

An Adaptive Algorithm for Managing Gradient Topology in Peer-to-Peer networks

Sara Fathipour Deiman

Faculty of Computer and Information Technology
Engineering, Sama technical and vocational training college
TehranBranch (Tehran), Islamic Azad University, Tehran,
Iran
S.Fathipour.IT@gmail.com

Ali Mohammad Saghir, Mohammad Reza Meybodi

Soft Computing Laboratory, Computer Engineering and
Information Technology Department, Amirkabir University
of Technology (Tehran Polytechnic), 424 Hafez Ave.,
Tehran, Iran
a_m_saghiri@aut.ac.ir, mmeybodi@aut.ac.ir

Abstract—Super-peer network is a type of peer-to-peer networks. In a super-peer network, a super-peer is a peer that has more ability than other peers have and is responsible for some of the tasks of network management. Since different peers vary in terms of capability, selecting a super-peer is a challenge problem. Gradient topology is a type of super-peer networks. Because of dynamicity of peers, adaptive methods are important for managing gradient topology. A problem of the existing management algorithms of gradient topology is that they are not sensitive to joining and leaving the peers. This problem becomes more challenging when a malicious peer frequently joins and leaves the network. The proposed algorithm being sensitive to removal of super peers, using learning automata, selects the new super-peers in an adaptive manner. According to the simulation results, the proposed algorithm can compete with the existing algorithms.

Keywords-component; Super-peer selecting; Learning automata; Gradient topology; malicious peer.

I. INTRODUCTION

In peer-to-peer networks, all peers have equal responsibilities. This means that, each peer acts as a client and also as a server at the same time [1]. Super-peer network is a type of the peer-to-peer networks. In a super-peer based network, a super-peer is a peer that has more ability than other peers have and is responsible for some of the tasks of network management [2]-[3]. Using super peer based networks has many advantages such as Scalability, load balancing, and also improving performance of the network. Therefore, super peer selection leads to a challenging problem [4]- [6]. There are four ways to select the super peer, such as selecting the super peers in a simple method [7-9], selecting the super peers based on group [5, 10-12], selecting the super peers based on Distributed Hash Table [13-15], and adaptive selection of super peers [16-25]. Many researches have been done on the selection of super peer. However, because of the dynamism of these networks' adaptive methods are more appropriate than non-adaptive methods. In the comparative method of selecting super peers, has been steadily tried to select appropriate super peers. In this method, selection of super peers is done according to criteria of peers' utility such as session time, storage space, processing power, bandwidth, and capacity of peer, and workload [20].

The criteria of becoming a super peer in [16] are the capacity of the peer, in a way that the ratio of super peers comparing the clients remains constant. The appropriate number of super peer is calculated based on the estimation of network characteristics. The criteria of becoming a super peer in [17, 18] are the capacity of the peer, in a way that the capacity of super peers must be greater or equal to all clients, so that the goal of SG-1 on the number of the super peers to be equal is fulfilled. In this algorithm, it is assumed that the capacity of peer is fixed and does not change. The algorithm for maintaining the characteristics of the system is based on the gossip model. The criteria of becoming a super peer in [19] are the capacity of peer and distance. SG-2 creates an overlay network with a few super peers in comparison with the size of the network that resists in face with the removal of a large number of peers. However, due to exchange of messages and selection of super peers, it takes a long time for the convergence to reach the high levels of super peer's capacity. In [20], making the firm overlay model is presented in line with peer-to-peer networks, which are based on super peers. This protocol has characteristics such as self-organizing, resiliency and consistency facing with Churn and fast and spontaneous accesses the high levels of capacity utilization in that unit. The capacity of a super peer is related to the number of peers that is able to support. Myconet super peers can be in different modes depending on the number of peers that are attached to them. In [21], Super Peer Selection (PPT) algorithm has been proposed. In this algorithm, the gossip model has been used to exchange peer information. In this algorithm, each peer with the help of gossip model periodically makes its own set of candidate super peers. Super peers are selected based on their capacity and the peer with the most capacity will be selected as the super peer candidate. Including adaptive super peer selection procedures in peer-to-peer networks, which are super peer based such as DLM [16], SG-1 [17, 18], SG-2 [19], Myconet [20], SPS [21], Jan Sacha algorithm [22,23] that is presented for Gradient Topology management and in this paper is expressed under the name of GT algorithm. In GT algorithm, a criterial called utility is used to select the super peers [22, 23]. The criteria of peers' utility are determined according to the particular application. Peers

