Technical Journal of Engineering and Applied Sciences
Available online at www.tjeas.com
©2013 TJEAS Journal-2013-3-22/3064-3070
ISSN 2051-0853 ©2013 TJEAS

A novel adaptive version of AODV routing protocol based on learning automata utilizing cognitive networks concept

Mahdi Ghorbani[1*], Ali Mohammad Saghiri[2], Mohammad Reza Meybodi[2]

1. Department of Computer & IT Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
2. Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran

Corresponding author: Mahdi Ghorbani

ABSTRACT: Computer networks experience an extensive growth these days with their design being related to some issues like wireless connection, heterogeneous environment and scalability. Concept of the cognitive networks is one of the new concepts which can cause some characteristics such as self-adaptive by creating a special kind of intelligence based on cognitive cycles. Various intelligent mechanisms have been proposed for improvement of AODV protocol. Our aim in this paper is providing and organizing a cognitive cycle in improvement of the AODV protocol which is obtained by a kind of intelligent delaying to the route discovery packets. The AODV route discovery algorithm is investigated first in this paper, and then a cognitive cycle is developed based on a network of automata which tries to learn the validity between the nodes according to the probabilities. At last, a changed protocol named LAODV is presented which follows a kind of intelligent delaying in the route discovery phase. This algorithm is finally examined regarding reduction of the failure rate of links in different conditions which indicates efficiency of the proposed algorithm.
Keywords: Computer networks, Cognitive networks, Routing algorithms, Learning automata

INTRODUCTION

First, development of the cognitive radio networks leads to introduction of the technology of the cognitive networks (Mitolla 2000; Mitola et al. 1999). Concentration of the cognitive radio networks was on management in lower layers of the wireless networks (Mahmoud 2007). Using the approach of cognitive networks for designing a computer network can decrease part of design complexities of the management procedures in this system.

A cognitive process enables learning from previous experiences for improvement of the decision making in future of the system. Application of the cognitive process in the cognitive radio networks improves efficiency indexes of these networks and thus makes it possible to employ this approach in optimization of other issues in the network (Mahmoud 2007).

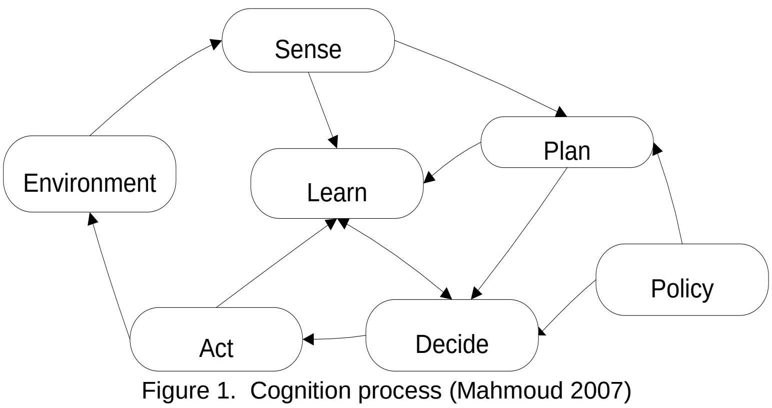

Figure 1. Cognition process (Mahmoud 2007)

Main focus of the cognitive networks is on designing improved management procedures using variables from all layers of the network. The cognitive networks by designing learning cycles would be able to cause some intelligent changes in the network. Using the approach of cognitive network in the network leads to an issue called adaptive software networks (Thomas et al., 2007; Perkins et al., 1999).

A greater number of nodes are involved in the process of learning and optimization in the cognitive networks. An interesting point is seen in a kind of high capacity systems for using distributed learning algorithms which can play a new role in setting parameters of the current protocols. Optimization methods in the technology of cognitive networks are different from those like improvement of cross-layered methods which are generally suggested together with each other in the network optimization technologies (Thomas et al., 2007).

