

Learning in cognitive wireless sensor networks in the course of completing cognitive engine: distributed method based on Q learning

Samira Davarghalati

Department of Computer
 Engineering and Information
 Technology (QIAU)
 Qazvin, Iran
 Davarghalati.s@gmail.com

Mohammad Reza Meybodi

Department of Computer
 Engineering and Information
 Technology, Amirkabir University
 Tehran, Iran
 Mmeybodi@Aut.ac.ir

Abstract— one of the most challenging issues in cognitive networks is the selection of appropriate mechanism and presenting a learning model in the course of complete cognitive engine. In fact, in cognitive process, learning is to complete decision making and it helps future decision making and planning through maintaining the previous decisions under a set of specified conditions, or by revealing behavioral patterns in network environment. In this paper, we use Q learning algorithm, which is one of reinforcement learning techniques, to obtain appropriate value of each parameters of protocol stack in cognitive engine of cognitive wireless sensor networks. Using Q algorithm provides learning opportunity to solve the problem in a large operational environment where there is limited information. The proposed protocol is simulated by NS2.35 simulator. In this paper, we compare the proposed method with the existed method based on simulated annealing. Some network's end-to-end criteria such as end-to-end delay average, oscillation of delay, throughput and packet loss, and generally the useful function which is measured by these criteria are evaluated. The results of simulation show that the proposed method has better performance than the previous one and increase significantly the utility function.

Keywords— wireless sensor networks; cognitive network; Q-learning; cognitive engine

I. INTRODUCTION

One of the tools for acquiring environmental information which has recently attracted the attention of many researchers is wireless sensor networks. A sensor network contains many sensor nodes which are small and cheap and also have limited computational and processing resources. Most of these sensor nodes can be rapidly expanded in a very large area of space in order to form a weak connection distributed network system called wireless sensor network [1]. Complexities in managing the wireless sensor networks are the main reason to pay

attention to cognitive networks [2]. For the first time, cognitive networks were introduced by Thomas in [3]. Despite all performed researches, some basic challenges related to cognitive networks have not been solved yet. One of these most important challenges is the lack of a proper method for learning. In fact in cognitive process, learning is to complete decision making and it helps future decision making and planning through maintaining the previous decisions under a set of specified conditions, or by revealing behavioral patterns in network environment. Thomas presents a general framework for cognitive networks, but does not discuss constitute elements of these networks. Cognitive network obtain the cognition of network level and all around the networks nods by means of performing cognitive cycle among layers of protocol stack. Such nods needs a model which be able to observe the network status, group reasoning to reach global goals, learning from previous acts and re-configuration of cognitive nods based on group decisions. For this purpose, we design an algorithm that learns without primary data and during interaction with environment and completes cognitive engine in cognitive networks by setting available network parameters. Also improves some important factors of performance such as delay, packet loss and throughput which totally describe our utility function.

The remainder of this paper organized as follows. In section 2, we review the related work and present Q-learning algorithm. Section 3 presents reinforcement algorithm. Section 3 presents the proposed method. Section 5 reports experimental results and presents an analysis of the properties of proposed model and Section 6 concludes the paper.

II. LITERATURAL REVIEREW

A. related work

In [4], an area is introduced which provides re-configuration of total protocol stack. In [5] it is assumed that the area presented in [4] exists and it is focused on adding learning abilities and distributed reasoning to it. Presented models in [4] and [5], show tasks and connections of a cognitive node, but still remained as a theory. In [6,7] by choosing Bayesian networks and simulated annealing(SA), as mechanisms of reasoning and learning, some architectures are presented for learning model. The abovementioned learning models lack necessary features to be converted to a useful framework. In [7] by employing “black box” technique and simulated annealing algorithm, has been discussed about of the design and optimization of algorithm for cognitive networks.

In recent years, calculation intelligence patterns are applied to various challenges of topics proposed in wireless sensor network. These algorithms, due to production of adaptive mechanism, reveal intelligent, autonomous and flexible behaviors in complex and dynamic environments such as wireless sensor networks. Among the proposed calculation intelligence patterns, reinforcement algorithms [8] and also Q-learning, because they don't require any data except reinforcement signal from the environment, are advantageous than other intelligent patterns. In reinforcement learning one learning factor reached to a control policy during learning by repetitive interaction with the environment.

