Computers and Electrical Engineering 72 (2018) 79–91

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

An adaptive channel assignment in wireless mesh network: The learning automata approach☆

Ziaeddin Beheshtifard[a,*], Mohammad Reza Meybodi[b]

[a] Department of Computer Engineering and Information Technology, Qazvin Branch, Islamic Azad University,Qazvin, Iran
[b] Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 4 June 2018
Revised 2 September 2018
Accepted 4 September 2018

Keywords:
Wireless mesh network
Multi-radio
Channel assignment
Learning automata

ABSTRACT

In wireless mesh networks, random changes in the environment can increase the complexity of the multi-channel assignment. In this work, a new channel assignment scheme based on learning automata is proposed, which adaptively improves the network's overall performance by predicting network dynamics. First, we use a practical utility function that reflected the user's preference regarding the signal-to-interference-and-noise ratio is applied. In the multi-automata learning algorithm, each user evaluates a channel selection strategy by computing a utility value in a stochastic iterative procedure. The utility function that potentially reflects a measure of satisfaction is used by every node as an environmental response to the current selected strategy. In the proposed algorithm, by changing network traffic pattern, the channel allocation varies adaptively with dynamic conditions of the network. Extensive simulation-based evaluation of our algorithm demonstrates that the proposed algorithm converges to an equilibrium point, which is also optimal for channel assignment policy.

1. Introduction

Wireless Mesh Network (WMN) is an emerging new technology that is in the process of being adopted as the wireless Internetworking solution in the near future. Characteristics of WMN—such as rapid deployment and self-configuration—make it suitable for transient on-demand network deployment scenarios such as disaster recovery, hard-to-wire buildings, conventional networks, and friendly terrains.

WMN architecture can be established from mesh clients and mesh routers. The mesh clients are often laptops, cell phones, and other wireless devices—either fixed or mobile. The mesh routers make a multi-hop infrastructure for wireless backbone, to ensure forward traffic from the mesh clients to the gateways to provide connectivity to the Internet. The mesh routers relay all the wireless network traffic on behalf of clients that may be outside of the communication range of their destinations. Some of the mesh routers act as a gateway to provide connectivity to the Internet or other wired networks. This functionality in mesh gateways increases the interoperability of wireless mesh networks and other types of wireless networks, such as cellular networks, MANETs, WSNs, Wi-Fi, and WiMAX.

Simultaneous communication on 802.11 radios may be lead to capacity degradation when the operation is within the same interference range [1]. Fortunately, the IEEE 802.11 specification in physical layer makes multiple orthogonal channels

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Associate Editor Dr. Hong Shen.

* Corresponding author.
E-mail addresses: beheshti@qiau.ac.ir (Z. Beheshtifard), mmeybodi@aut.ac.ir (M.R. Meybodi).

https://doi.org/10.1016/j.compeleceng.2018.09.004
0045-7906/

available. For example, the 802.11b/g standard provides three such channels while the 802.11a standards allow up to 12 non-overlapping frequency channels that can be used simultaneously within the neighborhood. Accordingly, to achieve high throughput, wireless mesh routers can be equipped with multiple radios and operate on multiple orthogonal frequency channels which arises to the advantage of a self-managing and high capacity wireless mesh network [2].

In the context of learning, learning automata is an adaptive, distributed approach to solve a wide range of engineering problems, especially in communication networks in which the automata are located at the network's nodes. Learning automata—as a stochastic optimization framework—has been used to model the uncertain aspects of network dynamics. This mechanism has been widely used to predict the unknown probability distribution in some challenging problems, such as dynamic channel assignment [3], link scheduling [4], and routing [5]. In this paper, some network characteristics related to the performance and the network feedback used to train the learning automata are represented by an abstract environment model for which only a few functional properties are assumed. An environment model in which the penalty probabilities are unchanged is unsatisfactory, because the observed evolution of the process is not memoryless.

