Received: 17 April 2018 | Revised: 31 October 2019 | Accepted: 10 August 2020
DOI: 10.1002/dac.4661

RESEARCH ARTICLE

WILEY

A self-adaptive algorithm for super-peer selection considering the mobility of peers in cognitive mobile peer-to-peer networks

Nahid Amirazodi[1] | **Ali Mohammad Saghiri**[2] | **Mohammad Reza Meybodi**[2]

[1]Department of Electrical, Computer and IT Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

[2]Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, Iran

Correspondence
Nahid Amirazodi, Department of Electrical, Computer and IT Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran.
Email: n.amirazodi@gmail.com

Summary

Cognitive peer-to-peer networks are obtained from a combination of cognitive networking concepts and peer-to-peer networks. These networks are able to improve their performance while operating under dynamic and unknown environments. A cognitive peer-to-peer network tries to learn an appropriate configuration for itself considering the unknown physical properties of peers. Cognitive mobile peer-to-peer networks refer to cognitive peer-to-peer networks which are built over mobile ad hoc networks. In these networks, heterogeneity of the mobility of peers and resource limitation in wireless networks create challenges for network management algorithms. Because of the dynamicity of these networks, the management algorithms should be designated in self-adaptive manner. In one type of these networks, some peers, called super-peers, undertake to perform network managerial tasks. The mobility of peers leads to connection failure among peers and reselection of new super-peers. Therefore, the selection of super-peers, due to their influential role, requires an algorithm that considers the peers' mobility. Up to now, no self-adaptive algorithm has been designated for super-peer selection considering the mobility of peers in a self-adaptive manner. This paper proposes M-SSBLA, a self-adaptive algorithm for super-peer selection considering the mobility of peers based on learning automata. The proposed algorithm is obtained from cooperation between a learning automata-based cognitive engine and MIS. MIS is a well-known super-peer selection algorithm in mobile peer-to-peer networks. We compared the proposed algorithm with recently reported algorithms, especially for a network with high mobility. Simulation results show that the proposed algorithm can cover maximum ordinary-peer with a few super-peer and improve robustness against super-peer failures while decreasing maintenance overhead.

KEYWORDS

cognitive mobile peer-to-peer networks, cognitive networks, learning automata, super-peer selection

1 | INTRODUCTION

Cognitive peer-to-peer (CP2P) networks learn from past experiences and improve their performance while operating under unknown and dynamic environments.[1] The underlying network architecture of CP2P has a significant impact on the overall network performance.[1] CP2P networks may be constructed over mobile ad hoc networks as overlay networks. Cognitive mobile peer-to-peer (CMP2P) networks try to adaptively find appropriate configuration for the networks in which the physical properties of the peers are unknown. In Saghiri and Meybodi,[1] it is studied that how learning algorithms may be used to solve network management problems of CP2P networks.

In one type of CMP2P networks, peers in every local area select one among themselves as *super-peer (SP)* in a distributed manner. *Super-peer* is in charge of managing other peers. All other participating peers in the local area are *ordinary-peers (OP)*. Each super-peer is connected to a set of ordinary-peers that are in its neighboring area. Ordinary-peers of an area are connected to only one super-peer of the area. Super-peers deal with all queries instead of respective ordinary-peers as well as communicate with other super-peers.[2,3]

Super-peer–based CP2P networks take advantage of centralized mechanisms while benefiting from the robustness of distributed algorithms. Utilizing super-peers leads to increasing query response time along with less overhead and traffic because query processing only affects super-peers.[3–6] The problem of designing a self-adaptive and distributed algorithms for selecting a set of peers as super-peers is challenging. It should be noted that the mobility pattern and many characteristics of peers, such as peer lifetime and capacity, are unknown for the super-peer selection algorithms. The peer mobility poses many challenges to the super-peer selection algorithms. The topology of network frequently changes because of the highly dynamic property of communication in ad hoc networks. This causes network performance degradation and system instability, and the super-peer selection process becomes more challenging.[3,7]

The performance of the networks depends on effectively establishing super-peers and how long-established sets can be maintained without breaking apart due to the mobility of peers. If super-peers move fast and ordinary-peers move slowly or on the contrary, the sets may be broken because their ordinary-peers cannot communicate with their own super-peers. Super-peer reselection imposes overhead traffic to the network and increases the number of unnecessary super-peers.[4,8] In addition, the topology of the super-peer layer may be changed by super-peer selection process, because of the incompatibility or transmission limitations imposed by the natural heterogeneity of super-peer–based management algorithms. Therefore, maintaining super-peer stability leads to performance preservation.[4,8] To design adaptive management algorithms for these networks, we need a distributed algorithm that can adapt to the dynamic conditions of the network. Up until now, no self-adaptive algorithm has been designed for super-peer selection considering the mobility of peers.

In this paper, we propose a self-adaptive algorithm for super-peer selection considering the mobility of peers that uses learning automata as an adaptive decision-making mechanism. Learning automata have been found to be useful in dynamic environments such as mobile peer-to-peer (MP2P) and peer-to-peer (P2P) networks.[2,9] The proposed algorithm is obtained from fusing a learning automata-based cognitive engine to *MIS*[10] algorithm that is a well-known super-peer selection algorithm in MP2P networks. The rest of the paper is organized as follows. Theory of learning automata and the cognitive networking are explained in Section 2. In Section 3, the related literature is reviewed. In Section 4, a self-adaptive super-peer selection algorithm considering the cognitive networking framework is proposed. Section 5 gives the performance evaluation of the proposed algorithm, and finally, Section 6 concludes the paper.

2 | PRELIMINARIES

In this section, in order to provide the necessary information for the remainder of the paper, a brief overview of learning automata and the cognitive networking framework are explained.

2.1 | Learning automata

Learning automaton (LA) is a self-adaptive decision-making model.[11,12] This model belongs to reinforcement learning field in artificial intelligence. This model improves its performance by learning how to choose an optimal action from a finite set of actions. The learning process of this model is done through repeated interactions with a random environment (Figure 1). LAs can be classified into two classes: fixed and variable structure.[2] A variable structure LA can be

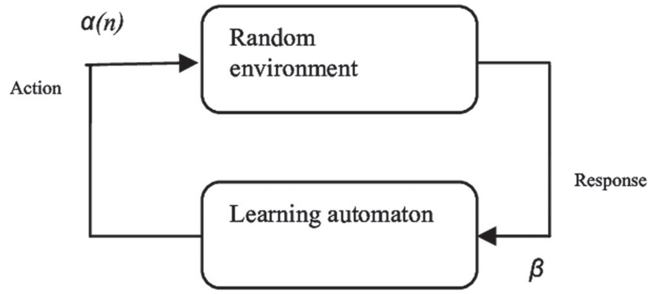


FIGURE 1 Learning automaton and the environment

represented by a triple <β,α,L> where β is the set of feedbacks of the environment, α is the set of actions, and L is learning algorithm. Let $\alpha_i(k) \in \alpha$ and P(k) denote the action selected by LA and the probability vector defined over the action set at instant k, respectively.

The a and b indicate the reward and the penalty parameters, respectively. Parameter r indicates the number of actions that can be selected by LA. At step n, the action is selected based on action probability vector. Then, the action probability vector P(n) is updated by the linear learning algorithm given in 1, if the selected action $\alpha_i(n)$ is rewarded by the environment, and it is updated as given in 2, if the taken action is penalized. If a = b, the recurrence equations 1 and 2 are called linear reward penalty (L_{RP}) algorithm. More information can be found in Najim and Poznyak[11]:

$$\begin{aligned} p_i(n+1) &= p_i(n) + a[1-p_i(n)] \\ p_j(n+1) &= (1-a)p_j(n) \qquad \forall j,\ j \neq i \end{aligned}, \tag{1}$$

$$\begin{aligned} p_i(n+1) &= (1-b)p_i(n) \\ p_j(n+1) &= \frac{b}{r-1} + (1-b)p_j(n) \quad \forall j,\ j \neq i \end{aligned}. \tag{2}$$

In the recent years, LA has been used in different applications such as cognitive networks,[1] ad hoc networks,[13] wireless sensor networks,[14] WiMAX networks,[15] network security,[16] wireless mesh networks,[17] mobile video network surveillance,[18] vehicular environment,[19,20] P2P networks,[9,21,22] wireless data broadcasting systems,[23–25] smart grid systems,[26] grid computing,[27] and cloud computing,[28] to mention a few.