One approach to implement the cognitive process is to apply learning cycle which is used in machine learning and artificial intelligence techniques (Mahmoud 2007). It is also possible to design a cognitive cycle based on learning automata which is a reinforcement learning model (Song et al. 2010; Song et al. 2007; Liu et al., 2010; Akbari Torkestani et al., 2010).

This paper proposes a new method for creating a cognitive process for improvement of the AODV (Perkins et al., 1999) route discovery algorithm which can reduce the failure rate of the links via a kind of intelligent delaying to the route discovery packages. At first, a new parametric algorithm is introduced in this paper for changing the AODV route discovery algorithm, and then a network of learning automata (Sastry et al., 2004) is utilized to set the parameters of this algorithm.

The remainder of the paper is organized as follows. Some related works in the literature are studied in Section Two and Section Three reviews the theory of learning automata. Section Four will determine position of the problem, while Section Five will suggest a new protocol and Section Six will evaluate it.

Related Works

Improvement of the AODV protocol has been studied so far from various aspects by different reinforcement learning methods (Marconett et al., 2012; Lee et al., 2007). For example, cross-layeredimprovement using Q-learning which is shown in (Lee et al., 2007) as a cognitive approach in the network management. However, these cognitive approaches will not be limited to cross-layered optimization according to (Thomas et al., 2007;Friend et al., 2009). It should be noted here that the approaches which are based on creation of the cognitiveprocesses are a bit different from the current intelligent making approaches which are mainly based on the AODV protocol. Cognitive radio networks are among the new approaches for creating the cognitive cycles in the AODV route discovery process, in which the AODV is somehow modified to increase using the free capacities. The field of cognitive networks includes a wide range of versatile algorithms in the computer networks the aim of which is creating learning cycles. Each kind of the intelligent making process cannot be considered as a cognitive approach (Mahmoud, 2007).

Using the learning automata for creating a cognitive cycle has been under consideration for a while. For instance, (Song et al., 2010; Song et al., 2007; Liu et al. 2010; Akbari Torkestani et al., 2010) can be pointed which are based on the learning automata (Zarei et al. 2008)and learning automata network (Sastry, 2004) of the cognitive processes. The learning automata have proper performance in unknown and uncertain environments which has made them an appropriate tool for the cognitive networks. This approach can be used in new designs for the route discovery algorithms such as AODV.

For additional approaches with the same function in using the learning automata as estimators of the node stability, (Zarei et al., 2008) can be named. However, the proposed mechanisms for the AODV-based improvement of the protocol, uses the second phase of route discovery for selection of the optimal route and creating the alternative route which is also compared with the suggested method. The proposed approach is rather different from the previous works in terms of algorithms, so that looking at the cognitive cycle based on the learning automata is not the only innovation provided. In fact, the approach based on intelligent delaying to the route discovery packages in the AODV protocol is major part of the provided work in which a cognitive mechanism is utilized to determine the delay parameters.

Learning Automata

The learning automaton is an adaptive decision making unit which is able to find an optimal action via interaction with the environment that is searching the state space in other words. Indeed, the learning automaton is a random automaton having interactions with the random environment.

Every automaton is specified by a set of internal states, input actions, probability distribution of states and reinforcement function. The learning automata do not need further information for their operation. In other words, failing to use additional information which was regarded a drawback, could also be an advantage in using the automata in the environment or unknown applications at the same time.

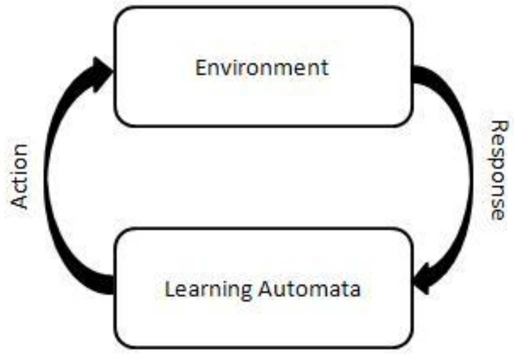

Figure 2. The relationship between the learning automaton and its random environment (Narenda et al., 1989).