with high utility locate in the center topology, while peers with low utility locate with gradually in the around the center. Selecting neighborhood in Gradient Topology is random also based on the utility of the peer. Recently Gradient Topology comparing to other mentioned adaptive methods has been used in many applications [30-33].

GT algorithm, which has been used to manage the gradient topology, is not sensitive to joining and leaving the peers. This problem becomes more challenging when a malicious peer frequently joins and leaves the network. In this paper, the proposed algorithm is sensitive to the removal of super peers and uses learning automata in order to select the super-peers in an adaptive manner. This paper is organized as follows. Section 2, an overview of learning automata that is used as the main learning strategy in the proposed algorithm, Section 3 states the problem, the proposed algorithm in Section 4, Section 5 Simulation results and in Section 6, the Conclusion is expressed.

II. THEORY OF LEARNING AUTOMATA

Learning automata [26-28], is a machine that can learn appropriate action from a finite set of actions in an unknown environment. The environment evaluates the selected action. The outcome of evaluation is given to the automata in a reward or punishment signal. The answer of the environment will affects on the learning automata to select the next action. The goal of the learning automaton is to learn to choose the best action from its own actions. The best action is the one that maximizes the probability of receiving a reward signal from the environment.

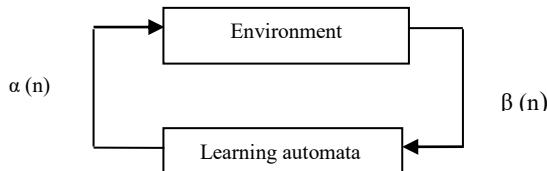


Figure 1: Relationship between learning automata and environment

Environment can be shown by $E \equiv \{\alpha, \beta, c\}$ in which $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ is the Set of inputs, $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ Set of outputs of the environment and $c \equiv \{c_1, c_2, \dots, c_r\}$ is the set of possibilities of receiving punishment signals. When β_i has two values, the environment is called the P-type. In this environment, $\beta_i(n) = 1$ is assumed as a punishment signal or failure and $\beta_i(n) = 0$, as a reward signal or success. In an environment of Q, β_i can discretely get one of the limited values in the interval [0,1]. In the S Model, β_i is a random variable between zero and one $\beta_i(n) \in [0,1]$ [29]. In S model with r action, if the action α_i is selected in n-th iteration step and the environment's answer to that is $\beta_i(n)$, the vector of automata probabilities will be updated according to the following equation.

$$P_i(n+1) = P_i(n) + a(1 - \beta_i(n))(1 - P_i(n)) - a\beta_i(n)P_i(n) \quad (4)$$

$$P_j(n+1) = P_j(n) - a(1 - \beta_j(n)) \cdot P_j(n) + a \cdot \beta_j(n) \cdot [\frac{1}{r-1} - P_j(n)] - a(1 - \beta_j(n)) \cdot P_j(n)$$

Based on the values of a and b three modes can be considered, When a and b are equal, the algorithm is called L_{RP}, When b is much smaller than a, the algorithm is called L_{ReP} and when b is equal to zero the algorithm is called L_{RI}. In the recent years, learning automata have been used in different applications such as Peer-to-Peer networks, Petri-net, Social networks, memetic computing, and Bayesian Optimization to mention a few [36]-[42].