2.2 | CMP2P networks

This section describes the structure of CMP2P networks. These networks learn from previous experiences and improve their decisions about the configuration of the network in a self-adaptive manner. In this section, we use the definition of CP2P networks given in Saghiri and Meybodi[1] to deploy them in CMP2P networks. CMP2P networks are defined as mobile networks with cognitive processes capable of learning from the results of their actions. A cognitive process recognizes current network situations (plans, conditions, etc.) and acts based on them.

The CMP2P network consists of a set of cognitive peers that each one corresponds to a peer in the mobile network. The topological structure of the CMP2P network is isomorphic toward the mobile network. The structure of the framework that is used in a cognitive peer is shown in Figure 2. This framework consists of three layers: *Requirement Layer, Cognitive Process Layer*, and *SAN Layer*. The definition of each layer is given in the rest of this section.

- Requirement layer: In this layer, the goals and the behaviors of the network are described using cognitive specification language.
- SAN layer: In this layer, the network status sensors and modifiable elements are designated. Each cognitive peer uses its network status sensors for gathering local information about its corresponding peer in the mobile network to observe its environment. Each cognitive peer also acts on its modifiable elements to change the local configuration of its corresponding mobile peer in the mobile network.
- Cognitive process layer: The cognitive process layer is implemented using a set of cognitive engines resided in the cognitive peers.

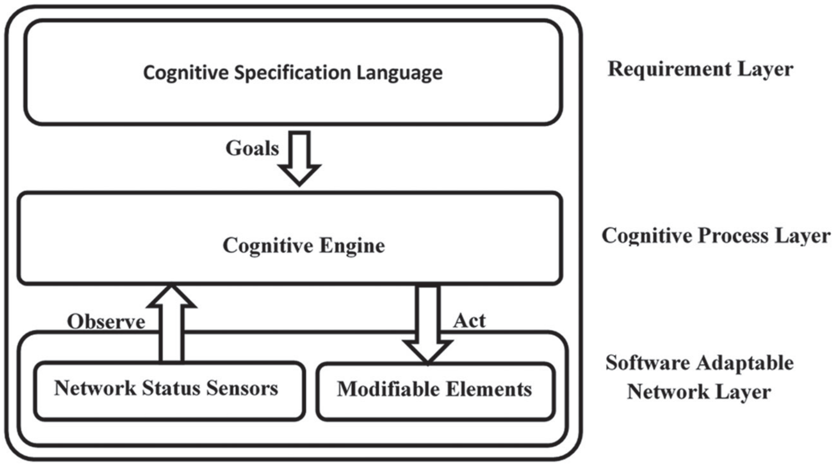


FIGURE 2 The framework which is used in a cognitive peer[1]

The main issue in CMP2P networks is the mobility of the peers that must be considered. This issue extremely effects on designing cognitive engine because the cognitive engine should make decisions in online fashion considering feedbacks from environments that are totally unknown during the network operation. In this paper, a cognitive engine based on learning automata will be designated for solving super-peer selection problem in a MP2P network.

3 | RELATED WORKS

Several algorithms for the super-peer selection problem are reported for MP2P networks in the literature. In the literature,[29,30] super-peers are selected manually. These algorithms suffer from several drawbacks such as lack of scalability and robustness to the changes that occur in the network.

In Han et al,[10] the MIS algorithm determines super-peers in a distributed greedy manner. This algorithm is much natural in realistic mobile P2P environments. The MIS algorithm selects a random number based on a maximal independent set for each peer in a local area. Then, a peer with the most significant number among its neighbor peers is upgraded to super-peer. The main drawback of MIS algorithm is that it is not self-adaptive. After each fault, it must be executed again that is a costly process.

The suggested algorithms in previous studies[3,4,10,31–34] consist of methods that initially separate the population of peers into several groups. Then super-peers are independently selected within each group. The grouping usually is based on peer properties such as physical location, network proximity, and semantic content. These algorithms focus on the capacity heterogeneity of their peers and take into account the difference among them.

In Bin et al,[31] the SSBDC algorithm tries to dynamically select the peers with high performance as super-peer considering peers capacity and stability and then adopts an alternative to back up super-peer. Also, this algorithm partially considers the mobility of peers by considering the rate of speed and communication radius. Its authors believe that peers with low speed and high radius are desirable to become super-peers. In Bin et al,[31] all peers compute their capacity, uptime, and movement rate, then compare their statistics with each other. Then, they select high capability peers as super-peers. The algorithms reported in other works[3,33] use static values as the capacity thresholds to select super-peers. A peer will be a super-peer candidate if its superiority ratio exceeds a certain threshold. Determining the thresholds is an application-dependent task.

In SSBLA algorithm,[2] an adaptive algorithm is suggested for super-peer selection that considers the capacity of peers. Nevertheless, this algorithm did not take into account the mobility of peers in an adaptive manner. Therefore as the speed of peers increase, the performance and the stability of the peers connections will be reduced.

The reported algorithms in the literature[4,8,31,35] are based on peer mobility patterns. These algorithms assume that low-speed peers are suitable to play a super-peer role. Their authors believe that selecting peers with low speed as super-peer leads to network stability.

All of the existing mobility-based super-peer selection algorithms reported in previous works[2–4,8,31,35] did not consider mobility of peers in a self-adaptive manner. In practice, the low-speed super-peer assumption of the mentioned algorithm is not correct in a network with high or diverse speed peers. In these situations, connections quickly break, and super-peer must be reelected frequently. These events impose significant overhead for the network. Therefore, these algorithms are not efficient. In this paper, an algorithm considering the mobility of peers will be proposed, which adaptively selects the super-peers. The proposed algorithm is obtained from the cooperation between a learning automata-based cognitive engine and an MIS algorithm.

4 | PROPOSED ALGORITHM: M-SSBLA

In this section, first, we provide a general overview of the system model and proposed algorithm. Afterwards, we will give a detailed description of the proposed algorithm (M-SSBLA).

4.1 | System model

A snapshot for a set of peers that trying to connect to each other through a P2P network is given in Figure 3. The network is formed in three steps described as bellow.

- **Step 1:** In this step, peers find their neighbors to exchange information with them. Each peer maintains a list of its neighbors in a set called local view.
- **Step 2:** In this step, a basic algorithm selects super-peers. Then other peers connect to corresponding super-peers in their local area as ordinary-peer. The MIS algorithm that was mentioned previously is used in the second step. Now, each peer in the constructed network can have three types of roles: super-peer, ordinary-peer, bridge-ordinary-peer. Super-peers and ordinary-peers are already illustrated in an example given in Figure 3. Some of the ordinary-peers will have only one super-peer within their communication range, while some others have several super-peers within their communication range that act as intermediate peers between super-peers and are called edges. Ordinary-peers change their position from edge to none-edge peer and vice versa during their presence in the network. Bridges-ordinary-peers can select a super-peer among neighboring super-peers that have "same-direction" with them. Same-direction means ordinary-peer and its corresponding super-peer have similar speed and movement paths.