A learning automaton is determined by limited actions of a quadruplet $< A, Q, R, L >$ or by an environment of $< A, R, D >$, where, A is defined by the set of $\{\alpha_1, \alpha_2, \dots, \alpha_r\}$ which indicates r actions of α type. On the other hand, they can be outputs for the environment.

Rdenotes the range of automata answers for the environment and includes $\beta(k)$s which are answers of the environment in one hand and inputs of the automata on the other hand against the $\alpha(k)$ action in the k^{th} step. The symbol D is also the set of reward probabilities which is given as $D = \{d_1, d_2, \dots, d_r\}$, where d_i is defined as $d_i(k) = E[\beta(k)|\alpha(k) = \alpha_i]$ at the k^{th} step. The probability vector based on which the automaton selects and action each time at the k^{th} step is called $P(k)$,

$$P(K) = [p_1^{(k)}, p_2^{(k)}, \dots, p_r^{(k)}] \tag{1}$$

and L is the learning algorithm or the reinforcement function which is used by the automaton for updating the internal state as depicted in the figure below.

$$Q(k+1) = L(Q(K), \alpha(k), \beta(k)) \tag{2}$$

The automaton selects an action at each step based on the probability vector. Then, they revise their internal variables according to the environmental feedback and the reinforcement function. The automaton continues this procedure until finding the optimal action. The probability of optimal action tends toward unit in the final state, while the reward approaches to its maximum value.

The learning automaton performs its selection based on the probability vector at each step of choosing an action and updates the probability vector at the end. The ultimate goal here is to create a network of automata which update the probability vector based on the previous records.

A sample of the learning algorithms in the learning automata is the linear learning algorithm. Assume that action α_i is performed in the n^{th} step. If the reward is received, then Equation (3) is employed to update the probability vector and otherwise, Equation (4) will be used.

$$p_j(n+1) = \begin{cases} p_j(n) + a[1 - p_j(n)] & j = i \\ (1-a)p_j(n) & \forall j, \quad j \neq i \end{cases} \tag{3}$$

$$p_j(n+1) = \begin{cases} (1-b)p_j(n) & j = i \\ \left(\dfrac{b}{r-1}\right) + (1-b)p_j(n) & \forall j, \quad j \neq i \end{cases} \tag{4}$$

In Equations (3) and (4) a and $b$represent the reward and penalty parameters, respectively. When$a = b$, then the learning algorithm is called L_{RP} and when$b = 0$, then the learning algorithm will be called L_{RI} (Sastry, 2004; Narenda et al., 1989).

Problem Statement

In Ad-Hoc networks, each network node is responsible for the routing such that each node must also act as a router in these networks. In these networks, the transmission is usually done in multiple steps. Because of the receiver node usually is not located in the radio range of the transmitter, the intermediate nodes between the receiver and the transmitter should transmit the data packets under study from the transmitter to the receiver. Furthermore, before deciding which node should do this task, routing must be done in the network. For this purpose, one may need to send route discovery packets or routing information (Garg, 2007).

The routing protocols are generally used to find and keep the routes between source and destination nodes. Two main categories of the Ad-Hoc routing protocols are those based on table (proactive) and those based on request (reactive).

In the first approach, each node keeps a routing table including some routes which exist between the network nodes. These nodes must continuously interchange messages with the routing information in order to keep their routing tables updated. The latter are highly potential for dynamic and large networks. Once a node needs a route to reach another node, a routing process is created to show the route (Garg, 2007).

AODV is a protocol based on request, so it can be said that the nodes which are not located on an active route neither keep the route information nor take part in periodic exchange of the information of these tables. A node will discover a route when it needs to communicate with that or it acts as an intermediate station for communication of the others. Objective of the algorithm is reduction of the route overhead just when needed (Perkins et al., 1999).

AODV is comprised of two phases, namely route discovery and route reply. Each of these phases has different problems which require setting of intelligent mechanisms. The route discovery phase is investigated together with the dimensions of the related problems in the following. Then, a solution will be suggested for it.
Two types of messages are defined with the names of RREQ and RREP in this protocol. Each entry of the routing table contains destination address, number of next node, number of destination and active neighbors.

