

A Novel Scheduling Mechanism Based on Artificial Immune System for Communication between Cluster Head and Cluster Members in WSNs

Arash Nikdel, Seyed Mahdi Jameii, and Hagar Noori

Abstract—Till now, different mechanisms have been proposed for communication between cluster members and cluster head in hierarchical protocols. Most of them determine a same time for each cluster member to communicate with cluster head without regarding the nodes conditions. In this paper, we propose a novel scheduling mechanism based on Artificial Immune System (AIS) called AISSM. In this mechanism, the nodes with more information have a better chance for communicating with cluster head. In addition, the proposed mechanism provides the proper nodes distribution in each cluster using virtual clustering concept. We have simulated LEACH protocol and used proposed scheduling mechanism in it (LEACH_{AISSM}) and then compared it with original LEACH protocol which uses TDMA scheduling mechanism (LEACH_{TDMA}). The results of simulation show the effectiveness of the proposed mechanism.

Index Terms—Wireless sensor network; artificial immune system; leach protocol; scheduling mechanism.

I. INTRODUCTION

The wireless sensor network (WSN) has emerged as a promising tool for monitoring the physical world [1]. One of the important issues on the field of sensor networks is routing. In [2], the routing protocols have been classified in four general categories such as Data-centric, hierarchical, location-aware and QoS Aware. In hierarchical routing, the nodes are divided into clusters. In each cluster, one node is considered as cluster head and other nodes are cluster members. Cluster members receive information from the environment and then send them to the cluster head. After that, the cluster head send the information to the sink node. Most of the hierarchical routing protocols consist of two steps. The first step is selecting the cluster head and the second one is routing. Hierarchical routing is a useful method in order to decrease the number of messages sending to the base stations and also to increase the lifetime of the network.

Different methods are used in hierarchical protocols in order to avoid interference between the data of cluster members. The most used method is Time-Division Multiple Access (TDMA) that is used in many protocols [3] [11]. TDMA method doesn't distinguishes between cluster member with low data and cluster member with high data and gives both of them the same chance to communicate with the

cluster head. While a member of cluster maybe located in a dynamic area and should have more time to send information to it in comparison with another members.

In this paper, we propose a scheduling mechanism based on Artificial Immune System (AIS) algorithm called AISSM for communication between a cluster members and cluster head.

The remaining of this paper is organized as follow: original LEACH protocol is explained in section 2. Artificial Immune System as a basic learning strategy used in the proposed protocol is discussed in section 3. Proposed mechanism is explained in section 4. Improved version of our mechanism is presented in section 5. Simulation results are shown in section 6. Section 7 is the conclusion.

II. LEACH PROTOCOL

LEACH [9] is an important well known hierarchical routing protocol. The operation of LEACH is broken up into rounds, where each round begins with a set-up phase, when the clusters are organized, followed by a steady-state phase, when data transfers to the base station occur. Initially, each node chooses a random number between 0 and 1. If the number is less than a threshold which is obtained by (1), the node becomes a cluster head for the current round. Cluster head for the current round broadcasts an advertisement message to the rest of the nodes. The non-cluster head nodes must keep their receivers on during this phase of set-up to hear the advertisements of all the cluster head nodes and decides the cluster to which it will belong for this round. The cluster-head node creates a TDMA schedule telling each node when it can transmit and broadcast back to the nodes in the cluster. Once the clusters are created and the TDMA schedules fixed, data transmission can begin.

$$T(n) = \begin{cases} P / (1 - P \times (r \bmod (1/P))) & \text{If } n \in G \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

The time slices are assigned to cluster members based on TDMA mechanism. At each time slice, cluster head communicates with one of its members and receives the information from it. After some time slices, the cluster head sends the received information to the sink node. For distribution the load among different nodes, at the end of each round the cluster heads are changed based on mentioned mechanism.

Manuscript received February 19, 2012; revised April 24, 2012.

The authors are with the Department of Computer Engineering, Islamic Azad University, Shoushtar branch, Shoushtar, Iran (e-mail: a.nikdel@khouzestan.srbiau.ac.ir, Jameii@Qodsiau.ac.ir, Hagar_noori@yahoo.com).