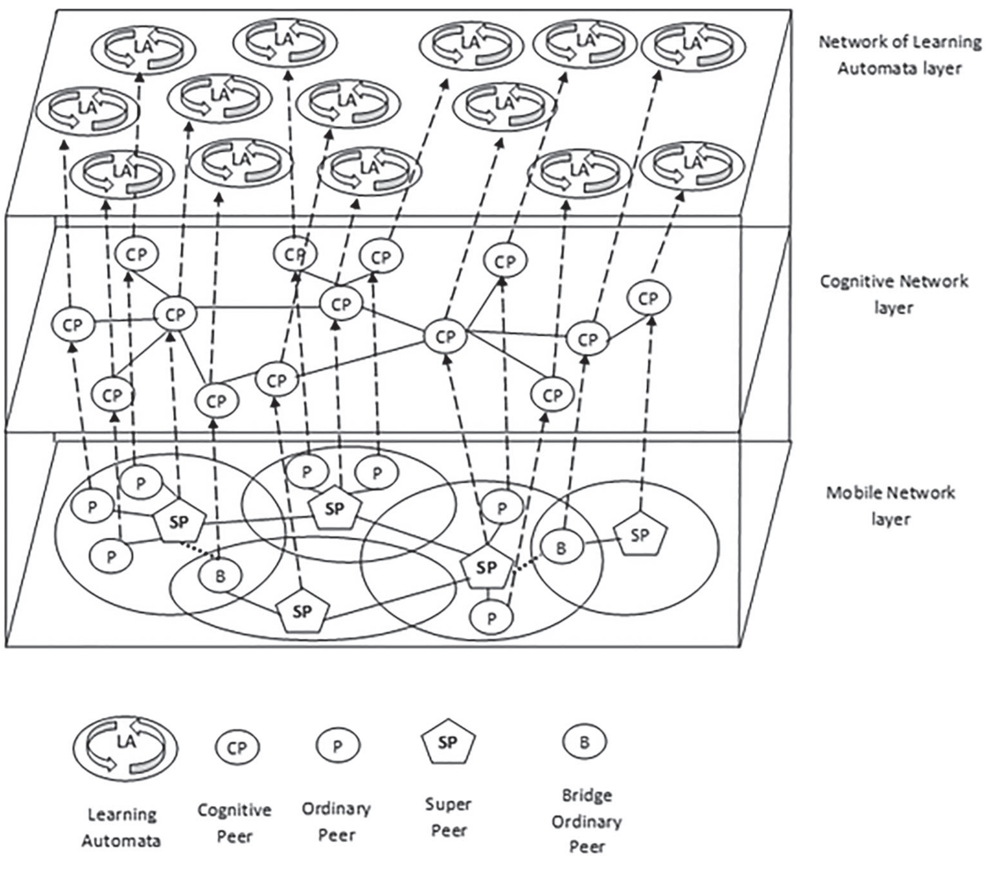


FIGURE 3 Super-peer–based CMP2P network and the network of learning automata

- **Step 3:** In this step, ordinary-peers begin to find same-direction super-peer using an LA. LA discovers which one of the super-peers in the local view is accessible most of the time. If a super-peer, most of the time, is accessible while all peers are moving, the super-peer has the same direction. Afterwards, ordinary-peer connects to desirable super-peer as far as possible in the next periods. In this way, ordinary-peers predict that whether the current super-peer is accessible or it will be lacked soon.

In the rest of this section, we explain the proposed algorithm (M-SSBLA) in more details.

Remark 1 The proposed algorithm that has the role of the cognitive engine in the network is obtained from the cooperation between a basic super-peer selection algorithm and a distributed decision-making mechanism based on LA. We select *MIS* algorithm[10] as a basic underlay algorithm because it is well-known. In other word, the rationale behind of fusing theory of learning automata to MIS can be used to enhance other super-peer selection algorithms.

4.2 | Proposed algorithm: M-SSBLA

Some parts of this algorithm were explained in the previous section, and therefore, we focus on processes executed after forming a super-peer network. Subsequently, the proposed algorithm periodically checks network environment changes (Figure 4). If there is not super-peer in the local view, ordinary-peer selects a new super-peer by *the MIS* algorithm. Else, if there is one super-peer in local view, ordinary-peer connects to it. Otherwise, if there are several super-peers in communication range, ordinary-peer can choose one of them based on the output of *the Desirable-super-peer-selector* algorithm given in Figure 5.

```
Algorithm M-SSBLA
    Assumptions:
    P: local peer
    SPp: P's super-peer
    Local-view: set of neighbor peers
Begin
P forms it's local view
P Calls P.MIS() to select super-peer
If According to P.MIS.Output ()P is Super-peer
   P manage ordinary-peers
Else if According to P.MIS.Output () P is ordinary-peer
   P waits while, local-view will change
      For each super-peer joins or leaves in transmission range of P
         If SPp lack, calls P.MIS()
          If  output super-peer of  Desirable-super-peer-LA is current super-peer
             P stays on current super-peer
          Else
             P connects to output super-peer of function desirable-super-peer-LA()
             //given in Fig.5
          End If
      End For
End If
End
```

FIGURE 4 M-SSBLA algorithm

FIGURE 5 Learning phase for each peer p

```
Algorithm Desirable-super-peer-LA

  Assumptions:
  Action set: {desirable-super-peer, undesirable-super-peer}
  I: index of super-peers in local view

  Begin
    For each super-peer(i) in local-view do
    Activate the learning automaton of super-peer(i) and choose an action
    Send hello message to super-peer(i)
    Wait while TTL time or ACK is received
    If selected action is desirable-super-peer then
      If ACK is received then
        Update action probability vector of super-peer(i) by equation (1)// rewarding
      Else
        Update action probability vector of super-peer(i) by equation (2)// penalizing
    Else if selected action is undesirable-super-peer then
      If ACK is received then
        Update action probability vector of super-peer(i) by equation (2)// penalizing
      Else
        Update action probability vector of super-peer(i) by equation (1)// rewarding
  End For
 End For
 Select a peer as super-peer in which the probability of selecting "desirable-super-peer" action of it learning automaton is maximum
 End
```

In the proposed algorithm, each ordinary-peer is equipped with variable structure LA and uses a super-peer selector algorithm based on an algorithm called Desirable-super-peer-selector. Desirable-super-peer-selector algorithm uses the LA with L_{RP} learning algorithm with two actions: "desirable-super-peer" and "un-desirable-super-peer." After connecting to a super-peer, each ordinary-peer repeats learning process upon departure and arrival of any new peer from/to its communication range. Figure 4 shows the high-level structure of the proposed algorithm.

4.3 | Function desirable-super-peer-selector

During execution of desirable-super-peer-selector algorithm, each ordinary-peer p establishes an LA for each super-peer_i in its communication range. The LA has two actions in action sets: "desirable-super-peer" and "un-desirable-super-peer." At first, the LA of super-peer_i randomly selects an action for it. Afterward, it sends a hello message to the super-peer_i. Based on the super-peer_i reaction and its selected action, the corresponding automata probability vector will be updated by rewarding or penalizing. If the super-peer_i response received in a certain time, super-peer_i is "desirable-super-peer" and the "desirable-super-peer" action in the probability vector gets reward. In this case, "un-desirable-super-peer" action gets penalty. Else, if super-peer_i is missed or does not answer in certain time, it is "un-desirable-super-peer" and the "un-desirable-super-peer" action in probability vector gets reward. Therefore, in this case, "desirable-super-peer" action gets penalty. The output of a desirable-super-peer-selector is the result of learning process. Based on this process, a super-peer with maximum "desirable-super-peer" probability will be returned as output (Figure 5).

Remark 2 The MIS uses random numbers for selecting super-peers. Its super-peer selection algorithm mimics the Luby's randomized maximal independent set algorithm. Each peer chooses a random number. Then among the peers within the communication range, a peer who has the largest number is selected as a super-peer.

Remark 3 During the learning process, the probability of selecting the same direction super-peer increases, and the probability of selecting none same direction decreases. As a result of learning process, peers with high speed and same direction, recognize each other and connect to appropriate peers. In addition, low-speed peers connect to appropriate (low-speed) peers.