If the source node has no route to the destination, then a route request (RREQ) message is issued extensively (with no delay).

When an intermediate node receives a RREQ message, it evaluates whether the message already exists in the table or not. If the intermediate has received the RREQ message before, it deletes the message and otherwise it creates a return route.

The intermediate nodes direct the RREQ nodes toward the next nodes. When the destination gets the first RREQ message, it turns back a reply message from the return route and stabilizes information of the intermediate node. As a result, those nodes which are located in the return route and get no RREP message clean up the return route after some time which is often set manually based on designer’s estimations.

Proposed Algorithm

An issue which emerges for the AODV protocol is that the route discovery is done continuously due to failure of the links. This is done in the base protocol without considering history of the nodes which causes the low quality links to be selected once again and the rate of route discovery operation to be rather high in the network.

Motivation

In the AODV protocol it is assumed that the first received RREQ has passed a much shorter route so that the destination will return RREP by looking at the first RREQ. It is possible that the route is built from transient nodes in this algorithm and more routes with greater stability exceed range of the answer. The cognitive cycle expected here for studying the histories is second probabilistic issuance of the RREQ packets. Issuance of this message for the second time in the preliminary methods was implemented immediately, extensively and without considering history of the route which was changed in the algorithm developed in the current research.

When the RREQ message issues a neighbor of inappropriate history rather late, it would be less promising to put the instable neighbor in the RREP route.

The RREQ search message can be probably issued with or without delay every time, though its delayed issuance would be certainly less probable. For example, an issuance graph is illustrated in Figure 3. If the marked route by the AODV protocol is defined as appropriate, then each node will give a greater validity to a neighbor from which has received the RREQ message. These nodes will also be among the first priorities of issuance in the next interactions.

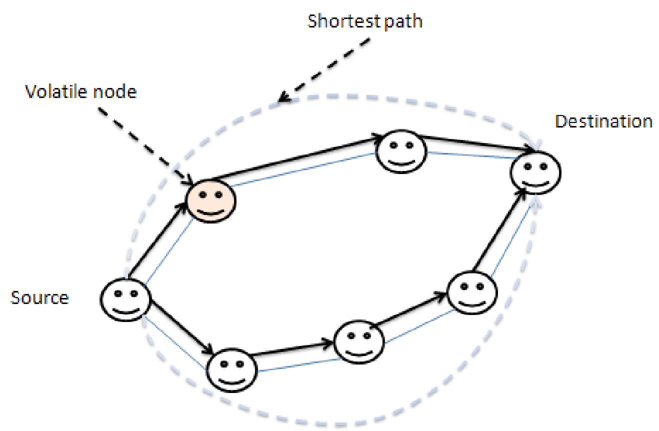

Figure 3. Different routes

Data Structure

In this protocol a learning automaton with two actions is defined per each destination. A bisection probability vector of $< P_1, P_2 >$is defined in the routing table which is equal to the actions α_1 and α_2 which keep settings of the learning automaton. Meanwhile, a counter called TS is assigned to keep validity of each neighbor inside it. Each node makes a learning automaton and a counter for each destination it knows, so it is obvious that the probability vector of this automaton is initialized with the value 0.5. The TS time scheduler has a double counter with minimum and maximum values of 0 and 1, respectively. The value of this counter must be set by a mechanism based on the neighbor state under study.

LAODV Protocol

In this protocol like AODV, the route discovery phase is started through issuance of the RREQ packet by the source node. Each node activates the automaton dependent on the destination under study immediately after receiving the message. Noteworthy here is that if the destination is unknown, an automaton with initial probability values of 0.5 will be created. This automaton decides whether to issue the message with delay or not. The value of TS counter is locally used for calculation of the delay such that $TS * (1/hopcount)$ is multiplied by the maximum hop count for the delay and the delay time is calculated.