B. Q-learning

Reinforcement learning is a learning technique based on trial and error. The basic idea behind reinforcement is that the learning system shown in fig. 1 can learn how to solve a complex task through repeated interaction with the environment.

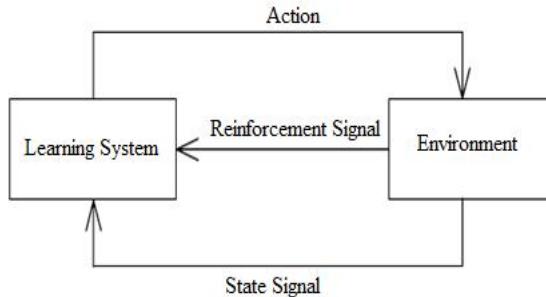


Fig. 1. Reinforcement learning system.

The learning system receive state information about the environment by means of its sensors, and this state information is used by a reasoning process to determine the action to be given state. After the action is implemented, the learning system receive a reinforcement signal from the environment to indicate the consequences of its action. The goal of the learning is to maximize the amount of reward it receive in the long term [8].

Watkins introduced the method of reinforcement learning called Q-learning in 1989[9]. It is a reinforcement learning

algorithm that attempt to learn a the state-action value $Q(x, a)$, whose value is the maximum discounted reward that can be achieved by starting in state x , taking an action a from state x at time t , the current state-action pair value estimate from a and x donated by $Q(x, a)$ is updated as follows:

$$Q_{t+1}(x, a) \leftarrow Q_t(x, a) + \alpha[r_{t+1} + \gamma \max_{b \in A} Q_t(y, b) - Q_t(x, a)] \quad (1)$$

When y is the actual next state, and $0 < \gamma < 1$ is the discount factor and r is the payoff the agent receives when action “ a ” is taken in state x . The state-action value estimate for other state and actions remain unchanged.

Distributed Q learning is presented for multi-factor models and also collaborative decision making processes which there is a set of simultaneous and independent activities, and in that set, factors try to maximize total rewards.

III. PROPOSED METHOD

Cognitive process, which is one of the most important features of cognitive networks, is responsible for learning and reasoning in cognitive network. In learning, the cognitive engine first evaluates the relationships between the previous actions and the present observations and also the relationships between concurrent observations, and then converts the result into the savable information in a knowledge base. At first we have some parameters in knowledge base of cognitive box which the least knowledge about them is categorizing them into settable and observable parameters and possible actions for settable parameters. Along with actions, possibilities related to actions also enter to the cognitive box. The general system model shown in fig. 2.

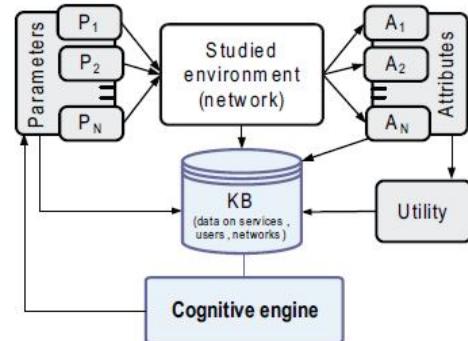


Fig. 2. Overall system model.

Cognition box part should perform the reasoning and learning successfully independent of various scenarios of network. Initial entries of cognition box are network parameters. Reasoning process starts its work. Its aim is to find the best performance (proper value to set the adjustable parameter) for each parameter by considering the final aim of the network. This duty is performed by Q- learning in various layers.

In the proposed algorithm based on Q- learning along with adjustable parameter, network situation is also considered; therefore there are duplex pairs "state / action". Network situation is considered regarding our criteria of service quality

by means of four parameters "delay, oscillation of delay, throughput and packet loss". Because the values of these parameters were real, in network situation we consider two values, zero and one, for each of these parameters; so we determine its being acceptable or not being acceptable threshold according to a limit, so that below threshold is considered as zero(0) and above that is considered as one(1). We use distributed Q- learning to solve the problem as follow. We consider a p table "state, action" for each parameter, so that we write the situation or state of the network based on impossibly features in this parameter. According to pairs value "state/ action" in each parameter, evaluation function selects a proper value; these results are given to utility function and update the possibilities.

The considered model for network is based on four layers standard model TCP/IP. In this paper, we assume transport layer protocol as reliable and our goal is to work on three layers of these four layers. Based on the process of data delivery in network, first the application layer is in relation to environment and receives the data, the value of each parameter in protocol stack is according to statistical information gathered of the network and based on [7].