III. STATEMENT OF THE PROBLEM

GT algorithm in Gradient Topology is not sensitive to removing super peers and there is no mechanism to consider it, in order to select new super peers, for this reason, for periodical selection of super peers, calculate adaptive threshold of selecting super peer. In GT algorithm that selection of super peers is done with two thresholds, the high threshold is shown by t_u and the low threshold is shown by t_l . The distance between the two thresholds is shown by Δ , whose value is obtained from equation 5. Selection of super peer using two thresholds is shown in Figure 2.

$$\Delta = t_u - t_l \quad (5)$$

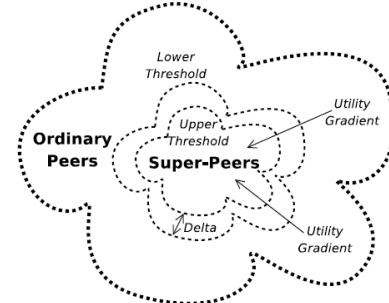


Figure 2: gradient topology[22]

In GT algorithm, threshold is computed in an adaptive way. If threshold is computed adaptively, peers must periodically perform some calculations. This calculation increases the network overhead. The overhead caused by network characteristics gossiping algorithm and estimation of super peer selection threshold. Each peer to become super peer in each period compares its utility with estimated threshold, which is a very costly. The algorithm for calculating adaptive threshold finally will cause the increase or decrease of super peer selection threshold, a change that will affect the distance between the two thresholds, i.e. Δ . If the distance of Δ increases, the error in the selection of appropriate super peer also increases [23]. If the selection of super peer is fixed threshold, the network cannot adapt the number of super peers with the current needs and holds the capacity of super peers in

maximum. Also in GT algorithm, super peer selection is only based on utility. In addition, peer stability to select it as super peer, is not considered and network is not sensitive to super peer deletion. This problem becomes more challenging when a malicious peer frequently joins and leaves the network.

IV. THE PROPOSED ALGORITHM

Initially, the network topology and data structure is described, the proposed algorithm is explained.

- Network has been created based on Gradient Topology with two thresholds and just in the first round, the thresholds are estimated adaptively based on [22, 23], after the first round, instead of recalculating the adaptive threshold, the primary adaptive thresholds remain fixed.
 - The utility of each peer has been considered according to its capacity. In addition, for peers P_i , utility $U(p_i)$ is equal to $C(p_i)$, the number of clients that give service to them.
 - It is assumed that the capacity of each peer is unique i.e. $U(p_i) \neq U(q_j)$.
 - We consider the relationship between all neighboring symmetric in a way that each peer discriminates between his own link and the link of its neighbors.
 - The entry and exit rate of peers are the same in a way that the size of the network, N , remains constant.
 - All peers which are in society of super peers, have a learning automata with variable actions of [28] operates with r act of $\{a_1, a_2, \dots, a_r\}$. r is the numbers of neighboring super peers, which are located at a distance of Δ , are corresponding to the learning automata. Note that, in this algorithm, the learning algorithm of the learning automata is L_{RP} and the model of environment is s .
 - p_{vector} vector, is the vector of probability vector of selection probability of learning automata function where p_i is the a_i -th probability of function selection or the i -th probability of neighboring selection as a super peer.
 - Each super peer in its neighboring table keeps the number of attached super peers, the number of connected clients also the probability vector of learning automata.

The steps of the proposed algorithm LAGT.1 is as follows:

Step 1. Each super peer periodically sends a Hello Packet to

its neighbors in the community of super peers, γ , and receives acknowledgement response from its neighbor, γ' . If $\log \gamma' / \gamma$ is

more than γ ”, this means that it did not receive any response from half of its super peer neighbors and predicts that some of the super peers be deleted.

Step 2. specifies the amount of Mt, i.e. the current number of super peers in the network. To do this, each super peer

collects the data of its neighboring super peer and estimates the rate of number of current super peer. Aggregation algorithm [35], in the community of super peers can achieve this goal. Aggregation algorithm acts based on periodic gossip.

Step 3. The number of optimization topology super peer can estimate the amount of deviation of the super peer's number of current topology using the following formula.

$$\mu_p = \log(M_t / M_t^*) \quad (6)$$

Table 1. Parameters of super peer topology

The current number of super peers in network	M_t
Optimization number of super peers in network	M^*_t
favorable rate of super peer in system	Q
Network size	N

Step 4. If μ_p , has a negative amount, it means that the number of super peers in network is low and some super peers should be chosen.