5 | PERFORMANCE EVALUATION

In this section, we have developed our protocol using omnet++ simulator[36] to evaluate the efficiency of the proposed algorithm called M-SSBLA in comparison with SSBLA,[2] SSBDC,[31] and MIS[10] algorithms. We have focused on the following major aspects: (1) the number of super-peer, (2) ordinary-peers coverage, (3) capacity utilization, (4) super-peer selection overhead, (5) robustness against of super-peer failure. The parameters of the experiments are given in Table 1. Results are averaged over 10 experiments with $a = 0.1$, $b = 0.1$ as reward and penalty parameters in L_{RP} learning algorithm. Peers are located in a network area of 1000 m × 1000 m. The departure and arrival of peers from/to the network area are governed by a Poisson distribution. The movements of peers are based on the random waypoint model in Maltz et al.[37] The maximum speed of each peer is 0–5 m/s with a maximum 12-s stay, and all peers are able to move anywhere. The communication range and capacity of peers are in power-law distribution. In the power-law distribution, the probability $P[cn = x]$ that a node n has a capacity x, with x contained in a limited range $[1; c_{max}]$, is equal to x-α, where α is the distribution exponent.[38] The maximum capacity available within the network is called c_{max}.

5.1 | Experiment 1

This experiment is conducted to study the impact of learning on the performance of the network. The simulation results are shown in Figures 6 to 9. The results indicate that utilizing learning capabilities of the network of learning automata increases the number of super-peer selection messages in comparison with random selection at early rounds of the simulation, because the learning process requires enough time (rounds) to learn proper action, which leads to finding appropriate roles for the network peers. After round nine, the number of messages in learning-based algorithms decrease, but it increases for random selection. This means that the learning process may reduce the super-peer selection overhead in self-adaptive manner. Additionally, in M-SSBLA, we do not focus on peer's capacity. Simulation results also show that the capacity utilization is enhanced by the proposed algorithm (Figure 7). Furthermore, Figures 8 and 9 show that deploying the learning process led to cover more ordinary-peer with fewer super-peer in comparison with the proposed algorithm without learning capabilities.

TABLE 1 Simulation parameters

Parameters	Values
Simulation time	20 min
Maximum number of peers	200
Network area	1000 m × 1000 m
Maximum capacity (c_{max})	15
Maximum communication range	c_{max} * 10
Maximum speed of a peer	0–5 m/s
A parameter in power-law distribution	2
Λ parameter in Poisson distribution	1
A (reward parameter) of LA	0.5
B (penalty parameter) of LA	0.2
Channel type	Wireless channel
Network interface	Wirelessphy

FIGURE 6 Impact of learning on overhead in M-SSBLA

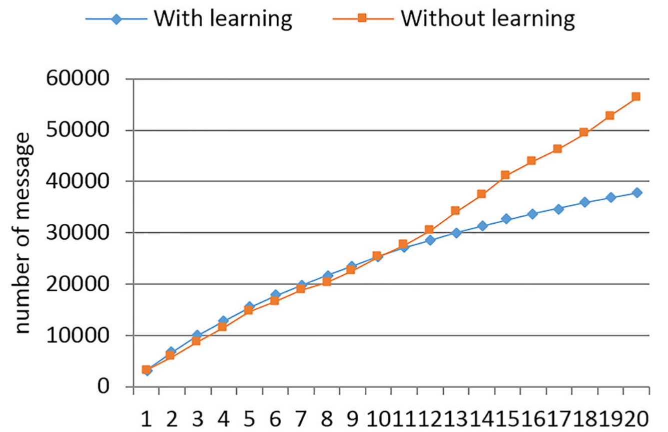


FIGURE 7 Impact of learning on capacity utilization in M-SSBLA

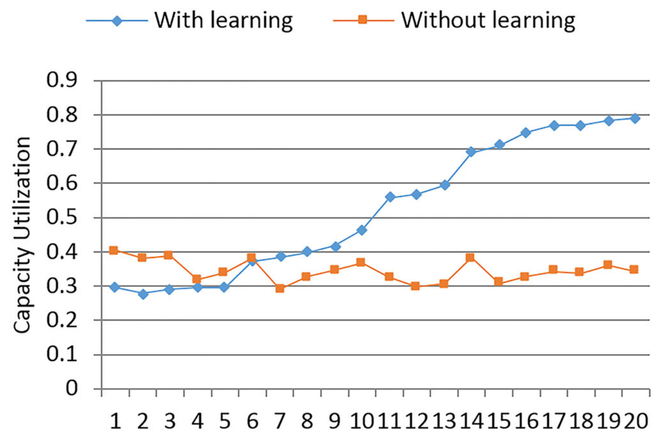
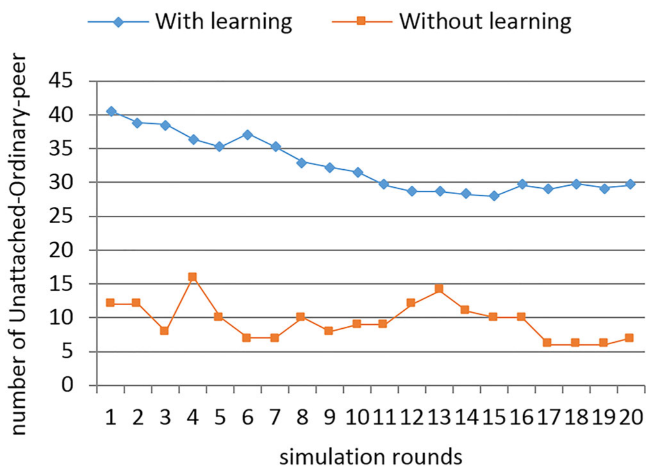


FIGURE 8 Impact of learning on unattached-ordinary-peer in M-SSBLA



5.2 | Experiment 2

This experiment is conducted to study the impact of reward and penalty parameters on the proposed algorithm. From Figures 10 to 13, L_{RP} and $L_{\epsilon p}$ models are evaluated with different values. According to the simulation results, utilizing L_{RP} model with $a = 0.1$ and $b = 0.1$ leads to better result with respect to evaluation parameters. Therefore, we set $a = 0.1$ and $b = 0.1$ respectively as reward and penalty parameters in the rest of this section.