III. ARTIFICIAL IMMUNE SYSTEM MODELS

AIS are distributed adaptive systems for problem solving using models and principles derived from the Human Immune System. The capabilities of the NIS is mainly the inner working and cooperation between the mature T-Cells and B-Cells that is responsible for the secretion of antibodies as an immune response to antigens. The various theories with regards to the functioning and organizational behavior of the natural immune system (NIS), is discussed. These theories inspired the modeling of the NIS into an Artificial Immune System (AIS) for application in non-biological environments. Many different AIS algorithm models have been built, including Classical View Models, Clonal Selection Theory Models, Network Theory Models and Danger Theory Models. Artificial immune systems have been successfully applied to many problem domains. Some of these domains range from network intrusion and anomaly detection, to data classification models, virus detection, concept learning, data clustering, robotics, pattern recognition and data mining [12].

IV. PROPOSED SCHEDULING MECHANISM

In this section, we describe the proposed mechanism called AISSM. This mechanism is employed after clustering phase in order that cluster heads collect information from their members. In this mechanism, each cluster head selects the members to communicate with them using AIS algorithm. In the proposed mechanism, we suppose that if one cluster member has a packet to send at the moment t , it will have another one to send very likely at the moment $t+1$ and vice versa. For this reason, if the selected member has data to send, it will be more probable to select it in the next time slices and otherwise, selection probability will be decreased. The proposed algorithm keeps the B bit history of communications with each node. We present this history by sequence of bits and called it case history. In case history, 0 means that node doesn't have any data to send and 1 means that node has data to send. At first, after performing the clustering phase, all case histories are considered equal to 0 as shown in Fig. 1.

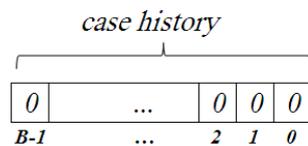


Fig. 1. B bit case history for each node

Whenever the cluster head communicates with a node, it updates the case history of that node. For this purpose, all bits are shifted to the right and the least significant bit is removed and the value of the most significant bit is calculated by f_r function. This process is shown in Fig. 2.

$$\begin{aligned} &\text{If the node doesn't have any data to send} \\ &f_r = 0 \\ &\text{Else} \\ &f_r = 1 \end{aligned} \quad (2)$$

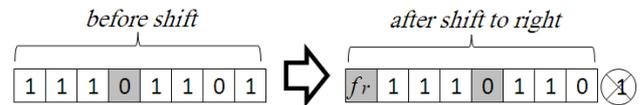


Fig. 2. Process of updating the case history

In each round of the proposed algorithm, the cluster head selects one of the cluster members. The probability of selecting each node is obtained by (3):

$$p_i = \text{affinity}_{node} / \sum_{j=1 \text{ to } N} \text{affinity}_{node-j} \quad (3)$$

In (3), the affinity value of node is calculated using case history of node according to (4).

$$\text{affinity}_{node} = \lambda + \sum_{b=0 \text{ to } B-1} (2^{b-1} \times a^{b-1}) \quad (4)$$

In (4), it is supposed that the node case history is B bit ($a_{B-1}, \dots, a_2, a_1, a_0$). Also λ is a constant number in order that the node affinity value does not exceed a determined level. In order that the affinity of a node with the most affinity value in maximum case becomes ζ times more than the node with minimum affinity value, the parameter λ is determined according to (5) to (8) as follows:

$$\text{affinity}_{node-min} = \lambda \quad (5)$$

$$\text{affinity}_{node-max} = \lambda + 2^{B-1} \quad (6)$$

$$\zeta = \text{affinity}_{node-max} / \text{affinity}_{node-min} = (\lambda + 2^{B-1}) / \lambda \quad (7)$$

$$\lambda = (2^{B-1}) / (\zeta - 1) \quad (8)$$

After selecting one node, the cluster head requests the selected member to send data packet. If the cluster member has packet to send, it will send it to the cluster head and otherwise, it will not respond. Then, the cluster head regarding that node has had data to send or not, updates the case history of mentioned node. The process of updating the case history was discussed before. In this way, the probability of selecting node in future becomes more or less.

In the proposed mechanism, the nodes with affinity value more than threshold will be selected as memory cells and have some quota. It means that after the specified number of time slices (η), the cluster head will communicate with them surely. In fact, they are given a chance to resend data. The affinity threshold is calculated according to (9).

$$\text{affinity}_{threshold} = \lambda + \sum_{b=R \text{ to } B-1} (2^{b-1}) \quad (9)$$

In order to give quota to the nodes selected as memory cells, the cluster head in some constant time rounds (after η time slice), determines μ time slice to communicate with the nodes selected as memory cells. The amount of μ is calculated by (10).

$$\mu = |M| \quad (10)$$

where $|M|$ represents the number of nodes selected as memory cells and μ is the number of time slices determined to communicate with the nodes selected as memory cells.