The values considered for each parameter are in table 1 where these parameters are also based on the collected statistical information and network information related to our features.

Table 1. Parameter values.

layer	parameter	values
Application	Sensing Interval	200,800,1400
AODV	Beaconing Interval	5,20,40
	Beaconing Timeout	200,800,1500
MAC	Duty Cycle	10,50,80,100
	Ack	"on", "off"
	Max.Retransmission	1,3,9
	Back off length	5,30,40

According to impossibly situations of the network on each parameter and its values, the parameter is determined based on duplex pair state / action. Therefore, we start from application layer parameter and the proper value for the parameter in this layer is determined by Q learning:

- Sensing interval: time interval of nodes in order to control the environment

Next, by assuming that transport layer protocol is reliable, we go to the parameters of network layer. We consider AODV as routing protocol. In this layer two following parameters should be adjusted:

- Beaconing interval: interval before starting data delivery which is used as routing (table formation) in AODV.
- Beaconing Time Out: the longevity of routing or records of the table, determines the routing table, that if it is short, the connection is disconnected in the data delivery and the routing should be performed again.

Therefore, we consider 802.11 protocol in the media access control layer. The values of four parameters that should be adjusted in this layer are as following:

- Duty cycle: the time which the nodes deliver their data sensors and receive accept validation.
- Max. Retransmit: we consider some re-delivery for a route between source and destination.
- Back-off length: A time interval that the sender has chosen a random number after an unsuccessful delivery of a packet and waits and then acts to re-delivery of the packet.
- Ack: the receiver delivers the accept validation message to the sender after receiving the packet.

For example, in sensing interval parameter because to investigate the situation of the network influenced we consider sensing interval. These parameters include loss of packets, delay and throughput, that finally establish eight various situations. Possible values to initialization, sensing interval, also can be 200, 800, 1400 by using available statistical information. As a result, 8×3 table for pairs state/ action is definable in this parameter. The process of adjusting a parameter is repeated until an appropriate action to be selected for the parameter. The above levels are repeated for all processes. Gradually a graphical model will be formed from the relationship between parameters based on network model in a knowledge base. The feedback of action selection is received all the time until the relationship with the environment is maintained. The goal of cognitive engine is optimization or maximizing utility as a function of the parameters considered by network service.

IV.

In this section, the simulation results of the proposed protocol with NS2.35 simulator [10] are shown. Simulations has been conducted for a network with 100 sensor nodes. These sensor nodes are randomly distributed in area of 10×10 m². All of the sensor nodes have assumed which are constant and flat networks. The Initial assumptions for the proposed algorithm parameters are shown in table 2.

Table 2. Parameters used in the simulation.

Parameters	Value
Network size	10×10
MAC	802.11
Routing	AODV
Number of nodes	100

The simulation is continued until the energy of the first sensor node to be ended in the network. Energy model for sensor nodes is considered based on the model described in [10]. In order to convergence, all tests are repeated for 10 times and the average of values obtained in simulation are used to investigate performance. In table 2, the used parameters in the simulations are shown.

Evaluation of efficiency of the proposed algorithm is performed by considering four items of service quality that finally form utility function and is compared to the existing method in [7].

- 1) The average of end to end delay: for every packet that is delivered, end to end delay shows a period that the packet use

between source and destination and is calculated for all packets as follow.

$$\text{Averagedelay} = (\text{packetarrival} - \text{packetstart}) / n \quad (2)$$

As we can see in fig. 3 the proposed method based on Q learning bears less delay for Q learning; the reason for this issue is no need to global information and using local information.

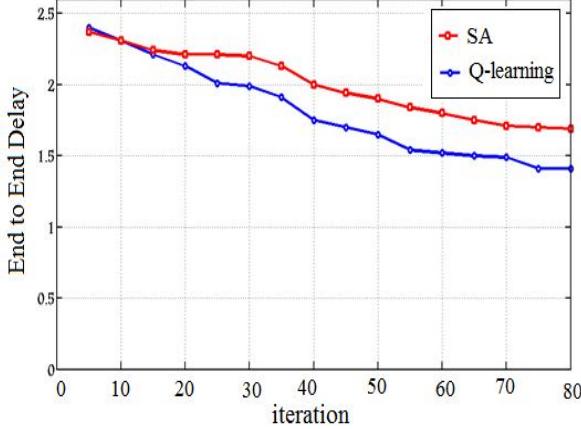


Fig. 3. Impact of increasing iteration on the end to end delay.

