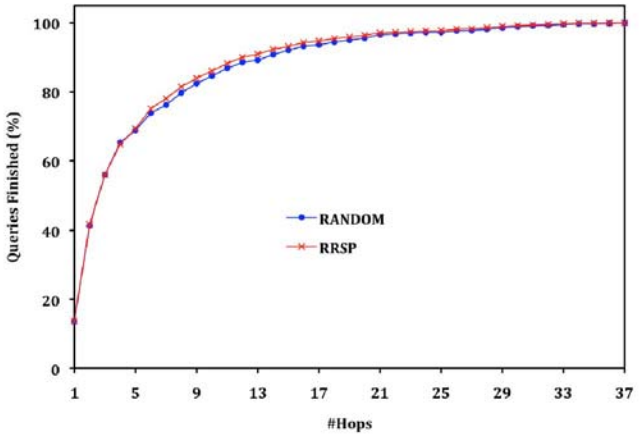

Figure 5. The queries finished in percentage of the two-replication algorithm versus the cumulative distribution of hops

The Random Replication with State Prediction is levelling with the existing random replication without state prediction strategy with a few percentages starting from the first to fourth hops. Nevertheless, the percentage of queries finished increases starting at the fifth hops for RRSP against the original random replication. Even though, there were lesser replicas present in the network as shown in Figure 6, our proposed scheme had maintained its performance against the existing random replication.

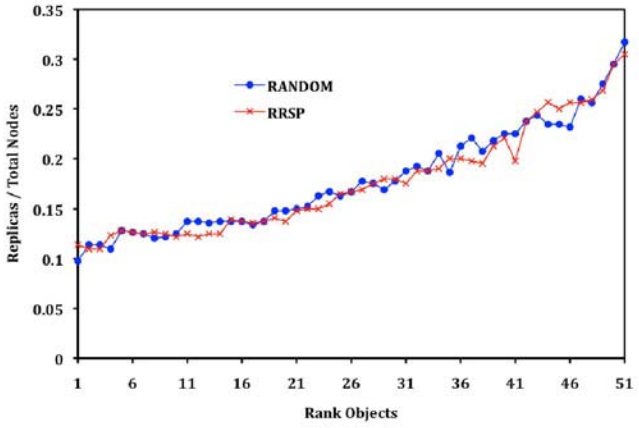

Figure 6. Average number of replicas per total number of nodes for every queried objects

In replication allocation the objective is to minimize the average search size by adjusting the number of replica for the object in the network. Thus, in a topology with more replicas, the lower the search latency is the result. However, it costs greater consumption in storage space. The average number of replicas per varying network size for the two-replication strategy is shown in Figure 6. Its shows the variations of sizes in the totality of replicas for 50 objects rank according to

volume. The RRSP consumes less storage space than the random replication in a network with dynamic number of nodes.

VI. Conclusions and Future Work

This paper has proposed a method of replicating objects in an unstructured, self-configuring and dynamic network, which follows the random replication with an additional feature of nodes state detection.

We have focused in the mobility of nodes, which were not considered in existing replication algorithms. It is the foremost feature of this proposed strategy. Determining the exact mobility of each node is a tough job to acquire. Thus, we have focused on predicting nodes state using the SOM technique to classify into category the conditions of individual nodes. In addition, we concerned ourselves in the totality of replicas present in a dynamic network that levels into the performance of object searching method.

We also presented the preliminary results of our proposed scheme of about 52% to 55% of accuracy in predicting the nodes state in leaving the network when varying the limit distance. Although, the percentage is not that high but it is a good start for the author to continue the study. This shows that our technique has the potential to increase its performance in the future since it is an important stage for our next step in improving the prediction for the availability of nodes.

Moreover, additional initial result shows that the proposed scheme is levelling to the existing replication algorithm in terms of number of queries finished per hops. Consequently, our strategy had minimized the replicas existed in a topology with varying number of nodes. As a result, the proposed algorithm had lessened the storage space utilization.

This study is our first step towards detecting accurately the nodes mobility and replicate objects of the leaving nodes to its neighbours. Therefore, more enhancement of this study is roughly needed. It will be useful to add some nodes features such as the nodes availability patterns as input to SOM technique, furthermore, grouping the inputs into particular combination to foster improvement in the prediction capability of the scheme. The groups are the MTTF and the current uptime as one group together with the RTT and the distance as another group. These groups were fed in into the SOM process for predicting nodes' state. Also, for the replication scheme, we could further exploit its potential by gathering the load-balance characteristics from the topology of unfixed number of nodes.

References

[1] Y. C. Hu, S. M. Das, and H. Pucha, "Peer-to-Peer Overlay Abstractions in MANETs," in *Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks*, edited by J. Wu, CRC Press, 2004.

[2] (2003) Gnutella: file sharing and distribution network. [Online]. Available: http://rfc-gnutella.sourceforge.net/.

[3] (2010) Freenet: The Free Network. [Online]. Available: http://freenetproject.org/index.html.

[4] R. Shollmeier, I. Gruber and M. Finkenzeller, "Routing in Mobile Ad Hoc Networks and Peer-to-Peer Networks, a Comparison," *Inter. Workshop on Peer-to-Peer Computing*, May 2002, vol. 2376, pp. 172-187, May2002.

[5] T. Kohonen, "The self-organizing map," in *Proceedings of the IEEE*, 1990, volume 78, pp. 1464-1480, September 1990.

[6] J. Zhou, L.N. Bhuyan and A. Banerjee, "An effective pointer replication algorithm in P2P networks," in *IEEE International Symposium on Parallel and Distributed Processing*, 2008, ISSN 1530-2075, pp. 1-11, April 2008.

[7] Q. Lv, P. Cao, E. Cohen, K. Li and S. Shenker, "Search and Replication in Unstructured Peer-to-Peer Networks," in *Proceedings of ICS '02*, ACM 1-58113-483-5, pp. 82-95, June 2002.

[8] E. Cohen and S. Shenker, "Replication Strategies in Unstructured Peer-to-Peer Networks," in *Proceedings of SIGCOMM '02*, ACM 1-58113-570-X/02/2008, August 2002.

[9] M. Pushpalatha, R. Venkataraman, R. Khemka and T. Rama Rao, "Fault tolerant and dynamic file sharing ability in mobile ad hoc networks," in *Proceeding of International Conference on Advances in Computing, Communication and Control '09*, pp. 474-478, January 2009.

[10] A. Günther and C. Hoene, "Measuring Round Trip Times to Determine the Distance Between WLAN Nodes," in *Proceedings of International IFIP-TC6 Networking Conference '05*, LNCS 3462, pp. 768-779, May 2005.

[11] (2010) U. Wilensky, NetLogo. [Online]. Available: http://ccl.northwestern.edu/netlogo/

[12] (2010) J. Heaton, Heaton Reasearch: Encog Java and DotNet Neural Network Framework. [Online]. Available: http://www.heatonresearch.com/encog

[13] G. Ding and B. Bhargava, "Peer-to-peer File-sharing over mobile ad hoc networks," in *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004*, pp. 104-108, March 2004.

[14] B. Yang and H. Gracia-Molina, "Improving Search in Peer-to-Peer Networks," in *Proceedings of 22nd IEEE International Conference on Distributed Computing Systems, 2002*, July 2005.