Finally, the value of delay-free issuance probability is extracted from the probability vector of the automaton upon issuance of the message and multiplied then by the probability of the route passed in RREQ. The steps for creating the return route are also the same as those in the AODV.

When finding the destination in the return route of RREP, it turns back the source node of RREP in the return route due to success of the routing procedure. Moreover, the intermediate nodes are rewarded with the amount of 1 in each node when they try positively (reduction of the counter) in making the route (by Equation (3)) or otherwise penalized (by Equation (4)).

A new message called $Corrector$ is defined in this method. The destination send them a penalty response when the previous routes are not appropriate and sends the now route a reward response immediately after finding the better route by receiving the delayed RREQs and calculating their probabilities in the destination. Therefore, each neighbor of the preliminary probability vector is temporarily kept for a while which enables updating the probability vector properly whenever a reward or penalty is issued for the route. Moreover, the only change which is made in the RREQ packets is adding a field for keeping the route probability.

The routes with no return restore their previous state at the end of timeout due to failure of the route discovery algorithm. However, if they receive the $Corrector$ message sent by the destination for them, they will update their probabilities.

In fact, when a better route is found in this method, it will not change the established connection; it rather corrects the probabilities for route discovery in the future.

The nodes of higher stability are more likely to exist in the final route. Each node achieves some validity after some time due to its interactions. These interactions enable the node to be located in the final route which is indicative of a kind of stability in the topology.

Evaluation of Proposed Algorithm

J-sim (Sobeih et al., 2005) has been utilized for evaluating the proposed protocol. The main purpose of this paper is to measure the average parameter of the failed links and the overhead created by three techniques of AODV, LAODV and RAODVA (Zarei et al., 2008).

Simulation Conditions

The sources and destinations are selected randomly and out of them 10 sources and 10 destinations are investigated. A number of nodes (30 to 200) were distributed in the environment in these simulations, some random number of them were static while some others were made dynamic using a random model. Movement rate of the nodes has been determined 0 to 10 m/s and the stationary times are assumed to be 20 s. The data packets were determined as 512 bytes with the transmission rate being set to 5 packets per second. The simulation diagrams for each of them were derived from average of 19 runs. The simulation environment was selected from three environments in the implemented simulations with the range of transmission being considered as 200 m.

First Scenario

Average failure rate of the links is measured in this experiment by different speeds of the nodes. The main purpose here was to examine the relationship between speed of the nodes and performance of the protocol. In this scenario, the number of nodes is 100, while the speed of nodes will be variable in the range of 0-5 with the size of environment being considered as 1000 m^2.

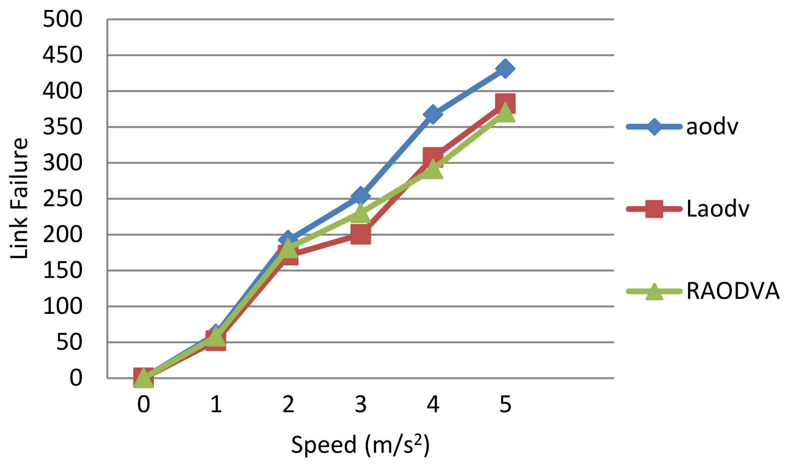

Figure 4. Link failure rate vs. Speed

This experiment shows that learning automata-based protocols present a better performance than AODV protocol. The reason is the improvement of decision-making processes in the proposed protocols which are based on environment changes. As it is showed in the graph, our proposed protocol has a better performance in low speeds than other protocols by emphasis on neighbors.