The only operation in AIS algorithm is mutation. We

suppose that the mutation rate of each node (τ_{node}) has reverse impact on its affinity value ($affinity_{node}$). τ_{node} is calculated for each node according to (11).

$$\tau_{node} = \psi / (affinity_{node} + \epsilon) \quad (11)$$

where ψ is a constant number that is calculated in the way that mutation rate doesn't become less than the determined level. Also ϵ is a constant number that should be selected properly in order that mutation rate doesn't exceed the determined level.

In the proposed mechanism, the cluster head after some time slices (σ), selects some of nodes randomly (∂ percent of nodes) and mutates the case history of nodes regarding their mutation rate. Fig. 3 shows a binary number with four different mutation rates.

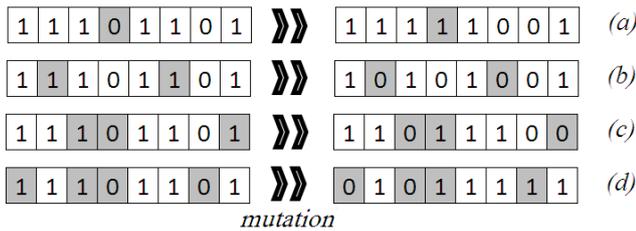


Fig. 3. A binary number with four different mutation rates

Regarding the binary values before and after mutation, it is clear that this operation makes smaller number bigger and big number smaller. When we have more mutation rate, then the change will be greater. Also, the nodes with less affinity have more mutation and vice versa.

In the proposed mechanism, the only way to give more chance to those nodes which had not any data to send in the past rounds and with low affinity value, is mutation. While it is possible that such nodes aren't given any chance to communicate with cluster head in some successive rounds.

In order to overcome this challenge, after each $\mu + \eta$ time slice, the δ number of time slices spent on communicating with α percent of nodes that their affinity are less than the $affinity_{threshold-below}$. The value of $affinity_{threshold-below}$ is calculated according to (12), Also the value of δ is calculated according to (13). Note that η and μ time slice is for communicating with cluster nodes and also with the nodes selected as memory cells.

$$affinity_{threshold-below} = \lambda + \sum_{b=0 \text{ to } B-U} (2^{b-1}) \quad (12)$$

$$\delta = \alpha \times |C| \quad (13)$$

where δ represents the number of time slices are allocated to communicate with such nodes, $|C|$ represents number of nodes with affinity value less than $affinity_{threshold-below}$. α is percent of nodes which are communicated and are given chance to send data. It is obvious that α will be between 0 and 1. The cluster head repeats these tasks after n time slices. The value of n is calculated according to (14). These repetitions are shown in Fig. 4.

$$\eta + \mu + \delta = n \quad (14)$$

This sequence continues until the new round of clustering.

After the number of time slices, the cluster head sends received information of its cluster members to the sink node.

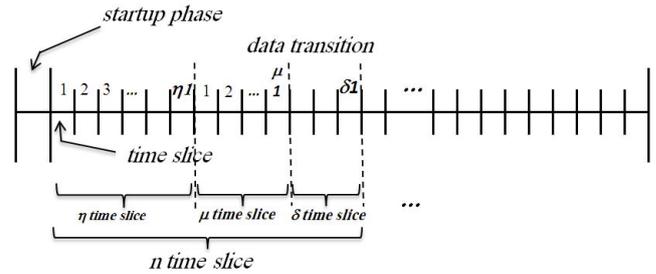


Fig. 4. The repetition of cluster head tasks

V. IMPROVING THE PROPOSED MECHANISM USING VIRTUAL CLUSTERING CONCEPT

In the proposed mechanism, there is no attention to the proper distribution of the selected nodes in the cluster. The nodes are selected only based on their affinity value. In order to improve the distribution of the selected nodes in cluster area, we propose the virtual clustering concept. In this method, it is supposed that each node is aware of its own location based on one of the existing methods. After performing clustering phase, the cluster head sends the special packet (please-register) to all nodes. This packet consists of geographic location of cluster head and also the number of virtual clusters. After receiving this message, each member calculates its virtual cluster number based on (15) to (18). After that, it informs the cluster head by sending a packet called "registerMe". This packet consists of its ID and Virtual Cluster number. This process is shown in Fig. 5.

