



Management of Channel and Bandwidth in Wireless Sensor Networks by Learning Automata based Algorithm

Farzad Kiani

Department of Computer Engineering, Engineering and Natural Sciences Faculty, Istanbul Sabahttin Zaim University, 34303, Kucukcekmece, Istanbul/Turkey
Farzad.kiani@izu.edu.tr

Abstract

The time synchronization and channel allocation are important research topics in the wireless sensor networks. In this paper, a scheduling mechanism based on learning automata is proposed for the association of cluster members with cluster headers. It was to management of network channel that can be shared between all node nodes. Learning automaton-based algorithms can be very suitable model for use in the sensor networks. Because they show such desirable performance in such distributed environments. In addition, they have good performance in the energy consumption because they behaviors based on its local information and do not have more computational overheads. In addition, the learning automata have the ability to adapt to changes in the environment and it is a very good option for these networks, which have a dynamic structure. In this paper, we implemented the proposed algorithm on the LEACH protocol and compared the result of proposed algorithm with TDMA based protocol. The simulation results are presented in two scenarios that in the received data rates the proposed algorithm is compared with two protocols in average received data in time and traffic load axis. It is observed that the proposed method gives better results than TDMA.

Keywords: wireless sensor networks, time scheduling, channel allocation, learning automaton.

1. INTRODUCTION

The Wireless Sensor Networks (WSNs) are different from the other wireless networks such as Mobile Ad-hoc Network so they are combining of large number of mini-size sensor nodes and a few Base Stations (BS) or sinks in a distributed platform. Each node is consists of sensing, processing, transceiver and power units. The nodes have low battery and limited processing unit [1]. In the network environment, sensor nodes sense events via sensing unit, collect and process their data via the processing unit, and transmit them to BS/neighbor nodes via transceiver unit. One of the reasons of development and progression of the WSNs is using the inexpensive, small, and affordable sensor nodes. Therefore, WSNs are used in many applications such as civil, medical, military, governmental and probability-based applications as volcano [2]. In addition, these networks have self-managing property so they can adapt in different conditions by various protocols. Thus, adaptive protocols implementation are an important issue in the

network design. For this reason, the resource limitation of sensor nodes are significant factor in the implementation phase. The protocols must use the processing and power units efficiently. So, the most researchers are concerned on these problems in the recent years as specially on the energy saving. Although, energy saving has tradeoff with some of the design factors such as reliability or system overhead, it is a need to maintain balance between all the factors [2]. In particular, it is an important justification and requirement in real-time systems.

These networks designers must have enough information about design factors, communication architecture and stack protocol of the WSNs in designing appropriate protocols. The design factors in the WSN are standard in the protocols implementation. In fact, they play a guideline role and designers can use them even to compare with other models. They are usually reliability, topology, scalability, power consumption, data reports, and transmission media. The WSN stack is consists of five layers and three planes. The layers are physical layer, data link layer, network layer, transport layer, application layer [3]. The planes are power management plane, mobility management plane and task management plane. Energy efficiency is debatable in all layers of protocol stack. For example, in [4] collision, packet overhead, latency, overhearing and idle listening are discussed and focus on their management to reach energy efficiency. Collisions must be control because they cause unnecessary transmission costs at the source and destination nodes. The collisions can are managed in the design phase by TDMA and so on protocols and in after the design by avoidance protocols. On the other hand, limitation in the processing unit can be solved by data compressing and processing for reducing the packets. However, this issue has been addressed in very few cases of researches. Many papers discus on the data reduction only for energy saving because almost of case studies do not have concern on beside of network. In particular, this requirement is very important when many packets of a data are sent in the network. It is also rarely possible scenarios in literature where the large volumes of data are transmit inside the system. Now in the age of new

technology, the data are talking. Therefore, the data analysis plays a very important role in various applications and its importance is increasing day by day especially when the data volumes are large.