Second Scenario

Average failure rate of the links is measured in this experiment for different nodes. The main purpose here was to examine the effect of the number of nodes on the average number of link failures. In this scenario, the number of nodes varies from 100 to 200, while the speed of nodes will be variable in the range of 0-5 with the size of environment being considered as 3000 m^2. This test shows that the suggested protocol has demonstrated a better performance when the number of nodes is much smaller. When the number of nodes reaches to 200, performance of the both protocols which are based on the learning automata will be almost the same due to increased density of the nodes and will result in a much smaller probability of route break down. However, when density of the nodes is smaller, the LAODV technique which is based on validation of the more stable routes will provide a better performance.

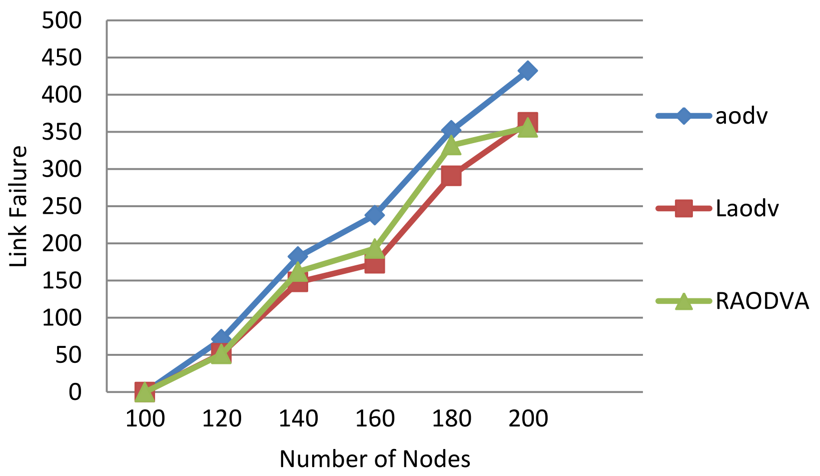

Figure 5. Number of nodes vs. Link failure rate

Third Scenario

Overhead of the protocol is assessed in this experiment in terms of the number of generated messages. Taking into account the proper performance of the proposed protocol in relatively low density and speed of the nodes, the aim here would be studying the overhead imposed on the network in appropriate conditions needed for execution. Since a relatively good performance was obtained for the developed protocol with 150 nodes and an area of 3000 m^2 at speed of 5, it is tried in this experiment to study the overhead of this protocol in appropriate conditions. For this purpose, speed of the nodes are set to maximum 5 and the number of nodes is increased from 100 to 300 in an area of 3000 m^2 large to measure the overhead. The obtained results indicate that the proposed protocol creates an overhead much smaller than that of RAODVA method in an appropriate environment. Although increasing the nodes in the previous tests had an appropriate performance for finding the more stable routes, it causes a larger overhead due to the increased number of route probability correction messages.