Step 5. Each super peer detects its neighbors that are in the interval Δ .

Step 6. Corresponding automata to each super peer, makes the initial value of function selection probability vector equal to $p_i = \frac{1}{n}$.

Step 7. learning automata of each super peer (p) using p_{vector} vector, randomly selects one of its neighbors (q) and as shown in formula 7, calculate the response of the environment i.e. β related to the selective neighboring.

$$\beta = \frac{1}{(C(p) - C(q))} \quad (7)$$

The more the amount of calculated β is closer to one, it will be rewarded and its selection probability vector will increase.

Step 8. Corresponding automata to super peer, again reviews the probability of its neighbors' selection from Δ to become super peer. This step is repeated until one of the neighbors is chosen at the previous replication θ .

Step 9. The peer, which is chosen at the previous replication θ , will be selected as super peer. This replication causes to select a peer as the super peer, which has long been in the neighborhood of previous super peer and after the selection, with high probability, for a longer time stays in the community of super peers. In other words, the peer will be selected that is more stable in network.

Step 10. The amount of deviation of the current super peers' number is estimated from the number of optimal topology super peer using 7 formula.

Step 11. If μ_p , has a negative amount, this means that the number of super peers in network is low and some super peers should be re-selected.

Step 12. The new selected super peer keeps its relation with the super peer that has previously been its client and starts to accept client.

Step 13. If μ_p , has a positive amount, this means that the number of super peers is high, so the super peer must be removed.

Step 14. To remove the super peer, each super peer, compares its capacity with its super peer neighbors so to decide for deletion and remove the super peer that has less capacity.

Step 15. In order to change the role of the super peer to client, the mentioned super peer cuts all its connections with its clients and will be the client of super peers, which has been connected to them.

Step 16. Each super peer estimates the amount of deviation of super peers' current capacity from the optimal amount of super peers' capacity using formula 8.

$$C_c(t) = C_t - C^*_t \quad (8)$$

In the above equation, C_t represents the capacity of super peers and C^*_t represents the optimal capacity of super peers. Since the threshold are considered based on the client, super peers' optimal capacity and current capacity is obtained by the following equations.

(9)

$$C^*_t = N - M^*_t$$

$$C_t = N - M_t \quad (10)$$

Step 17. If $C_c(t)$ has a negative amount, this means that the capacity of super peers has decreased, hence the new super peer should be selected.

Step 18. While If $C_c(t)$ has a positive amount, this means that the capacity of super peers has increased and some of the super peers must be degraded.

The algorithm presented in Figure 3, is performed in each super peer.

/* γ */ = Find out neighbors of super-peer just in super peer set but not send Acknowledge
 P_{vector} = Set of probability values for r
LAP = Embedded Learning Automata
 θ = The number of iteration as super peer selection
 M_t = current number of super-peers
 M^*_t = optimum number of super-peers
 C_t = total super-peer capacity
 C^*_t = optimum super-peer capacity */

```

while (TRUE)
{
    if super-peer
    {
        Hello Packet to neighbor & wait to receive Acknowledge;
        if receive Acknowledge >  $\gamma$ 
            calculate  $\mu_p = \log(M_t / M^*_t)$ ;
            if  $\mu_p < 0$ 
                Loop1:
                {
                    r = Find out neighbors of super-peer just in  $\Delta$ ;
                     $P_{vector}$  = Create probabilities vector (r);
    }
}

```

```

Initialize probabilities vector ( $P_{vector}$ );
/* Select a peer as a super peer according to the
action probability vector of learning automata */
do
{
    select neighbor from  $P_{vector}$ ;
    compute  $\beta$ ;
    Update LAP by EQ (4);
} while ( $\theta = 5$ )
Become super peer();
}

 $\mu_p = \log(M_t / M^*_t)$ ;
 $C^*_t = N - M^*_t$ ;
 $C_t = N - M_t$ 

 $C_c(t) = C^*_t - C_t$ ;
if  $\mu_p < 0$  or  $C_c(t) < 0$ 
    Go to Loop1:
    Else
        if  $\mu_p > 0$  or  $C_c(t) > 0$ 
            delete super peer with lowest capacity
}

```

Figure 3: LAGT.1 Proposed algorithm

V. SIMULATION RESULTS

In order to stimulate the proposed super peer selection algorithm, the PeerSim [34] stimulation is used. The proposed algorithm of LAGT.1 is compared with the two algorithms of SG-1 and GT [22, 23]. Table 2 summarizes the simulation assumptions with their initial values.