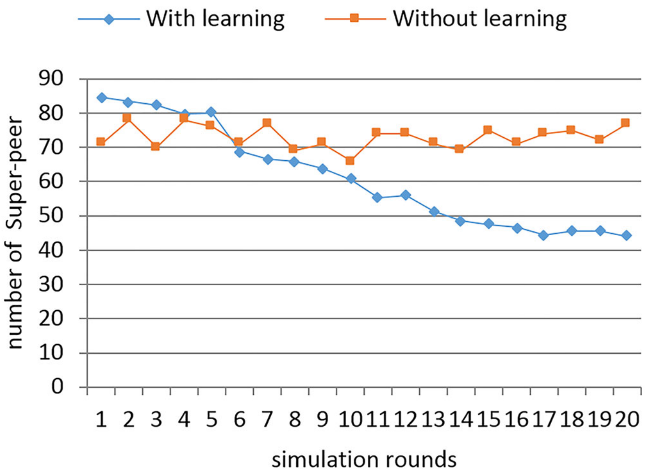


FIGURE 9 Impact of learning on super-peer in M-SSBLA

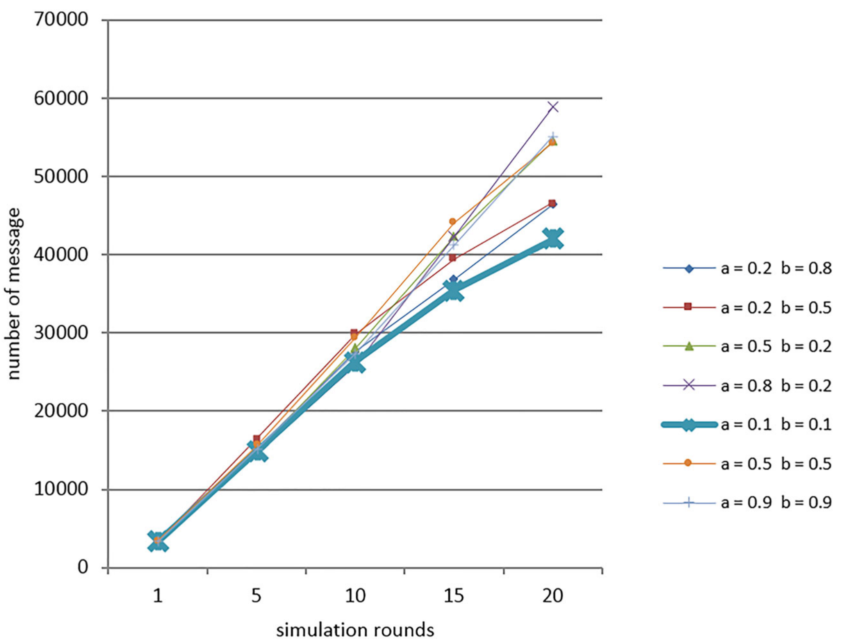


FIGURE 10 Impact of learning automata reward and penalty parameters on overhead in M-SSBLA

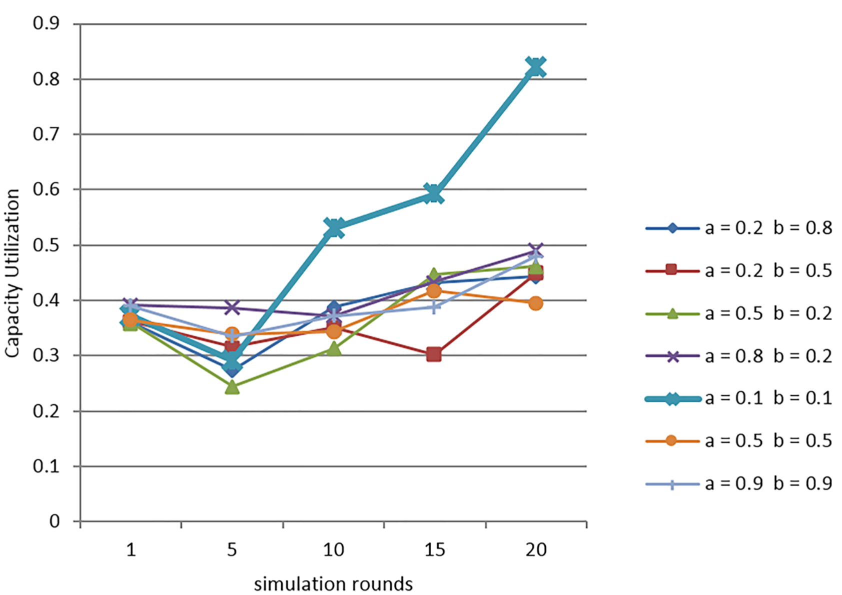


FIGURE 11 Impact of learning automata reward and penalty parameters on capacity utilization in M-SSBLA

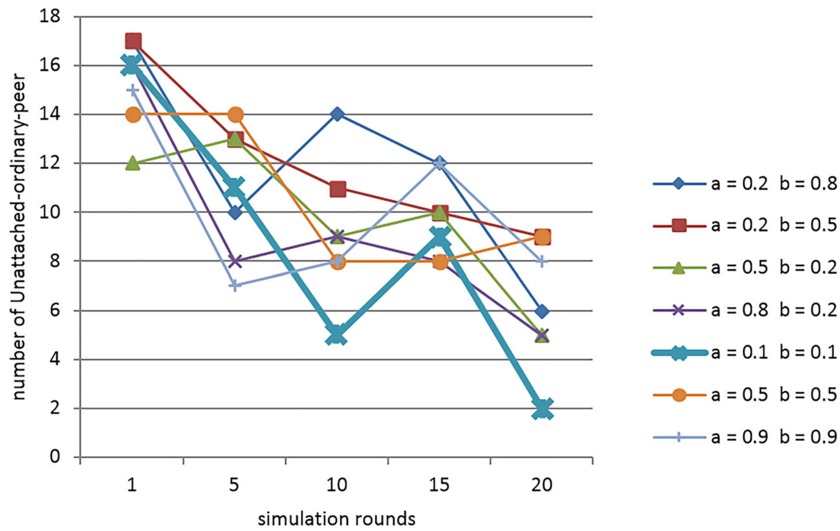


FIGURE 12 Impact of learning automata reward and penalty parameters on number of unattached-ordinary-peers in M-SSBLA

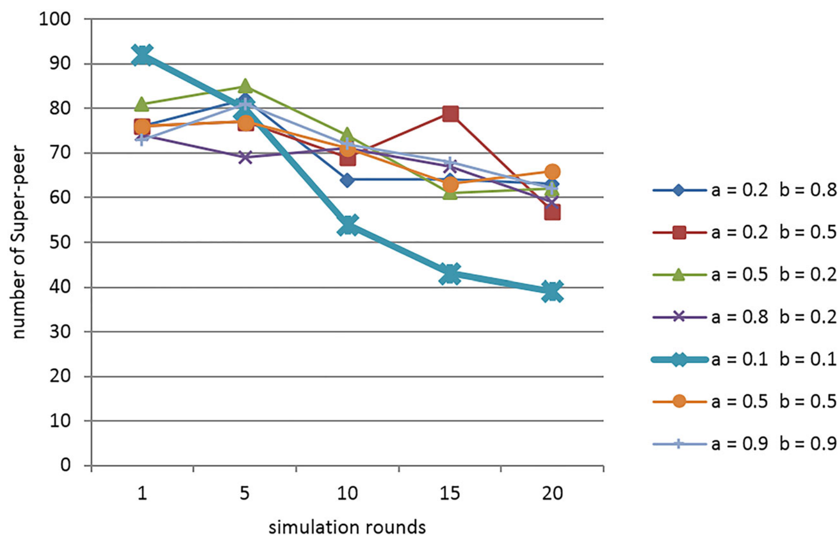


FIGURE 13 Impact of learning automata reward and penalty parameters on number of super-peers in M-SSBLA

5.3 | Experiment 3

This experiment is conducted to study the number of super-peers and ordinary-peers. One of the purposes of super-peer selection algorithms is to find a super-peers set with minimum cardinality to cover maximum ordinary-peer according to network conditions. A low number of super-peers along with high coverage for ordinary-peer means that the algorithm performs well. Capacity utilization factor is used to study the maximum number of ordinary-peers that a peer can handle as super-peer. Capacity utilization (*CU*) is defined as the ratio of the number of attached ordinary-peer to total capacity provided by super-peers, as given in 3 (let S denotes the number of selected super-peers).

$$\text{Capacity utilization} = \frac{\text{Number of attached ordinary-peer}}{\sum_{I=1}^{S} \text{Capacity(I)}}. \tag{3}$$

In this experiment, all parameters have been assigned under Section 5 and Table 1. The simulation result in Figures 14–16 illustrates the impact of the proposed algorithm on the number of super-peers and ordinary-peers. In the proposed algorithm, each peer learns about its super-peer accessibility using its learning algorithm. The proposed algorithm continuously checks available super-peers and switches to an appropriate accessible super-peer before dealing with the lack of existing super-peer. As shown in the simulation results, the number of super-peers in *MIS* is similar to