To the best of our knowledge, this work is the first study on utilizing learning method for multi-radio channel assignment under the Signal-to-Interference-Plus-Noise-Ratio (SINR) model for interference that can be implemented in a distributed fashion. In this work, a game of automata-based channel assignment scheme that does not require any knowledge about network topology and configurations, and that can be implemented in a fully distributed fashion under a more realistic model of physical interference, was developed and adapted. The rest of the paper is organized as follows: Related studies on the channel assignment problem are summarized in Section 2. Sections 3 and 4 focus on the system model, which consists of the stochastic learning method used in learning automata and proposes a distributed multi-radio channel assignment based on leaning automata that gradually learn the stochastic behavior of the network. The final section provides some simulation result and conclusions.

2. Channel assignment problem

Channel access mechanisms provided by the MAC layer protocol are developed for coordinating access to the shared medium and alleviating conflicts. These methods, especially in wireless networks, are schedule based such as frequency division multiple access (FDMA), time division multiple access (TDMA), and code division multiple access (CDMA) or contention-based, such as carrier-sensing multiple access with collision avoidance (CSMA/CA).

Using more than one radio in each wireless mesh router is an effective approach for improving the performance of wireless mesh networks. IEEE 802.11a provides 12 non-overlapping channels that can be used simultaneously for communication within a neighborhood when different radios are tuned to use non-overlapping channels [IEEE]. The more radios are assigned to the same channel, the more connectivity is achieved, but the more interference is induced as well [6]. Consequently, increase in connectivity and decrease in interference tradeoff should be balanced.

Channel allocation is faced with some constraints. First, two communicating nodes require a shared distinct channel to establish a communication link. Second, the traffic load imposed on the wireless links should not exceed the nominal capacity of the channel. Third, the number of available channels is fixed. Fourth, the number of channels used by each node is limited to the number of radio transmitters that belong to that node [7]. According to these restrictions, it has proven that the problem of optimal channel assignment can be mapped to multi-constraint graph coloring problem and arises as NP-Hard problem [8].

Channel assignment schemes can be classified in terms of the implementation as centralized and distributed. Most of the centralized approaches follow the graph model to solve multi-constraint graph coloring problem. Marina [9] has developed a topology control approach called "Connected Low Interference Channel Assignment (CLICA)." This approach uses a greedy heuristic algorithm based on unit disk graph to enable a feasible topology formation control. Although this approach ensures the network connectivity and avoids link revisits, it does not take into consideration the impact of traffic on optimal channel assignment. Tang et al. propose the use of INSTC as an interference-aware channel assignment and routing algorithm. In their study, the authors formally define and present an effective heuristic for the minimum interference survivable topology control (INSTC) problem, which seeks a channel assignment for the given network, such that the induced network topology is interference-minimum among all K-connected topologies of the given unit disk graph [10]. Like CLICA, INSTC suffers from sacrificed fairness problem when it chooses a channel for a wireless link in a locally optimal manner. C-HYA—another centralized channel assignment scheme for wireless mesh networks based on Hyacinth—is proposed in Raniwala et al. [8]. Their proposed algorithm assigns channels to ensure network connectivity and meet the capacity limitation of the link. It estimates the expected traffic load on each wireless link based on current and past offered traffic. Then, it traverses links in decreasing order of estimated traffic load and assigns channel in a greedy manner. Although this scheme presents a method for channel allocation that incorporates connectivity and traffic patterns, the assignment of channels on links may cause a ripple effect.

In dynamic assignment, channels are assigned to radios for a short time period; the radio interfaces should frequently switch between channels per-packet or per-flow [11]. In this paper, each node uses a pseudo-random sequence mechanism to switch channels synchronously, so that all neighbors meet periodically in the same channel. Another distributed and dynamic channel assignment algorithm called D-HYA is proposed in Raniwala et al. [12]. This algorithm assigns channels according to traffic load, in order to achieve load balancing and improve aggregate throughput.