Table 2. Simulation parameters

The maximum number of peer in networks	100000	
Maximum number of neighbors of each peer	150	Based on similarity(10-74)
		At random(10-74)
Utilities	0-1000	
A	0.5	
B	0.5	
γ	0.3	
Churn	50% - 10%	

The simulation time is equal to 30 time rounds. Each round is equal to a second. Each test is performed for several times and the mean number is calculated.

• Experiment 1:

In this experiment, it will be examined that if some percent of super peers are removed and the maximum capacity of super peers is decreased, how many rounds it takes until each algorithm reach to the maximum capacity of super peers. In this experiment, the threshold is considered based on the client's threshold. Results are studied for a case that the capacity of super peers is 36×10^4 and we want it to be 43×10^4 , which is the maximum optimal capacity of super peers. In this experiment, the network has 10%, churn. At around 5, some percent of super peers will be removed and it will be checked that in which around the maximum capacity of super peers will be achieved. In algorithm SG-1, 25 rounds, in algorithms GT- clients, 13 rounds and in algorithm LAGT.1 is

6-7 rounds. The main reason for the less time of convergence in the proposed algorithm is the removal of gossip and reduction of calculation in the entire network, to estimate the threshold. As can be seen, the proposed algorithm has a linear trend chart but after around 5 that learning in algorithm has shown itself, slope of the curve decreases and more rapidly achieves the desired capacity. GT-Clients algorithm, due to the existence of churn and lack of attention to the stability of peer and selection of super peer only based on the utility, has worse performance than the proposed algorithm.

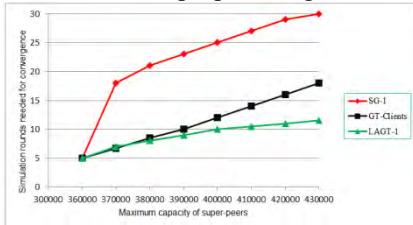


Figure 4. Convergence time to reach maximum capacity of super peers

• Experiment 2:

In this experiment, we have examined the robustness of the network, against the removal of 10%, 20% and 30% of super peers in Round 7. The results indicate the relationship between removal of a percentage of super peers with the number of clients that remain without the super peers and the number of rounds, which is required for selection of super peers in order to recover the network. In SG-1 algorithm, in every test with removal of super peers, more clients remain without super peer and it takes more number of rounds for the network to recover itself. After removing the super peers in round 7, for several rounds GT algorithm comparing to LAGT-1 have had a better performance. The reason is that the proposed algorithm is in the early learning capability, but LAGT-1 algorithm comparing to GT needs the number of rounds for selecting the appropriate super peers to recover the network. The results indicate that LAGT-1 have been more robust against removing a percentage of the super peers. In table 3, the relationship between the removals of a percentage of super peers with the required number of rounds to recover the network, is shown for three algorithms of SG-1, GT and LAGT-1.