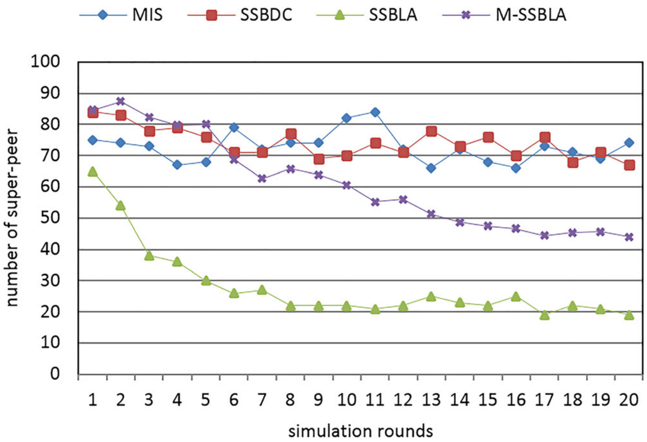


FIGURE 14 Number of selected super-peers as simulation proceeds

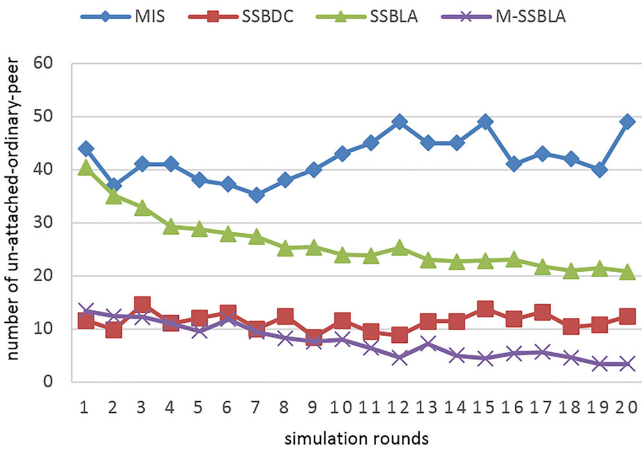


FIGURE 15 Number of unattached ordinary-peers as simulation proceeds

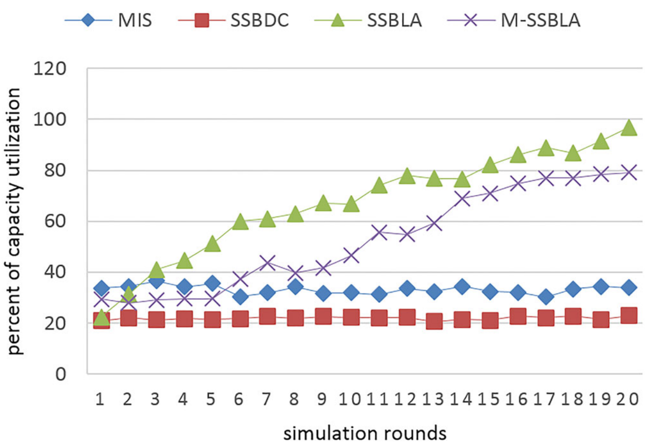


FIGURE 16 The capacity utilization during the operation of the network

SSBDC. By appending learning automata network to *MIS*, the number of selected super-peers reduces during operation of the network. The capacity utilization in SSBLA is higher than M-SSBLA, SSBDC, and MIS, and the number of super-peers in SSBLA is lower than other algorithms. Figure 15 shows that the number of unattached ordinary-peer changes during the simulation progresses in M-SSBLA. In the proposed algorithm the number of unattached ordinary-peer is less much than that other algorithms. As the figure reveals, the utilization of the learning automata network in the *MIS* leads to reducing the number of unattached ordinary-peers in this algorithm.

5.4 | Experiment 4

This experiment is conducted to study the overhead of the proposed algorithm. The number of messages transmitted among peers during super-peer selection process is considered as overhead of the proposed algorithm. In this experiment, all parameters have been assigned under Section 5 and Table 1. As shown in Figure 17, the overhead of M-SSBLA and SSBLA is similar to MIS and less than SSBDC until round eight. After round eight, overhead of M-SSBLA is reduced gradually in comparison with two other algorithms. M-SSBLA had more overhead in comparison with SSBLA because M-SSBLA continually sends messages to super-peers in its communication range, but SSBLA exchanges messages when a new peer arrives to its communication range. Both SSBDC and MIS do not lead to an overhead reduction during operation of the network.

5.5 | Experiment 5

This experiment is conducted to study the robustness of the proposed algorithm. To evaluate the robustness of the proposed algorithm considering super-peers failure, we organized a scenario in which 50% of super-peers are removed at round 10. All parameters have been assigned under Section 5 and Table 1. Figures 18 to 21 show the performance of the network. After early rounds of the simulation, some of ordinary-peers that their super-peer have been crashed and become unattached, and the algorithms behave as usual. Algorithms repair the overlay topology by selecting new super-peers among the peers. Figures 18 and 19 show the number of remaining super-peers and the number of unattached ordinary-peers, respectively. Since the number of selected super-peer in MIS and SSBDC is high, more than half of peers are demoted to unattached ordinary-peer. According to the results, one may conclude the following. M-SSBLA is more robust than SSBLA because M-SSBLA avoids repetition of the super-peer selection and helps to maintain

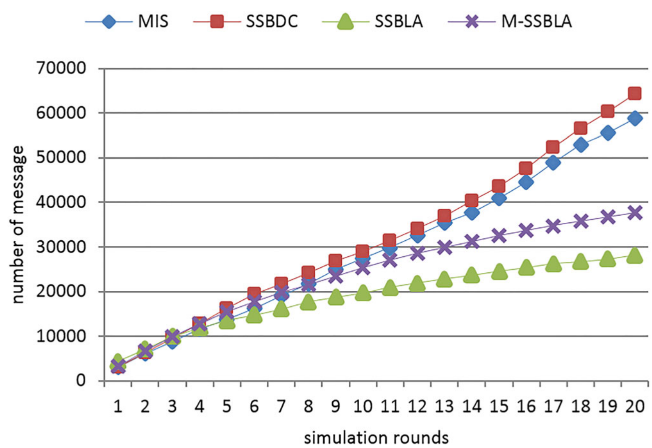


FIGURE 17 The overhead of the algorithms

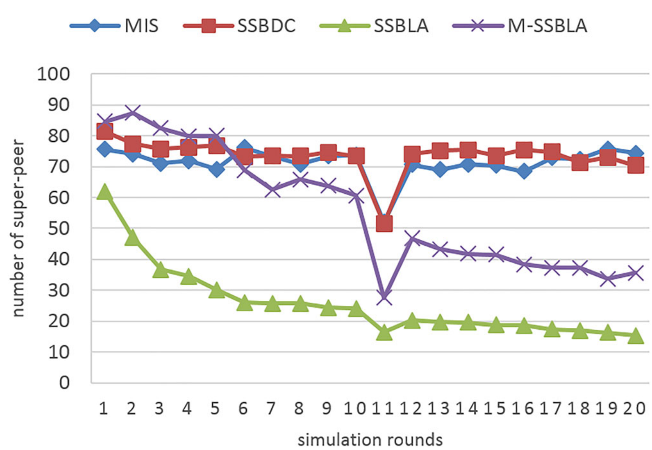


FIGURE 18 Number of super-peers in the face of 50% super-peers' failure

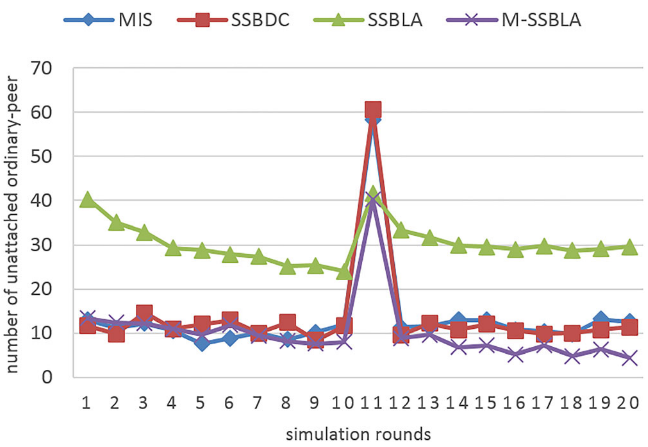


FIGURE 19 Number of unattached ordinary-peers in the face of 50% super-peers' failure