We can realized many of network requires by a routing protocol. Classification of routing protocols in WSNs is useful to a designer in selecting the routing method in accordance with his fashion its application. Routing protocols in WSNs may be concern on various parameters such as QoS, high-speed data transferring, reliable data transferring, reducing packet loss and packet delay, increase network lifetime and optimizing overhead of network by consider on energy and resource limitations. The routing protocols in wireless sensor networks are generally in three categories as flat, hierarchical and location based protocols. The energy issue is one of the most fundamental issues in these networks. Hierarchy methods are more prominent because energy consumption is efficient in these protocols. In hierarchical based routing protocol, the sensor nodes are divided into logical clusters. In each cluster, one or more nodes are as cluster head (CH) node and other nodes are considered as cluster members. The number of CHs is usually defined one for each cluster, but in the literature, more than one is selected in some methods [1, 5]. The cluster members obtain the information based on using the environment and then send this information to the CH nodes. Then, they also collect this information and transmit to the base station (BS) node. Hierarchical based routing protocols use different methods to prevent interference in the relationship between cluster members and their CH node. The used mechanism normally to communicate cluster members with CH is TDMA, which has been used in many protocols [6-10]. TDMA protocol does not distinguish between nodes with and without data. Indeed, it gives the same time to both group cluster members to communicate with the CH node. It is possible a member of the cluster is placed in a region where its environmental information is constantly changing but the other cluster members do not feel such changes in their surroundings. In this case, a member whose local information is constantly changing should have more time wait to communicate with the cluster and send information to it than other members. Therefore, the channel allocation and management is important issue in these networks design.

In this paper, a channel management mechanism based on learning automata based algorithm is suggested to communicate between a cluster header and its members. In this method, each cluster head is equipped with a learning automaton, such that, the machine is responsible in controlling the relationship between the cluster heads and their members. Learning automata learns over time that members of the cluster who have

more information to send have a greater chance of connecting with the CH. The proposed mechanism can be implemented in different hierarchical protocols. In order to evaluate the advantages and disadvantages of the proposed mechanism, this mechanism is used in the LEACH protocol, which is the first and most prominent hierarchical routing protocol [6]. It is noteworthy that researchers have improved the relevant LEACH protocol over the past years, but none of them has focused on channel management issues and all of them follow the channel management approach of LEACH protocol. In the origin LEACH protocol, the duration of network activity is divided into periods (rounds). At the beginning of each period, a number of nodes are randomly selected as the cluster header. For this, each node generates a random number between 0 and 1. If this number is less than the defined threshold, then the node is introduced as the header. The chance of clustering for nodes is not repeated. After determining the nodes of the cluster, other nodes decide, based on the signal strength received from each cluster, to which cluster membership. It should be noted, however, that cluster joining is done in two ways. One of these methods is the formation of clusters, then choosing the head of clusters. In other approach, the CHs are selected after forming the clusters. The cluster node divides its responsibility into a number of time slots as is shown in figure 1. Cluster members based on the TDMA mechanism share these slots. At each slot, the cluster communicates with one of the cluster members and receives the information packets of that member. The cluster sends received data from its members to the BS node at some time intervals. In order to distribute loads on all nodes of clusters, the CHs are changed in each rounds.

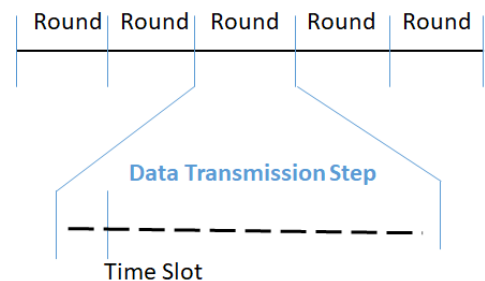


Fig. 1. Rounds and time slots in the LEACH protocol [6].

In the section 2 is introduced a channel management mechanism based on learning automata based algorithm. The results of simulations and evaluation are explained in the section 3. Last section presents the conclusion of the paper.

2. PROPOSED ALGORITHM

By studying the characteristics of learning automata [11] and the characteristics of sensor networks, it can be concluded that learning automata is a very suitable model for use in sensor networks. One of the reasons is that it works well in distributed environments. Distribution of the sensor network environment causes increasing the inclination to use algorithms that can be distributed and utilized by local information. This problem is especially important given the energy constraints in sensor nodes and the tendency to reduce the transmission of additional information to prevent energy dissipation [12]. In generally, the learning automaton based algorithms have appropriate performance in the distributed systems. The learning automata can be a good option for saving energy issue in the WSNs, because they have a simple structure and the very simple amplification signal that they get it from their surroundings [13]. On the other hand, the computational overhead of learning automata is very low.