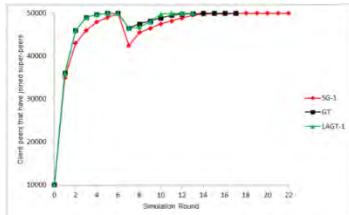


Figure 5. Network robustness against removal of 10% of super peers

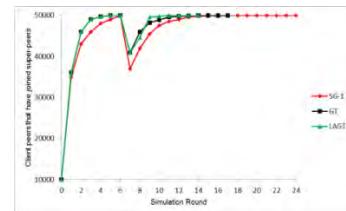


Figure 6. Network robustness against removal of 20% of super peers

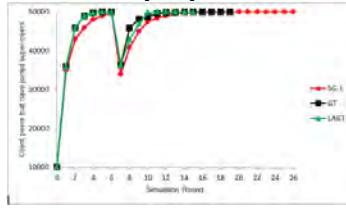


Figure 7. Network robustness against removal of 30% of super peers

Table 3. The maximum round for recovering the network for algorithms SG-1, GT and LAGT

	The maximum round, in case of removing 10% of super peers	The maximum round, in case of removing 20% of super peers	The maximum round, in case of removing 30% of super peers
SG-1	55	52	52
GT	91	91	91
LAGT-1	91	92	92

• Experiment 3:

In this experiment, we have examined the robustness of the network, against the frequently adding and removing 20% of random super peers as malicious peers. In table 4, the results are reported. The results indicate that LAGT-1 have been more robust against malicious super peers.

Table 4. The maximum round for recovering the network for algorithms SG-1, GT and LAGT

	The maximum round when the super-peers frequently added and removed between round 1 to 10	The maximum round when the super-peers frequently added and removed between round 30 to 40
SG-1	28	27
GT	23	21
LAGT-1	19	18

VI. CONCLUSIONS

In this paper, an adaptive algorithm for managing gradient topology utilizing learning automata is proposed. The proposed algorithm unlike GT algorithm in Gradient Topology that is not sensitive to removing super peers, is the first algorithm that is sensitive to the removing super peers. According to the simulations, results show that the proposed algorithm compared to the SG-1 and GT, in case of removing a percentage of super peers, in less time reaches the maximum capacity of super peers and comparing to mentioned algorithms selects appropriate super peers faster to reconstruct the

network. Moreover, in fewer numbers of rounds reaches the desired maximum capacity and number of super peers.