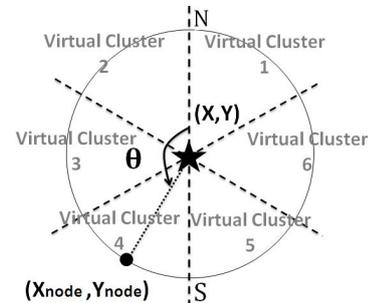


Fig. 5. Dividing a cluster to the virtual sub-clusters

$$\theta = \text{Sin}^{-1} ((x - x_{node}) / d) \quad (15)$$

$$d_{node} = ((x - x_{node})^2 + (y - y_{node})^2)^{1/2} \quad (16)$$

$$\Omega = 2\pi / r \quad (17)$$

$$VirtualCluster_{node} = \lceil \theta / \Omega \rceil \quad (18)$$

In these equations, (x_{node}, y_{node}) is node coordinates and (x, y) is cluster head coordinates, r is the number of virtual clusters, $VirtualCluster_{node}$ is virtual cluster number of which the node is belong to it. In Fig. 5, one cluster is divided to six virtual clusters, so $r=6$. If it is supposed that $\theta=150$, $\Omega=60$ and $VirtualCluster_{node}=3$ then as a result the node will be the member of third virtual cluster. After receiving all "registerMe" packets, the cluster head will be informed of the

number of all nodes in cluster and the number of nodes in each virtual cluster. After determining the number of existing nodes in each virtual cluster, the cluster head applies the AISSM algorithm to each virtual cluster separately and determines the number of necessary time slices in order to guarantee the quality of service. The cluster head calculates parameters for each virtual cluster separately according to (19) to (26).

$$N_1 + N_2 + \dots + N_r = \sum_{i=1 \text{ to } r} N_i = N \quad (19)$$

$$\eta_1 + \eta_2 + \dots + \eta_r = \sum_{i=1 \text{ to } r} \eta_i = \eta \quad (20)$$

$$\eta_i = (N_i / N) / \eta \quad (21)$$

$$\mu_i = |M_i| \quad (22)$$

$$\mu_1 + \mu_2 + \dots + \mu_r = \sum_{i=1 \text{ to } r} \mu_i = \mu = |M| \quad (23)$$

$$C_1 + C_2 + \dots + C_r = \sum_{i=1 \text{ to } r} C_i = C \quad (24)$$

$$\delta_1 + \delta_2 + \dots + \delta_r = \sum_{i=1 \text{ to } r} \delta_i = \delta \quad (25)$$

$$\alpha_i = (C_i / C) / \alpha \quad (26)$$

In the above equations r represents the number of virtual clusters, N represents the number of all nodes and N_i is the number of nodes exists in the i^{th} virtual cluster. η shows the number of all time slices allocated before giving quotas and η_i represents the number of time slices allocated to communicate for the i^{th} virtual cluster. μ is the number of all nodes selected as memory cells, μ_i is the number of existing nodes in the i^{th} virtual cluster selected as memory cells. M represents the number of existing nodes in memory cells set and M_i is number of nodes belonging to the i^{th} virtual cluster which also are in the memory cells set. Fig. 6 shows the time slices of this mechanism.

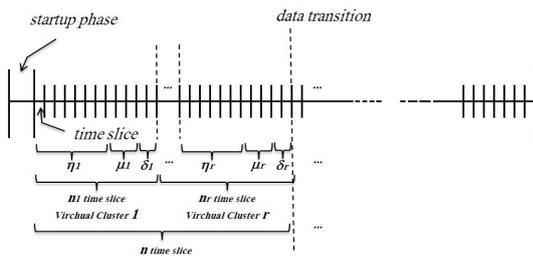


Fig. 6. Using virtual clustering to improve the node's distribution

VI. SIMULATION RESULTS

In this section, our proposed mechanism is simulated using NS2 simulator [13]. In order to evaluate it, we simulated LEACH protocol and used proposed scheduling mechanism in it (LEACH_{AISSM}) and then compared it with original LEACH protocol which uses TDMA scheduling mechanism (LEACH_{TDMA}). In order to perform simulation, 100 sensor nodes are distributed in an area equal to 100×100 square meters. The number of virtual clusters in our simulation is