In this section, the proposed channel management mechanism, called the CHMLA, is described. This mechanism is used after clustering and to collect information from cluster members. In the proposed mechanism, each cluster is equipped with a learning automaton. The cluster chooses the cluster member to communicate with using its learning automaton. In the proposed mechanism, it is assumed that if a cluster member has a packet for sending at the instant t , it is likely that there will also be a packet at the moment $t + 1$. For this reason, if the selected member has data to send, the chances of choosing it in the next steps will increase according to the Equation 1. Otherwise, it decreases according to Equation 2 [14].

$$p_i(n+1) = p_i(n) + \alpha [1 - p_i(n)]$$

$$p_j(n+1) = (1 - \alpha)p_j(n) \quad \forall j, j \neq i \quad (1)$$

$$p_i(n+1) = (1 - b)p_i(n)$$

$$p_j(n+1) = \frac{b}{r-1} + (1 - b)p_j(n) \quad \forall j, j \neq i \quad (2)$$

The number of operations of each learning automaton is equal to the number of cluster members corresponding to it. Selecting an action by the learning automaton means choosing a member of the cluster to communicate with the CH node. The vector of probability selection of the learning automata of each cluster is initialized in accordance with Equation 3. Where, n is the number of cluster members and i is the number of the action automata.

$$\forall i \quad i \leq n \quad P_i = \frac{1}{n} \quad (3)$$

The algorithm is executed as follows. First, the CH learning automaton selects one of its actions. This is the same as a member of the cluster. The CH collects requests from the selected member by the learning automata based algorithm. If the cluster member has a packet to send, it sends the packet to the CH. In this case, the selected action is rewarded. In the event that the cluster member does not have a package to send, the selected action of the learning automata is penalized. This trend continues until a new network round. The CH sends received cluster members data to the BS after some time intervals.

In this paper, the LEACH protocol is simulated based on proposed algorithm. To implementation of the algorithm on the LEACH protocol a header file is defined for all required timings and routing factor. Other file, called packet format, is formed all data structures, constants and macros related to different types of data packets. In this file, five types of packets are named Hello, Head, Member, Send-data, and Data. In the routing section of the proposed algorithm, the scheduler is defined to send various packets. In addition, the functions of constructing each of the protocol packets are identified. Each of these packet functions is assigned using the *allocation-packet* function. Then, is reached a shared header, an IP header, and a packet header. Eventually all these headers are set to the required values (according to the LEACH protocol). The *next_hop* field of the common header is the most important field where the next node address should be written. This filed presents IP_BROADCAST for hello and head packets, address of CH for member and data packets, and number one of the cluster members for Send-data packet. The *forward_data* function is used to send packets to the correct destination. In this function, data packets are examined. This function decides a packet must be received by upper layer agents or be sent to other nodes. The implementation results are simulated in MATLAB [15] and are presented in the next section.

3. SIMULATION

For simulations, 100 sensor nodes are distributed in an area of $100 * 100$ square meters in MATLAB [15] program. The BS node is randomly selected among these 100 nodes. The nodes receive the temperature data from the environment and notify the changes to the BS. In this comparison test the efficiency of LEACH_{CHMLA} and LEACH_{TDMA} protocols for various p values (the ratio of the number of clusters to the number of nodes in the network) have been tested. Any one selected as a cluster header is responsible for 500 slot times which each slot is considered five milliseconds. After every 5 slots, each CH node stores the received data in a packet

and sends it to the BS node. The network has been tested in traffic load of 0.0125, 0.0250, 0.0375, 0.0500, 0.1000, 0.1500, 0.200, 0.2500, 0.3000, 0.4000 and 0.5000. These values are average number of packets sent in per slot at a time by a node. The simulation time is considered to be 30 seconds, during which 10 rounds of clustering are performed. In each round, time for the setup clustering is considered 0.5 seconds and 2.5 seconds for the data transmission.

First Test: In this experiment, it is shown that the performance of the proposed mechanism improves over time. That is, learning automata will overtime define some members of the cluster that have more data to send and provide them a better chance to be CH nodes. For this purpose, the LEACH protocol with a value of $P = 0.077$ was implemented using two scheduling mechanisms TDMA and CHMLA. The average number of packets sent per slot at a time by a node for this test is considered to be 0.1. The experiment results are shown in Figure 2.