REFERENCES

- [1] Schollmeier, R., 2002, *A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications*, In Proceedings of the First International Conference on Peer-to-Peer Computing UniversitatMunchen, Germany, P2P'01, pp 27–29.
- [2] Yang, B., Garcia-Molina, H., March 2003, *Designing a super-peer network*, In Proceedings of the 19th International Conference on Data Engineering, Bangalore, India, IEEE Computer Society, pp 49-60.
- [3] Lua, E. K., Crowcroft, J., Pias, M., Sharma, R., Lim, S., Apr 2005, *A Survey and Comparison of Peer-to-Peer Overlay Network Schemes*, in IEEE Communications Surveys and Tutorials, vol 7, pp72–93.
- [4] Karger, D. R., Ruhl, M., 2004, *Simple efficient load balancing algorithms for peer-to-peer systems*, In Proceedings of the 16th Annual ACM Symposium on Parallelism in Algorithms and Architectures, New York, NY, USA, pp 36–43.
- [5] Lo, V., Zhou, D., Li, J., 2005, *Scalable super node selection in peer-to-peer overlay networks*, In Proceedings of the Second International Workshop on Hot Topics in Peer-to-Peer Systems, Washington, DC, USA, IEEE Computer Society, pp 18–27.
- [6] Min, S. H., Holliday, J., Cho,D. S., 2006, *Optimal Super-peer Selection for Large-scale P2P System*, International Conference on Hybrid Information Technology.
- [7] Liang, J., Kumar, R., Ross, K., April 2006, *The KaZaA overlay: A measurement study*, Computer Networks, pp 842_858.
- [8] LimeWire, Accessed on 25th March 2008, How gnutella works, http://wiki.limenwire.org/index.php?title=How_Gnutella_Works.
- [9] Kleis, M., Lua, E. K., Zhou, X., 2005, *Hierarchical peer-to-peer networks using lightweight super peer topologies*, In Proceedings of the 10th IEEE Symposium on Computers and Communications, pp 143-148.
- [10] Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., 2002, *EDUTELLA: a P2P networking infrastructure based on RDF*, In Proceedings of the 11th international conference on World Wide Web, ACM, pp 604_615.
- [11] Wang, T. I., Tsai, K. H., Lee, Y. H., 2004, *An efficient and stable distributed resource lookup protocol*, In Proceedings of the 1st International Conference on Embedded and Ubiquitous Computing, vol 3207 of Lecture Notes in Computer Science, pp 1075_1084.
- [12] Aditya, K., Christopher, G., David J. M., George, K., January 2015, *Optimizing cluster formation in super-peer networks via local incentive design*, in *Peer-to-Peer Networking and Applications* Journal, Volume 8, Issue 1, pp 1-21.
- [13] Pirro, G., Talia, D., Trunfio, P., April 2012, *A DHT-based semantic overlay network for service discovery*, *Future Generation Computer Systems*, Elsevier Journal, Volume 28, Issue 4, pp 689–707.
- [14] Li, Y., Huang, X., Ma, F., Zou, F., 2005, *Building efficient super-peer overlay network for DHT systems*, In Proceedings of the 4th International Conference on Grid and Cooperative Computing, Springer, vol 3795, pp 787_798.
- [15] Shi, L., Zhou, J., Huang, Q., Dec 2013, *A Chord-based super-node selection algorithm for load balancing in hybrid P2P networks*, IEEE/*Mechatronic Sciences, Electric Engineering and Computer (MEC)*, pp 2090 – 2094.
- [16] Xiao, L., Zhuang, Z., Liu, Y., 2005, *Dynamic layer management in superpeer architectures*, IEEE Transactions on Parallel and Distributed Systems, vol 16, pp 1078-1091.
- [17] Montresor, A., 2004, *A robust protocol for building superpeer overlay topologies*, In Proceedings of the 4th International Conference on Peer-to-Peer Computing, IEEE Computer Society, pp 202-209.
- [18] Gholami, Sh., Meybodi, M. R., Saghiri, A. M., 2014, *A Learning Automata-Based Version of SG-1 Protocol for Super-Peer Selection in Peer-to-Peer Networks*, Proceedings of the 10th International Conference on Computing and Information Technology (IC2IT2014), Volume 265, pp 189-201.
- [19] Jesi, G. P., Montresor, A., Babaoglu, Ö., 2006, *Proximity-aware superpeer overlay topologies*, In Proceedings of the 2nd IEEE International Workshop on Self-Managed Networks, Systems, and Services, vol 3996, pp 43-57.
- [20] Snyder, P. L., Greenstadt, R., Valetto, G., Sept 2009, *Myconet: A Fungi-inspired Model for Superpeer-based Peer-to-Peer Overlay Topologies*, 3rd IEEE International Conference, pp 40-50.
- [21] Liu, M., Harjula, E., Ylianttila, M., 2013, *An efficient selection algorithm for building a super-peer overlay*, Springer, Journal of Internet Services and Applications, vol 4(4), pp 1–12.