In recent studies, joint multi-radio channel assignment with power control and routing protocol were proposed in [13]. The authors proposed a heuristic method called MP-CARA for the minimum power channel assignment and routing problem. MP-CARA is a centralized algorithm and requires full knowledge of the current network status, such that it requires a large amount of message passing between nodes when there is the need to be implemented in the real world. Another recent study on channel assignment coupled with power control [14], the authors of the paper proposed a multi-channel multi-radio assignment algorithm to control network topology and network connectivity. In this method, using the estimation of the traffic load and the interference of each wireless link, the channel is assigned with the minimum interference factor to the radio interfaces. Although this method has better end-to-end latency compared to other similar methods, because of the greedy approach in channel selection with minimum interference, it is not throughput optimal.

In terms of learning method, the present work is a generalization of previous study based on stochastic channel assignment [3]. In terms of channel assignment, this work is similar to the approach presented in Pediaditaki et al. [15] but different in terms of channel state expectation. In Pediaditaki et al. [15], the authors propose an approach in which each mesh node discovers its neighbors and the channel usage in its neighborhood periodically. In the proposed scheme, the nodes do not need to pass messages to neighbors, and independently expect channel state according to the ability of packet transmission. Another heuristic method for joint channel assignment and scheduling presented in Selvakumar et al. [16]. A Tabu search optimization framework used for minimizing delay by traffic scheduling enhancing the quality of services, but their algorithm is based on the simplistic graph based interference model.

In this paper, we propose a new method for dynamic channel assignment that can be used in physical interference model be implanted in a distributed fashion. In our approach, we exploit a novel interference estimation method by following a measurement approach that can be used as the network response to the channel selection strategy. This method facilitates the use of the channel assignment algorithm in SINR interference model of the real world. Our experiments show the stability of the network when the efficient learning-automata-based channel assignment is applied, as well as the adaptability of the network behavior when the traffic pattern in network flows are changed.

3. System model

3.1. Network model

In this section, we develop a mathematical formulation of the problem of channel assignment. Also, a brief description of learning automata as a stochastic optimization framework is considered. Suppose a wireless mesh network with $\mathcal{N} = \{\mathrm{n}_1, \mathrm{n}_2, \ldots, \mathrm{n}_N\}$ as a set of nodes and $\mathcal{L} = \{\mathrm{l} : \mathrm{l} = (\mathrm{n_i}, \mathrm{n_j}), \mathrm{n_i}, \mathrm{n_j} \in \mathcal{N}\}$ is a set of links in the wireless mesh backbone. $C = \{c_1, c_2, \ldots, c_k\}$ is a set of available channels and $k = |C|$ is the number of available channels that could be assigned to radio interfaces. Also, some end-to-end communication between source and destination nodes are represented by a set of flows $\mathcal{F} = \{f_1, f_2, \ldots, f_F\}$, when data traffic for every flow f_i may be routed over different paths. Each link l uses multiple queues associated with the flows that pass it. Each queue—represented by $Q_l^d(t)$—denotes the number of packets queued at link l for destination d at the beginning of time slot t. The evolution of the queue for link l and for destination d may be written as:

$$Q_l^d(t+1) = \left[Q_l^d(t) - S_l^d(t)\right]^+ + A_l^d(t) \tag{1}$$

where S is the served traffic in link l and A is the generated packets at time t in link l to destination d. A queuing system is called stable if for any feasible rate, the average queue-size is bounded over time as $\lim_{t\to\infty} \sup_t (E[Q_l(t)]) < \infty$.

3.2. Utility maximization

The user satisfaction with the provided service can be expressed using utility functions that represent the degree of satisfaction of the user function of the rate allocated by the wireless network [17]. This function strongly depends on network resources capacity. The capacity region of wireless networks is defined as the closure of the set of all arrival rates of flows that are stably supportable by the network [18]. Two aspects of capacity regions are fundamental to our problem formulation: the channel capacity region at the MAC and physical layer, and the routing capacity region at the network layer. The channel capacity region H defines a set of (c, h) such that channel assignment in h can support link capacity vector c. The routing region R defines a set of (x, f) such that the throughput vector x can be supported by flow rate in f. The overall utility optimization framework are not relevant to the details of the channel capacity region so the details of them are discussed in its own context.