considered equal to 4 ($r=4$). Other assumptions are as follow: $B=8$, $\eta=50$, $\alpha=0.05$, $\lambda=128$, $\zeta=3$, $\psi=128$, $\sigma=50$ and $\partial=50\%$. Each node that is selected as a cluster head will be responsible in 500 time slices. The cluster head puts the received data in a packet after 5 time slices and sends it to the sink node. The network is tested for loads (the average number of packets sent by each node in each time slice) 0.0125, 0.0250, 0.0375, 0.0500, 0.1000, 0.1500, 0.2000, 0.2500 and 0.3000. The simulation duration is 600 seconds and during simulation, the clustering step is performed several times.

A. The First Experiment

In this experiment, we show that the performance of proposed mechanism is improved. It means that the AISSM algorithm determines some members of cluster with more sending information. Thus, cluster head give them more chance to communicate with cluster heads. For this purpose, LEACH protocol with $P=0.077$ is employed using two scheduling mechanism. After clustering step based on LEACH protocol, the cluster heads begin to collect the data packets from their own cluster members using AISSM mechanism or TDMA. In this experiment, load parameter (i.e. the average number of packet sent by a node in each time slice) is considered 0.1. The results of experiment are shown in Fig. 7. Regarding 500 time slices for each cluster head to communicate with cluster members and also the number of received packet in various time slices, it is clear that in LEACH_{AISSM} protocol, the cluster head communicates with cluster members in more time slice and so, cluster head knows its members conditions better than before. Thus cluster head communicates with a member which has more information to send. Therefore, it receives more data packets from its members. While in LEACH_{TDMA} protocol, the cluster heads do not distinguish between their members.

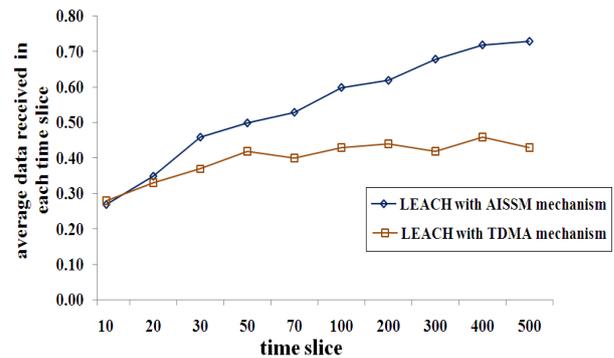


Fig. 7. Average data received in different time slices

B. The Second Experiment

In this experiment, the average of received data in each time slice of two protocols LEACH_{AISSM} and LEACH_{TDMA} for different loads with $P=0.077$ are evaluated. Fig. 8 shows the results of this experiment. As it is shown, the average of received data in each time slice in LEACH_{AISSM} is higher than LEACH_{TDMA}. This is because in LEACH_{AISSM}, the cluster head knows its active members by AISSM mechanism and so, these members are allocated more time to send information.

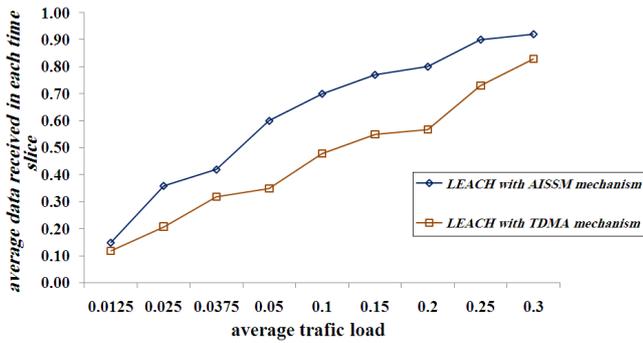


Fig. 8. Average data received in each time slice in different Traffic load

C. The Third Experiment

The purpose of this experiment is to evaluate the average time that each package waits for the proposed scheduling mechanism. For this purpose, LEACH_{AISSM} and LEACH_{TDMA} protocols with $P=0.077$ for different loads are employed and the average of waiting time for each package is calculated. Fig. 9 shows the results of this experiment.