2) Jitter: This parameter is delay of packet's changes and has a serious influence on quality. This parameter is used as adaptation and consistency index of a network. Equation (3) shows the method of calculating this criterion.

$$\text{jitter} = ((\text{recvtime}(j) - \text{sendtime}(j)) - (\text{recvtime}(i) - \text{sendtime}(j))) / (j - i) \quad (3)$$

The simulation results shown in fig. 4 illustrate that in the proposed method there is more consistency, because oscillations bear less delay. And finally is converged to 0.23.

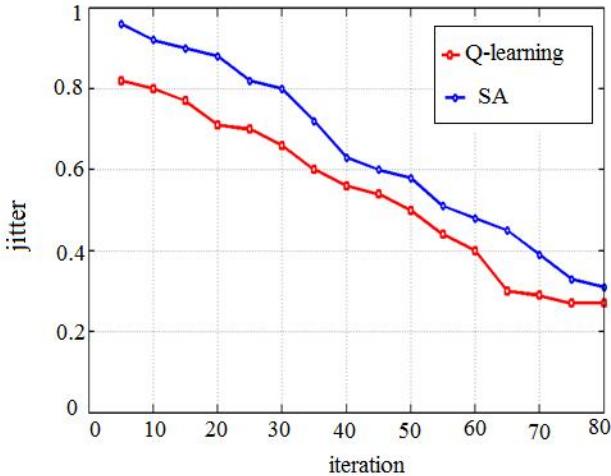


Fig. 4. Impact of increasing iteration on the delay.

3) Packet loss: the packet loss rate is the average of losing packets that belong to an individual flow that happens in network infrastructure for various reasons.

$$\text{packetloss} = (\sum \text{loss} \cdot \text{packetsize} / \sum \text{packetsize}) \times 100 \quad (4)$$

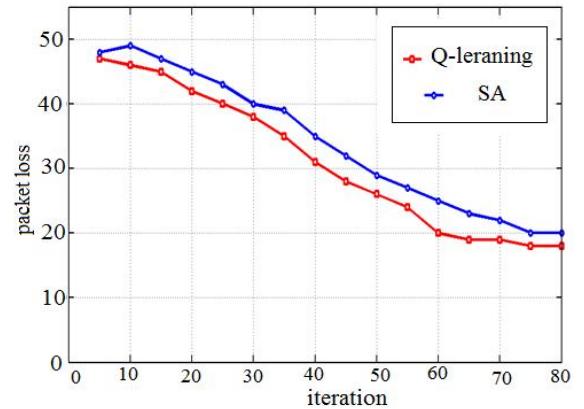


Fig. 5. Impact of increasing iteration on the packet loss rate.

4) Throughput: Throughput based on the definition, includes some packets which are arrived the destination in a determined time successfully.

$$\text{throughput} = \sum \text{packetsize} / (\text{packetarrival} - \text{packetstart}) \quad (5)$$

In throughput criteria such as loss rate and delay in arriving packets are important. As we can see in fig. 6, the proposed method has a better throughput. The reason of this issue is that by passing time packet losses and delay in arriving packets will be reduced.

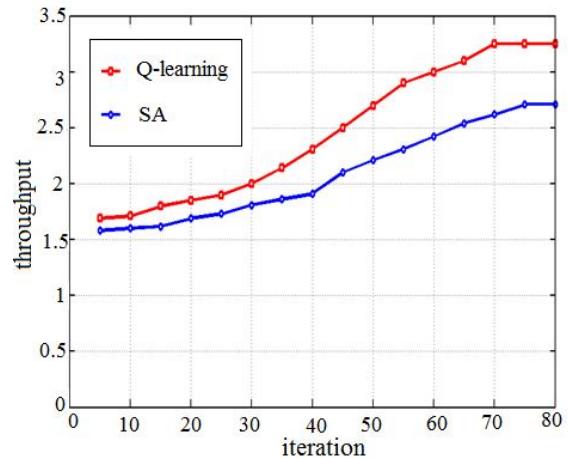


Fig. 6. Impact of increasing iteration on the throughput.