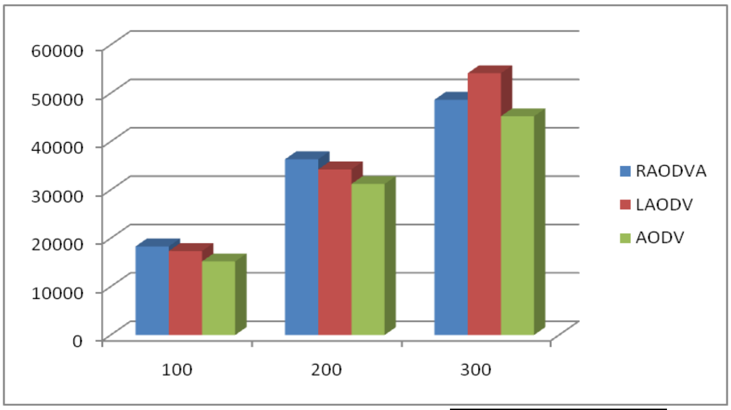

Figure 6. Relationship between Number of n… of protocol

CONCLUTIONS

Various techniques have been suggested to improve the AODV protocol. One new solution has been introduced in this paper. In this protocol, the probability of those nodes with weaker history has been reduced in the final route and much more stable routes are formed by using a kind of delaying to reissuance of the RREQ message again. The results of simulation also indicate that the suggested mechanism acts better when the network is more stable. A mechanism was proposed in this paper for routing the AODV in a probabilistic form which was operated by decision making based on histories of a node in the routing process. This decision making can be improved by increasing lifetime of the network, enhancement of the experiences and improvement of the selection probabilities.

REFERENCES

Akbari Torkestani J, Meybodi MR. 2010. A learning automata-based cognitive radio for clustered wireless ad-hoc networks. Journal of Network and Systems Management 19: 1–20.

Cacciapuoti AS, Calcagno C, Caleffi M, Paura L. 2010.CAODV: Routing in mobile ad-hoc cognitive radio networks. Presented in Wireless Days. venice,

Friend DH. 2009.Cognitive Networks: Foundations to Applications. PhD dissertation. Virginia Polytechnic Institute and State University, Virginia,

Garg VK. 2007.Wireless communications and networking. Morgan Kaufmann Pub.

Lee M, Marconett D, Ye X, Yoo S J. 2007.Cognitive network management with reinforcement learning for wireless mesh networks. Presented in Proceedings of the 7th IEEE international conference on IP operations and management, San Jos, USA,

Liu L, Hu G, Xu M, Peng Y. 2010.Learning automata based spectrum allocation in cognitive networks. Presented in International Conference on Wireless Communications, Networking and Information Security. Beijing, China,

Mahmoud Q H, 2007. Cognitive networks. Wiley Online Library,.

Marconett D, Lee M, Ye X, Vemuri R, Yoo SJB. 2012. Self-adapting protocol tuning for multi-hop wireless networks using Q-learning.International Journal of Network Management. 21(2): 12–24.

MitolaJ. 2000. Cognitive radio: an integrated agent architecture for software defined radio, PhD dissertation. Royal Institute of Technology (KTH), Stockholm, Sweden.

Mitola J, Maguire GQ. 1999. Cognitive radio: making software radios more personal. Personal Communications IEEE 6(4):13–18.

Narendra KS, Thathachar MAL. 1989. Learning automata: an introduction. Prentice-Hall, Inc.

Perkins CE, Royer EM. 2002.Ad-hoc on-demand distance vector routing. presented in Mobile Computing Systems and Applications, 1999. Presented in proceedings WMCSA'99. Second IEEE Workshop on,

Sastry PS. 2004. Networks of learning automata: Techniques for online stochastic optimization. Kluwer Academic Publishers.

Sobeih A,Chen WP, Hou JC, Kung LC, Li N, Lim H, Tyan HY, Zhang H. 2005.J-sim: A simulation environment for wireless sensor networks. Presentedin Proceedings of the 38th annual Symposium on Simulation. San Diego, CA, USA,

Song Y, Fang Y, Zhang Y. 2007.Stochastic channel selection in cognitive radio networks. Presented in IEEE Global Telecommunications Conference. Washington, D.C,

Song Y, Zhang C, Fang Y. 2010. Stochastic traffic engineering in multihop cognitive wireless mesh networks. IEEE Transactions on Mobile Computing 9(3): 305–316.

Thomas RW. 2007.Cognitive Networks. PhD dissertation. Virginia Tech, Blacksburg, Virginia,

Thomas RW, Friend DH, DaSilva LA, MacKenzie AB. 2007. Cognitive networks: adaptation and learning to achieve end-to-end performance objectives. Communications Magazine IEEE 44(12): 51–57.

Zarei MM, Faez K, Moosavi Nya J, Abbaszadeh Meinagh M. 2008.Route Stability estimation in Mobile Ad Hoc Networks using Learning Automata. Presented in 16th Telecommunications forum. belgrade, Serbia,