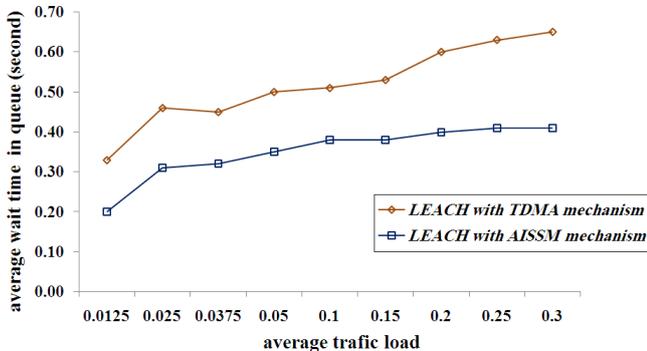


Fig. 9. The average waiting time for each packet in different traffic loads

The experiment results show that the performance of LEACH_{AISSM} is better in comparison with LEACH_{TDMA}. Regarding that in LEACH_{AISSM}, the active cluster members have more time to send information, they can send their data at a higher rate, therefore, the average of waiting time for their packets is decreased.

VII. CONCLUSION

In this paper, we proposed a scheduling mechanism based on AIS algorithm for communication between cluster heads and their members. In order to evaluate our mechanism, we simulated LEACH protocol and used proposed scheduling mechanism in it (LEACH_{AISSM}) and then compared it with original LEACH protocol which uses TDMA scheduling mechanism (LEACH_{TDMA}). The results of simulation show the effectiveness of the proposed mechanism.

REFERENCES

- [1] J. Jia, J. Chena, G. Changa, and Z. Tana, "Energy efficient coverage control in wireless sensor networks based on multiobjective genetic algorithm," Elsevier, *Computers and Mathematics with Applications*, pp. 1756-1766, 2009.
- [2] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," Elsevier *Ad Hoc Network Journal*, pp. 325-349, 2005.
- [3] A. Manjeshwar and D. Agrawal, "TEEN: A routing protocol for enhanced efficiency in wireless sensor networks," *1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, 2001.
- [4] A. Manjeshwar and D. Argrawal, "Apteen: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks," *IPDPS*, 2002.
- [5] M. Younis, M. Youssef and K. Arisha, "Energy-Aware routing in cluster-based sensor networks," in *proc. of the 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS2002)*, Fort Worth, Texas, 2002.
- [6] M. Younis, M. Youssef, and K. Arisha, "Energy-Aware management in cluster-based sensor Networks," *The International Journal on Computer Networks*, vol. 43, no. 5, pp. 649-668, 2003.
- [7] I. S. Misra, S. Dolui, and A. Das, "Enhanced-Efficient adaptive clustering protocol for distributed sensor networks," *ICON*, 2005.
- [8] T. Voig, A. Dunkels, J. Alonso, H. Ritter, and J. Schiller, "Solaraware clustering in wireless sensor networks. *Computers and Communications*," *ISCC, Ninth International Symposium on*, vol. 1, no. 28, 2004.
- [9] T. Huynh and C. S. Hong, "Prolonging network lifetime via intracluster routing in wireless sensor networks," in *proc. of ICMU2005*, pp. 162-167, 2005.
- [10] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energyefficient communication protocol for wireless microsensor network," in *proc. of the 33rd Hawaii International Conference on System Sciences (HICSS '00)*, January 2000.
- [11] R. Musunuri and J. A. Cobb, "Hierarchical-battery aware routing in wireless sensor networks," *Vehicular Technology Conference*, 2005.
- [12] P. Andries Engelbrecht, "Computational Intelligence: An Introduction," 2007.
- [13] The Network Simulator - ns-2. <http://www.isi.edu/nsnam/ns/>.



Arash Nikdel received the B.S. degree in Computer Engineering from the Islamic Azad University and the M.S. degrees in Computer Engineering from Ahvaz Science and Research Branch in Iran, in 2007 and 2011, respectively. His research interests include computer networks and evolutionary algorithms.



Seyed Mahdi Jameii is a Ph.D. Candidate in Software Engineering. He is currently a faculty member at the computer department of Shahr-e-Qods azad university. His research interests include wireless sensor networks, distributed systems and distributed databases



Hagar Noori received the B.S. degree in Computer Engineering from the Islamic Azad University and the M.S. degrees in Computer Engineering from Arak Science and Research Branch in Iran, in 2007 and 2012, respectively. His research interests include computer networks and evolutionary algorithms.