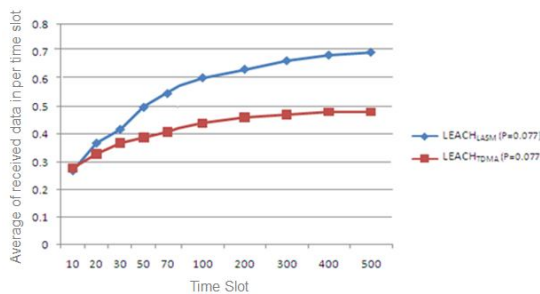


Fig. 2. Increasing average received data in overtime in LEACH_{TDMA} and LEACH_{CHMLA} with $P=0.077$.

Given the consideration of 500 slot times for each cluster to communicate with cluster members and the number of packets received in different slots, the proposed algorithm shows a better performance so whatever the time passes and the CH communicates with cluster members at higher slots, the CH will better understand the conditions of its members. It communicates more in more time slices with the member who has more data to send and as a result, it receives more data packets from its members. As can be seen, if CH communicates more with its cluster members at more slots, it receives more data. While over time, cluster headers do not differentiate between their members in overtime in TDMA based LEACH protocol. To investigate the effect of P and traffic load parameters on CHMLA and TDMA protocols, these protocols have been tested for different values of P and traffic load that are shown the results in the table 1 to 3. It can be seen that the average of received data in a time slice increases with time over the CHMLA protocol.

Second Test: In this experiment, two protocols were simulated in different traffic loads and the average received data rates were calculated at the end of simulation time. The results of this experiment are shown in Figure 3. As you can see, the LEACH_{CHMLA} protocol always has the average of the received data from the LEACH_{TDMA} protocol at any time. The reason for this is that in the LEACH_{CHMLA} protocol, the CH node identifies the active members of the cluster over time with the help of its learning automata algorithm and as a result, it gives more times to members in data transmission process.

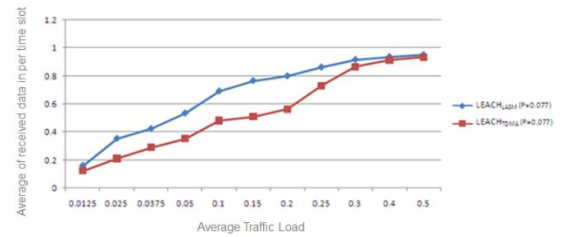


Fig. 3. The average received data in each slot time in different traffic loads for LEACH_{TDMA} and LEACH_{CHMLA} with $P=0.077$.

When the nodes do not have too much data to send, or when most nodes have data for transmission, the results from the average data received at each time slot is approximately close together for the low and high traffic loads. The results of simulation over two protocols for different values of P and traffic load that are shown in the table 4. It can be seen that the CHMLA protocol has best than TDMA protocol.

4. CONCLUSION

In this paper, a scheduling mechanism based on learning automata was proposed for the association of cluster members with cluster headers. It was to management of network channel that can be shared between all node nodes. The channel allocation in the wireless sensor networks is as important as energy. Even, they are related together. Therefore, the all parameters of network design must be defined and presented relationships of aims. The paper was proposed a new algorithm for channel management inside clusters by learning automata. It can be usable on different exists or will be presented routing protocols with different design parameters. In this paper, we implemented it on LEACH protocol and compared the result of proposed algorithm with TDMA based protocol. To implementation of the algorithm on the LEACH protocol all requires functions were defined and they were simulated in MATLAB. The simulation results were presented in two scenarios only in this paper. Therefore, our next case studies will coverage other scenarios in different algorithms. One of the output parameter of this paper was system

performance in the received data rates that we compared with two protocols in average received data in time and traffic load axis so it was observed that the proposed method gives better results than TDMA.