Finally in this part, the aim of evaluating general efficiency of the network is considered according to the four service quality criteria. Based on the results of the fig. 5, by increasing the iterations number, the proposed method based on Q learning has more value.

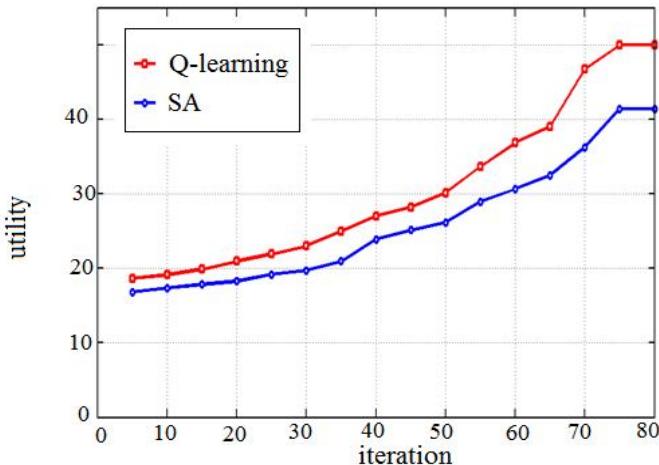


Fig. 7. Impact of increasing iteration on the utility.

In this paper the presented learning model is suitable for dynamic environments in automatic configuration of the network without requiring initial information or global information. The presented algorithm is compared with the existing method based on SA. Simulations based on criteria such as end to end delay average, oscillation of delay (jitter), packet loss and network throughput, and utility function with increasing of network density have been done. In the proposed method by using distributed Q learning, proper value of each parameter in various layers have been obtained. Due to use of local information and not requiring obtaining lots of knowledge of the network, Q learning algorithm, has proper performance in wireless sensor networks.

In the future works we attempt to add movement to the network nodes, speed, and orientation of node's movement in setting of evaluation network. We will investigate other performances of metaheuristic algorithms in cognitive cycle of cognitive networks as well.

REFERENCES

- [1] J. Yick, B. Mukherjee, and D. Ghosal," Wireless sensor network survey," Proceedings of Computer Networks Elsevier, vol. 52, no. 22, pp. 2292-2330, 2008.
- [2] C. Fortuna, M. Mohorcic, "Trends in the development of communication networks: Cognitive networks," IEEE transaction on Computer Networks, vol. 53, no. 25, pp. 1354-1376, 2009.
- [3] R. W. Thomas, "Cognitive Networks," PhD thesis, Electrical and Computer Engineering Department, Virginia Tech University, Blacksburg, Virginia, 2007.
- [4] P. Sutton, L. E. Doyle, and K. E. Nolan, "A reconfigurable platform for cognitive networks," In Cognitive Radio Oriented, 1st International Conference on Wireless Networks and Communications, pp. 1-5, 2006.
- [5] D. H. Friend, "Cognitive Networks: Foundations to Applications," PhDthesis, Electrical and Computer Engineering Department, Virginia Tech University, Blacksburg, Virginia, 2009.
- [6] G. Quer, H. Meenakshisundaram, B. Tamma, B. Manoj, R. Rao, and M. Zorzi, "Cognitive Network Inference through Bayesian Network Analysis," In IEEE Global Telecommunications Conference (GLOBECOM 2010), pp. 1-6, 2010.
- [7] E. Meshkova, J. Riihijarvi, A. Achtzehn, and P. Mahonen, "Exploring simulated annealing and graphical models for optimization in cognitive wireless networks," In IEEE Global Telecommunications Conference GLOBECOM, pp. 1-8, 2009.
- [8] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," Journal of Artificial Intelligence Research, vol. 4, pp. 237-285, 1996.
- [9] Huang, J., Yang, B., Liu, D. Y., "A Distributed Q learning Algorithm for Multi-agent Team Coordination," Proc. 5th International Conference on Machine Learning and Cybernetics, Guangzhou, pp. 18-21, 2005.
- [10] "Network Simulator" [online] <http://www.isi.edu/nsnam/ns>.
- [11] Maraiya, K., Kant, K., Gupta, N., "Efficient Cluster Head Selection Scheme for Data Aggregation in Wireless Sensor Network", International Journal of Computer Applications, Vol. 23, No. 9, pp. 10-18, 2011.