- [22] Sacha, J., July 2009, *Exploiting Heterogeneity in Peer-to-Peer Systems Using Gradient Topologies*, A thesis submitted to the University of Dublin, Trinity College in fulfillment of the requirements for the degree of Doctor of Philosophy (Computer Science).
- [23] Biskupski, B., Sacha, J., Dahlem, D., Cunningham, R., Meier, R., Dowling, J., Haahr, M., 2010, *Decentralising a service-oriented architecture*, *Peer-to-Peer Networking and Applications*, The Netherlands, Ireland and Sweden, Springer, Vol. 3.
- [24] Garbacki, P., Epema, D. H. J., Steen, M., March 2010, *The Design and Evaluation of a Self-Organizing Superpeer Network*, Computers IEEE Transactions, vol 59, pp 317-331.
- [25] Teng, H. Y., Lin, C. N., Hwang, R. H., September 2013, *A self-similar super-peer overlay construction scheme for super large-scale P2P applications*, Springer US, *Information Systems Frontiers* Journal, Volume 16, Issue 1, pp 45-58.
- [26] Narendra, K. S., Thathachar, M., 1989, *Learning Automata: an introduction*, New York, Prince-Hall.
- [27] Najim, K., Poznyak, A. S., 1994, *Learning automata: theory and application*, in Proceeding of the Tarrytown, New York, Elsevier Science Publishing Ltd.
- [28] Thathachar, M. A. L., Bhaskar, R. H., 1987, *Learning automata with changing number of actions*, IEEE Transactions on Systems, Man and Cybernetics, vol. 17, no. 6.
- [29] Narendra, K. S., Thathachar, M. A. L., *Learning Automata: An Introduction*, Prentice Hall, 1989.
- [30] Payberah, A. H., Dowling, J., Rahimian, F., Haridi, S., June 2012, *Distributed optimization of P2P live streaming overlays*, Springer Journal.
- [31] Payberah, A. H., Dowling, J., Haridi, S., 2011, *Glive: the gradient overlay as a market maker for meshbased p2p live streaming*, In: Proceedings of the 10th IEEE international symposium on parallel and distributed computing (ISPDC'11), pp 153–162.
- [32] Payberah, A. H., Dowling, J., Rahimian, F., Haridi, S., 2010, *gradientv:Market-based p2p live media streaming on the gradient overlay*, In: Proceedings of the 10th IFIP international conference on distributed applications and interoperable systems (DAIS'10), pp 212–225.
- [33] Payberah, A. H., Dowling, J., Rahimian, F., Haridi, S., 2010, *Sepidar: incentivized market-based p2p livestreaming on the gradient overlay network*, In: Proceedings of the IEEE international symposium on multimedia (ISM'10), pp 1–8.
- [34] Jelasity, M., Montresor, A., Jesi, G.P., Voulgaris, S., 2009, *PeerSim: P2P Simulator*, <http://peersim.sourceforge.net>.
- [35] Sacha, J., Dowling, J., Cunningham, R., Meier, R., June 2006, *Using aggregation for adaptive super-peer discovery on the gradient topology*, In Proceedings of the 2nd IEEE International Workshop on Self-Managed Networks, Systems & Services (SelfMan), number 3996 in Lecture Notes in Computer Science, Springer- Verlag, pp 77-90.
- [36] Saghiri, A. M. and Meybodi, M. R., *An Approach for Designing Cognitive Engines in Cognitive Peer-to-Peer Networks*, Journal of Network and Computer Applications, 2016
- [37] Saghiri, A. M. and Meybodi, M. R., *A Distributed Adaptive Landmark Clustering Algorithm Based on mOverlay and Learning Automata for Topology Mismatch Problem in Unstructured Peer-to-Peer Networks*, International Journal of Communication Systems, 2015, DOI: 10.1002/dac.2977
- [38] Saghiri, A. M. and Meybodi, M. R., *A Self-adaptive Algorithm for Topology Matching in Unstructured Peer-to-Peer Networks*, Journal of

Network and Systems Management, Springer, Vol. 24, pp. 393-426, 2016, DOI: 10.1007/s10922-015-9353-9

- [39] Vahidipour, M. and Meybodi, M. R. and M. Esnaashari, *Learning Automata Based Adaptive Petri net and Its Application to Priority Assignment in Queueing Systems with Unknown Parameters*, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 45, No. 10. pp. 1373-1384, October 2015, DOI: 10.1109/TSMC.2015.2406764
- [40] Rezvanian, A. and Meybodi, M. R., *A New Learning Automata Based Sampling Algorithm for Social Networks*, International Journal of Communication Systems, 2015, DOI: 10.1002/dac.3091
- [41] Rezapoursaleh, M. and Meybodi, M. R., *A New Memetic Algorithm based on Cellular Learning Automata for Solving the Vertex Coloring Problem*, Memetic Computing, 2016, DOI: 10.1007/s12293-016-0183-4
- [42] Moradabadi, B., Ebadzadeh, M. H. and Meybodi, M. R., *A New Real-Coded Stochastic Bayesian Optimization Algorithm for Continuous Global Optimization*, Genetic Programming and Evolvable Machines, Vol. 17, No. 2, pp. 145-167, 2016, DOI 10.1007/s10710-015-9255-3