REFERENCES

- [1] F. Kiani, E. Amiri, M. Zamani, T. Khodadi, AA. Manaf, Efficient intelligent energy routing protocol in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2015, 1-13, 2015. <https://doi.org/10.1155/2015/618072>
- [2] F. Kiani, Designing New Routing Algorithms Optimized for Wireless Sensor Network, LAP LAMBERT Academic Publishing, Dusseldorf, Germany, 2014.
- [3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey", *Computer Networks*, 38 (4), 393–422, 2002. [https://doi.org/10.1016/S1389-1286\(01\)00302-4](https://doi.org/10.1016/S1389-1286(01)00302-4)
- [4] E. Niewiadomska, P., Kwaśniewski, I. Windyga, "Comparative study of wireless sensor networks energy-efficient topologies and power save protocols", *Journal of Telecommunication and Technology*, 3, 68-75, 2009.
- [5] M. Arshad, M.Y.AAlsalem, F.A.Siddiqui, Energy efficient cluster head selection in mobile wireless sensor networks, *Journal of Engineering Science and Technology*, 9(6), 728 -746, 2014.
- [6] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient communication protocol for wireless mi-crosensor network", in: *Proceedings of the IEEE System Sciences*, pp.1-10, January 2000.
- [7] A. Manjeshwar and D. Agrawal, "TEEN: A routing protocol for enhanced efficiency in wireless sensor networks", in: *Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, 2001. <https://doi.org/10.1109/IPDPS.2001.925197>
- [8] A. Hassan, W. Shah, M.F. Iskandar and A.Mohammed, Clustering Methods for Cluster-based Routing Protocols in Wireless Sensor Networks: Comparative Study, *International Journal of Applied Engineering Research*, 12(21), 11350-11360, 2017.
- [9] L. Bruscatto, T.Heimfarth and E.P. Freites, Enhancing Time Synchronization Support in Wireless Sensor Networks, *Sensors mdpi* 2017, 17(12), 2956, 1-18, 2017.
- [10] B. Kim, H. Park, K. Kim, D. Godfrey, K. Kim, "A survey on Rael-Time Communications in Wireless Sensor Networks", *Wireless Communications and Mobile Computing Journal*, 2017, 1-14, 2017.
- [11] K. S. Narendra and M. A. L. Thathachar, "Learning automata: An introduction", *Proceedings of the Prentice Hall*, 1989. <https://dl.acm.org/citation.cfm?id=64802>
- [12] Z. Shariyat, A. Movaghar, M. Hoseinzadeh, "A learning automata and clustering-based routing protocol for named data networking", *Telecommunications Systems*, 65(1), 9-29, 2017. <https://link.springer.com/article/10.1007/s11235-016-0209-8>
- [13] S. Tanwer et al., "LA-MHR: Learning Automata Based Multilevel Heterogeneous Routing for Opportunistic Shared Spectrum Access to Enhance Lifetime of WSN", *IEEE Systems Journal*, 1-11, 2018. <https://ieeexplore.ieee.org/document/8351993/> <https://doi.org/10.1109/JSYST.2018.2818618>
- [14] F. Kiani, Improvement of Automated Learning Methods based on Linear Learning Algorithms, *International Journal of Machine Learning and Networked Collaborative Engineering (IJMLNCE)*, 2(2), 2018.
- [15] <https://www.mathworks.com/>

BIOGRAPHY



Farzad Kiani has PhD degree and he is Assistant Prof. in computer engineering dept. at Istanbul Sabahattin Zaim University (IZU) from 2014 to now. In addition, he is deputy of head of computer engineering and head of wireless sensor networks and IoT Labs at IZU from 2016 and 2017 to now, respectively. His current research interests include wireless sensor networks, Vehicle Ad-hoc networks, flying Ad-hoc networks, machine learning, game theory and IoT. He is also working on optimization algorithms and security.

Table 1. Average received data in a slice time for the number of different time slices for the LEACH protocol with different scheduling mechanisms for different p with traffic load of 0.125.

								Protocol
	LEACH _{TDMA} (P=0.1)	LEACH _{CHMLA} (P=0.077)	LEACH _{TDMA} (P=0.077)	LEACH _{CHMLA} (P=0.0625)	LEACH _{TDMA} (P=0.0625)	LEACH _{CHMLA} (P=0.05)	LEACH _{TDMA} (P=0.05)	Number of time slots
0.053	0.059	0.064	0.051	0.041	0.061	0.052	0.047	10
0.063	0.072	0.070	0.073	0.067	0.088	0.084	0.075	20
0.070	0.084	0.083	0.086	0.084	0.096	0.106	0.090	30
0.093	0.093	0.107	0.097	0.104	0.111	0.125	0.107	50
0.103	0.096	0.112	0.102	0.122	0.112	0.148	0.119	70
0.111	0.100	0.120	0.105	0.139	0.120	0.165	0.128	100
0.120	0.105	0.145	0.115	0.173	0.130	0.182	0.140	200
0.124	0.104	0.153	0.120	0.188	0.138	0.193	0.141	300
0.127	0.105	0.159	0.122	0.191	0.142	0.204	0.143	400
0.128	0.105	0.160	0.122	0.193	0.143	0.208	0.146	500

Table 2. Average received data in a slice time for the number of different time slices for the LEACH protocol with different scheduling mechanisms for different p with traffic load of 0.1.