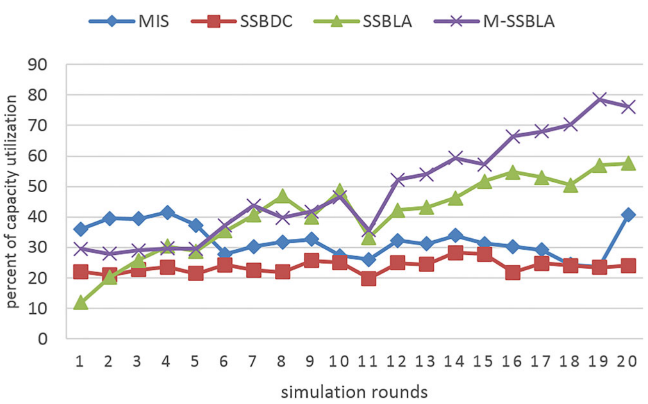


FIGURE 20 Capacity utilization in the face of 50% super-peers' failure

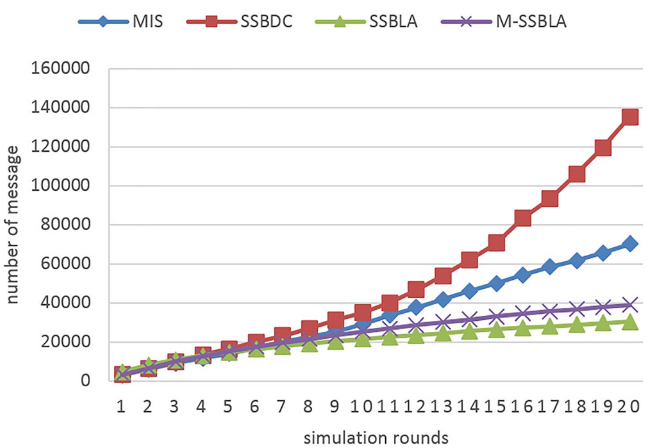
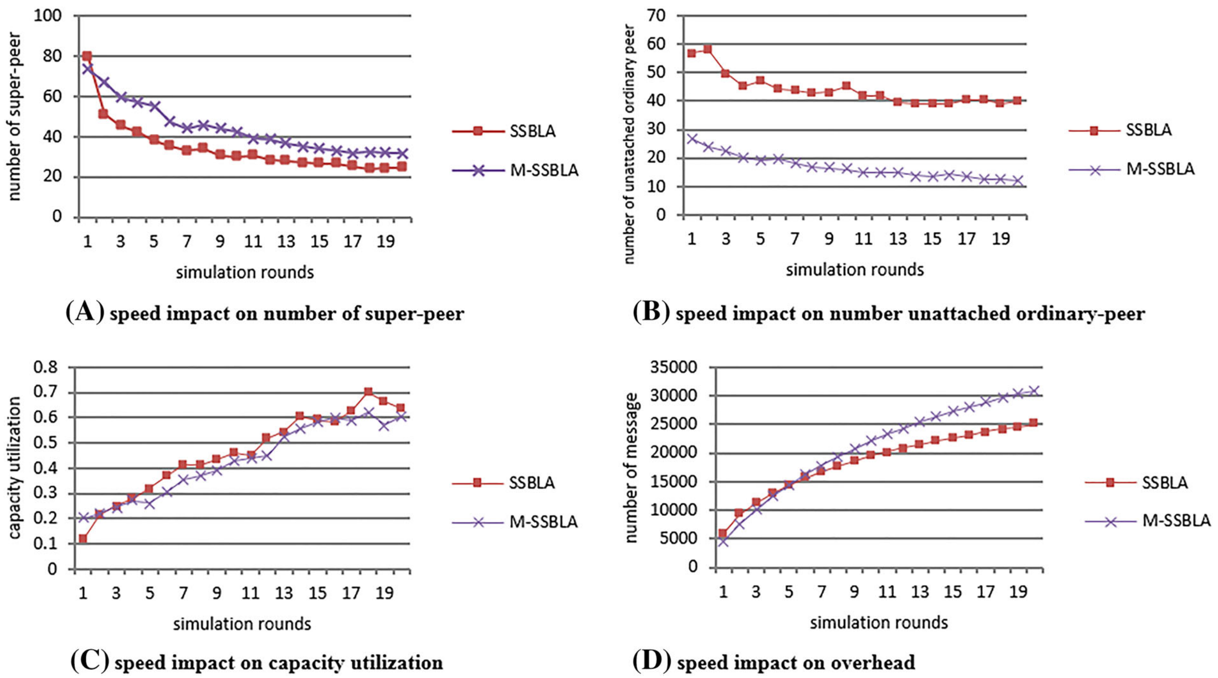


FIGURE 21 The overhead in the face of 50% super-peers' failure

network stability by switching from weak or lacked super-peer to more accessible super-peer. Figure 20 shows the effect of crash on the capacity utilization. As shown in this figure, M-SSBLA capacity utilization in the face of super-peer failure is more than maximum capacity utilization obtained by MIS and SSBDC. M-SSBLA and SSBLA almost have same cost but M-SSBLA performance with respect to other three factor is appropriate. All in all, the results show that the proposed algorithm is more robust than other algorithms in presence of super-peers' failure.

5.6 | Experiment 6

This experiment is conducted to study the mobility of the peer on the performance of the network. In order to evaluate the impact of speed on the network parameters, the proposed algorithm is executed for a network with 5–10 m/s speed



(A) speed impact on number of super-peer

(B) speed impact on number unattached ordinary-peer

(C) speed impact on capacity utilization

(D) speed impact on overhead

FIGURE 22 (A) Speed impact on number of super-peer. (B) Speed impact on number unattached ordinary-peer. (C) Speed impact on capacity utilization. (D) Speed impact on overhead. Impact of peers speed 5–10 m/s on M-SSBLA in comparison with SSBLA

for the peers, and the results are compared with SSBLA. Other parameters have been set based on the configurations mentioned in Section 5 and Table 1. Increasing speed of peers led to rising topology dynamicity and reducing stability of connections of peers. Based on the results, by speed increasing, a number of unattached-ordinary-peers is raised, and subsequently, capacity utilization is reduced. As the number of unattached-ordinary-peers increases, the number of control messages to select a new super-peer increases as consequence of learning process for finding accessible super-peers. According to the simulation results (Figure 22A–D), M-SSBLA performs better than SSBLA in the presence of dynamicity of peers with respect to coverage of unattached ordinary peers.

6 | CONCLUSION

This paper presents a self-adaptive super-peer selection algorithm in CMP2P network considering the mobility of peers. The proposed algorithm utilizes the capability of a network of learning automata over the MIS algorithm to learn either a super-peer is an accessible super-peer in the ordinary-peer communication range or not. This function takes into account mobile network dynamicity. The proposed algorithm is designated to cover maximum ordinary-peers and avoid lacking super-peers. According to the simulations, the proposed algorithm provides more coverage for whole ordinary-peers by fewer super-peer while reducing the maintenance overhead in comparison with other algorithms. Moreover, the proposed algorithm increases capacity utilization and robustness against failures. From another perspective, the proposed algorithm enhance MIS to find accessible super-peers in a self-adaptive manner.

ORCID

Nahid Amirazodi https://orcid.org/0000-0002-7135-5920

Ali Mohammad Saghiri https://orcid.org/0000-0003-0797-314X

Mohammad Reza Meybodi https://orcid.org/0000-0003-3775-5565

REFERENCES

1. Saghiri AM, Meybodi MR. An approach for designing cognitive engines in cognitive peer-to-peer networks. *J Netw Comput Appl*. 2016;70:17-40.
2. Amirazodi N, Saghiri AM, Meybodi M. An adaptive algorithm for super-peer selection considering peer's capacity in mobile peer-to-peer networks based on learning automata. *Peer-to-Peer Netw Appl*. 2018;11:74-89.