We follow the parameterized family of utility functions as *α-fair* often used in the literature and the utility maximization problem can be formulated as:

$$U(x) = \begin{cases} \frac{x^{1-\alpha}}{1-\alpha} & \alpha > 0,\ \alpha \neq 1 \\ \ln(x) & \alpha = 1 \end{cases} \tag{2}$$

$$\begin{aligned} &\text{Maximize } \sum_{s \in S} U_s(x_s) \\ &\text{Subject to : } \left(\sum_{s: l \in s} x_s \right) \le c_l \ \forall l \in L \\ &\quad c \in H \end{aligned} \tag{3}$$

where S is the set of traffic sources, x_s is the information rate that generated by user s, and c is the channel capacity. The first constraint models link capacity constraints such that the total traffic injected to link l that uses channel c cannot exceed channel-nominated capacity c_l. The second constraint models the dependence of effective channel bandwidth on channel assignment at each node.

The overall solution framework proposed for solving (3) is an iterative primal-dual schema that switches between solving primal sub-problems and updating dual variables. A critical observation of the optimization problem (3) is that the channel region H and the routing region characterize variables from the MAC/PHY layer and the network layer respectively, and are relatively independent. The only coupling constraint between them is $\boldsymbol{x} \le \boldsymbol{c}$. To remove the $\boldsymbol{x} \le \boldsymbol{c}$ from the constraint set we can apply the Lagrangian relaxation technique [19]. Let p_l be the Lagrange multiplier associated with the capacity constraint at link l and let p denote the vector of Lagrange multipliers. Then, the Lagrangian is given by:

$$\boldsymbol{L}(\boldsymbol{U}, \boldsymbol{p}) = \sum_{\boldsymbol{s} \in \boldsymbol{S}} \boldsymbol{U_s}(\boldsymbol{x_s}) - \sum_{\boldsymbol{l} \in \mathcal{L}} \boldsymbol{p_l} \left(\sum_{\boldsymbol{s} \in \boldsymbol{S}} \boldsymbol{x_s} - \boldsymbol{c_l} \right) \tag{4}$$

The resulting optimization problem is naturally decomposed into two smaller sub-problems, including the Channel Assignment Sub-problem at the MAC/PHY layer:

$$\begin{aligned} &\text{Maximize } \sum_{l \in \mathcal{L}} c_l \\ &\text{Subject to : } c \in H \end{aligned} \tag{5}$$

and the Routing Sub-problem at the network layer:

$$\begin{aligned} &\text{Maximize } \sum_{s \in S} U_s(x_s) - \sum_{l \in \mathcal{L}} p_l \left(\sum_{s \in S} x_s - c_l \right) \\ &\text{Subject to : } \left(\sum_{s: l \in s} x_s \right) \le c \end{aligned} \tag{6}$$

where the p_l's are the Lagrange multipliers.

The routing problem at the network layer has been extensively studied in the literature during the past decade. According to our formulation, the Routing Sub-problem easily can be solved by a Lagrangian relaxation based back-pressure algorithm [20]. We remark that p_l has the same dynamics of the queue length at the link l. It increases at rate $\Sigma_{s:\ l \in s} x_{sl}$, which is the arrival rate, and decreases at rate c_l, the capacity of the link. Thus, the links do not have to compute their dual variables and they are simply the queue lengths and packets of the link with maximum differential backlog can be served.