								Protocol/ Number of time slots
LEACH_{CHMLA}(P=0.1)		LEACH_{CHMLA}(P=0.077)	LEACH_{TDMA}(P=0.077)	LEACH_{CHMLA}(P=0.0625)	LEACH_{TDMA}(P=0.0625)	LEACH_{CHMLA}(P=0.05)	LEACH_{TDMA}(P=0.05)	
0.268	0.229	0.273	0.278	0.251	0.273	0.303	0.254	10
0.345	0.288	0.368	0.334	0.365	0.355	0.417	0.336	20
0.383	0.322	0.415	0.373	0.422	0.378	0.461	0.383	30
0.441	0.356	0.499	0.391	0.510	0.415	0.527	0.453	50
0.475	0.373	0.545	0.413	0.549	0.436	0.583	0.482	70
0.503	0.387	0.595	0.435	0.581	0.460	0.634	0.518	100
0.547	0.409	0.634	0.460	0.652	0.485	0.704	0.554	200
0.573	0.413	0.664	0.472	0.680	0.496	0.741	0.579	300
0.592	0.418	0.677	0.475	0.704	0.500	0.760	0.588	400
0.606	0.421	0.693	0.480	0.722	0.505	0.773	0.594	500

Table 3. Average received data in a slice time for the number of different time slices for the LEACH protocol with different scheduling mechanisms for different p with traffic load of 0.3.

								Protocol
	LEACH _{TDMA} (P=0.1)	LEACH _{CHMLA} (P=0.077)	LEACH _{TDMA} (P=0.077)	LEACH _{CHMLA} (P=0.0625)	LEACH _{TDMA} (P=0.0625)	LEACH _{CHMLA} (P=0.05)	LEACH _{TDMA} (P=0.05)	Number of Time Slots
0.494	0.501	0.553	0.532	0.563	0.522	0.532	0.582	10
0.599	0.592	0.665	0.650	0.653	0.630	0.643	0.690	20
0.661	0.643	0.721	0.719	0.699	0.690	0.714	0.738	30
0.729	0.690	0.778	0.766	0.751	0.754	0.788	0.803	50
0.764	0.713	0.807	0.788	0.790	0.793	0.831	0.832	70
0.794	0.735	0.832	0.811	0.829	0.816	0.866	0.857	100
0.831	0.761	0.880	0.847	0.881	0.851	0.905	0.890	200
0.845	0.775	0.900	0.857	0.897	0.865	0.923	0.903	300
0.854	0.779	0.912	0.862	0.908	0.873	0.936	0.908	400
0.859	0.783	0.920	0.866	0.913	0.878	0.942	0.913	500

Table 4. Average received data per slot time for different protocols with different traffic load values.

0.500	0.400	0.300	0.250	0.200	0.150	0.100	0.050	0.0375	0.0250	0.0125	Proposed traffic Load $(\frac{packet}{slot})$ Protocol
0.959	0.958	0.913	0.768	0.643	0.592	0.594	0.374	0.325	0.198	0.146	LEACH _{TDMA} (P=0.05)
0.959	0.973	0.942	0.867	0.909	0.816	0.773	0.556	0.474	0.331	0.208	LEACH _{CHMLA} (P=0.05)
0.977	0.916	0.878	0.799	0.571	0.535	0.505	0.314	0.324	0.211	0.143	LEACH _{TDMA} (P=0.0625)
0.979	0.939	0.913	0.894	0.814	0.761	0.722	0.488	0.449	0.328	0.193	LEACH _{CHMLA} (P=0.0625)
0.933	0.912	0.866	0.731	0.563	0.508	0.480	0.351	0.290	0.208	0.122	LEACH _{TDMA} (P=0.077)
0.953	0.938	0.918	0.864	0.801	0.767	0.693	0.535	0.424	0.354	0.160	LEACH _{CHMLA} (P=0.077)
0.954	0.819	0.783	0.698	0.593	0.427	0.421	0.288	0.283	0.181	0.105	LEACH _{TDMA} (P=0.1)
0.97	0.879	0.859	0.887	0.867	0.721	0.606	0.431	0.400	0.289	0.128	LEACH _{CHMLA} (P=0.1)