3. Mahdy AM, Deogun JS, Wang AJ. "A dynamic approach for the selection of super peers in ad hoc networks" presented at the the Sixth International Conference on Networking (ICN'07), Martinique; 2007:18. https://doi.org/10.1109/ICN.2007.1
4. Elgazzar K, Ibrahim W, Oteafy S, Hassanein HS. RobP2P: a robust architecture for resource sharing in mobile peer-to-peer networks. *Procedia Comput. Sci.* 2013;19:356-363.
5. Guo F, Xu J. Searching methods of MP2P based on diffusion strategy of resources index. *Int J Comput Netw Info Security*. 2011;3:26-33.
6. Rahmani M, Benchaïba M. PCSM: an efficient multihop proximity aware clustering scheme for mobile peer-to-peer systems. *J Ambient Intell Humaniz Comput*. 2018:1-18.
7. Shah N, Abid SA, Qian D, Mehmood W. A survey of P2P content sharing in MANETs. *Comput Electr Eng*. 2017;57:55-68.
8. Kim S-K, Lee K-J, Yang S-B. An enhanced super-peer system considering mobility and energy in mobile environments. In International Symposium on Wireless and Pervasive Computing, Hong Kong; 2011:1-5. https://doi.org/10.1109/ISWPC.2011.5751333
9. Gholami S, Meybodi MR, Saghiri AM. A learning automata-based version of SG-1 protocol for super-peer selection in peer-to-peer networks. In: *Recent Advances in Information and Communication Technology*. Springer; 2014:189-201.
10. Han J-S, Lee K-J, Song J-W, Yang S-B. Mobile peer-to-peer systems using super peers for mobile environments. 2008 International Conference on Information Networking, Busan; 2008:1-4. https://doi.org/10.1109/ICOIN.2008.4472831
11. Najim K, Poznyak AS. Basic Notions and Definitions,. *Learning Automata: Theory and Applications*. Elsevier; 1994:6-18.
12. Narendra KS, Thathachar MA. *Learning Automata: An Introduction*. New York, NY: Dover; 2012:784.
13. Torkestani JA, Meybodi MR. An intelligent backbone formation algorithm for wireless ad hoc networks based on distributed learning automata. *Comput Netw*. 2010;54(5):826-843.
14. Safavi S, Meybodi MR, Esnaashari M. Learning automata based face-aware mobicast. *Wirel Pers Commun*. 2014;77(3):1923-1933.
15. Misra S, Chatterjee SS, Guizani M. Stochastic learning automata-based channel selection in cognitive radio/dynamic spectrum access for WiMAX networks. *Int J Commun Syst*. 2015;28:801-817.
16. Krishna PV, Misra S, Joshi D, Gupta A, Obaidat MS. Secure socket layer certificate verification: a learning automata approach. *Sec Commun Netw*. 2014;7:1712-1718.
17. Kumar N, Lee J-H. Collaborative-learning-automata-based channel assignment with topology preservation for wireless mesh networks under QoS constraints. *IEEE Syst J*. 2014;9:675-685.
18. Kumar N, Lee J-H, Rodrigues JJ. Intelligent mobile video surveillance system as a Bayesian coalition game in vehicular sensor networks: learning automata approach. *IEEE Trans Intell Transp Syst*. 2014;16:1148-1161.
19. Kumar N, Misra S, Obaidat MS, Rodrigues JJ, Pati B. Networks of learning automata for the vehicular environment: a performance analysis study. *IEEE Wirel Commun*. 2014;21:41-47.
20. Kumar N, Misra S, Obaidat MS. Collaborative learning automata-based routing for rescue operations in dense urban regions using vehicular sensor networks. *IEEE Syst J*. 2014;9:1081-1090.
21. Saghiri AM, Meybodi MR. A distributed adaptive landmark clustering algorithm based on mOverlay and learning automata for topology mismatch problem in unstructured peer-to-peer networks. *Int J Commun Syst*. 2017;30:1-22, e2977.
22. Saghiri AM, Meybodi MR. A self-adaptive algorithm for topology matching in unstructured peer-to-peer networks. *J Netw Syst Manag*. 2016;24:393-426.
23. Polatoglou M, Nicopolitidis P, Papadimitriou GI. On low-complexity adaptive wireless push-based data broadcasting. *Int J Commun Syst*. 2014;27:194-200.
24. Nicopolitidis P, Chrysostomou C, Papadimitriou GI, Pitsillides A, Pomportsis AS. On the efficient use of multiple channels by single-receiver clients in wireless data broadcasting. *Int J Commun Syst*. 2014;27:513-520.
25. Nicopolitidis P. Performance fairness across multiple applications in wireless push systems. *Int J Commun Syst*. 2015;28:161-166.
26. Misra S, Krishna PV, Saritha V, Obaidat MS. Learning automata as a utility for power management in smart grids. *IEEE Commun Mag*. 2013;51:98-104.
27. Hasanzadeh M, Meybodi MR. Distributed optimization grid resource discovery. *J Supercomput*. 2015;71:87-120.
28. Morshedlou H, Meybodi MR. Decreasing impact of sla violations: a proactive resource allocation approachfor cloud computing environments. *IEEE Trans Cloud Comput*. 2014;2:156-167.
29. Seet B-C. Mobile P2Ping: a super-peer based structured P2P system using a fleet of city buses, presented at the IEEE3th International Conference on Pervasive Computing and Communications Workshops, Kauai Island, Hawaii; 2004.
30. Tang B, Zhou Z, Kashyap A, Chiueh T-C. An Integrated Approach for P2P file Sharing on Multi-Hop Wireless Networks. IEEE International Conference on Wireless And Mobile Computing, Networking And Communications, IEEE: WiMob'2005; 2005;3:268-274.
31. Bin L, Huan XY, Yan Z, Ting CB, Jian W. The algorithm of super-peer selection based on dynamic performance in mobile peer-to-peer network, presented at the EICE2012, Macau, China; 2012.
32. Tsai MG, Tzu-Chieh J. Multicluster, mobile, multimedia radio network. *J Wireless Netw*. 1995;1:255-265.
33. Wang TI, Tsai KH, Lee YH. Crown: an efficient and stable distributed resource lookup protocol distributed resource lookup protocol. In: Yang LT (eds). Embedded and Ubiquitous Computing. EUC 2004. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer; 2004:3207. https://doi.org/10.1007/978-3-540-30121-9_103
34. Wei Y, Xie G, Li Z. A hierarchical cross-layer protocol for group communication in MANET. 2007 IEEE International Conference on Telecommunications and Malaysia International Conference on Communications, Penang; 2007:62-67. https://doi.org/10.1109/ICTMICC.2007.4448610

35. Lee K-J, Choi J-H, Yang S-B. Fuzzy inference-based super peer selection for a practical double-layered mobile peer-to-peer system. *Adhoc Sensor Wireless Netw*. 2014;21:327-351.
36. Varga A, Hornig R. An overview of the OMNeT++ simulation environment. In Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops, 2008:60.
37. Josh B, Maltz DA, Johnson DB, Hu Y-C, Jetcheva J. A performance comparison of multi-hop wireless ad hoc network routing protocols. In Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking, 1998:85-97.
38. Montresor A. A robust protocol for building superpeer overlay topologies. In Proceedings. Fourth International Conference on Peer-to-Peer Computing, 2004. Proceedings, Zurich, Switzerland; 2004:202-209. https://doi.org/10.1109/PTP.2004.1334948

How to cite this article: Amirazodi N, Saghiri AM, Meybodi MR. A self-adaptive algorithm for super-peer selection considering the mobility of peers in cognitive mobile peer-to-peer networks. *Int J Commun Syst*. 2020; e4661. https://doi.org/10.1002/dac.4661