3.3. The channel assignment sub-problem

In this section, we construct a detailed model of channel region that is affected by the channel assignment scheme. As mentioned earlier, total link rate is bounded by the channel capacity R_l. The channel assignment sub-problem attempts to maximize channel capacity R_l, but despite the wired networks, link capacity in wireless networks entirely depends on other nodes' dynamics and cannot be determined earlier. Also, channel capacity is highly dependent on realistic physical factors of the wireless links, such as signal propagation model and interference model. The propagation model seeks to capture how a signal attenuates as it travels in the air. This model can be formulated by a decreasing path loss function of signal strength (P_s), depending on the distance between the sender (n_s) and receiver (n_r) nodes:

$$P_r = \frac{P_s}{d(n_s + n_r)^{\alpha}} \tag{7}$$

where α is the path-loss exponent, a constant dependent on the medium, typically between 2 and 6 [21].

The propagation model leads to a physical interference model that can be formulated as $\frac{P_r}{I_r + N}$, when P_r is the signal strength at the receiver n_r, I_r is the accumulated interference that is generated by other senders such can be sensed by n_r, and N is the ambient noise. More precisely, according to signal-to-interference-and-noise ratio (*SINR*), a message from a transmitter of link l can be successfully decoded by a receiver of l when the received *SINR* is greater than a hardware-dependent ratio β.

$$SINR = \frac{G_{ll} P_l}{\sum_{l' \neq l} G_{ll'} P_{l'} + N_{amb}} \tag{8}$$

where $G = [G_{ll'}]$ is a probabilistic gain matrix and $G_{ll'}$ is the power gain from the transmitter on link l' to the receiver on link l. Finally, according to the Shannon–Hartley theorem, effective link capacity function is assumed to be of the form [22]:

$$R_l = \log\left(1 + \frac{\phi K G_{ll} P_l}{\sum_{l' \neq l} G_{ll'} P_{l'} + N_{amb}}\right) \tag{9}$$

where φ is the coding gain, $K = -1.5/\log(5BER)$, is an error rate scaling factor, and N_{amb} is the ambient noise. The probability distribution of the gain matrix G depends on the stochastic behavior of the network elements. The link capacity can vary randomly over time, thereby resulting in congestion and queuing delay at the link buffers [23].

Our channel assignment algorithm should maximize potential link capacity R_l. Thus, with fixed power level usage, it should minimize the effect of interfering links ($\sum_{l' \neq l} G_{ll'} P_{l'}$) with appropriate channel assignment scheme. Thus, in our algorithm, SINR can be used as channel assignment utility function (u^c). Maximizing this factor leads to maximization of the channel rate; consequently, the channel capacity region expands. The channel assignment utility function u^cshould be computed in the receiving node; but unfortunately, nodes cannot use Eq. (8) directly to compute SINR. There are two impediments. The first is that the fading property of the radio signals at the receiving nodes is a hopelessly complicated function that depends on many physical properties of the environment. As described above (7), to avoid this complexity, several researchers use a simplified model that calculates attenuation based on the distance between the nodes. The second difficulty is that the parameters of the SINR model cannot be obtained by directly using commodity hardware. We need the strength of the incoming signal ($G_{ll}.P_l$) and the distribution of interfered signals($G_{ll'} P_{l'}$). However, the parameter that can widely be measured across wireless cards is the reported received signal strength indicator (RSSI) value. RSSI is a measure of the energy at the receiver during decoding, and so conflates signal strength and interference: RSSI = Signal + Interference. Additionally, RSSIs are available only when packets are received successfully, which increases the difficulties of estimating the interference around the nodes.

Despite these problems, we can use measurements to leverage the SINR model. The exact form varies with receivers and network deployments, but we still expect the signal strength, interference, and noise relationships to be consistent with the SINR model. To capture these relationships, we use a measurement method that is similar to what is described in Reis et al. [24] to estimate SINR parameters. The measured RSSI values allow each receiver to compute several parameters: First, the mean RSSI for packets received at r from another node s, which we denote as $\overline{P_l}$, can easily be computed by averaging. Second, we can also estimate the average external interference at a node r, which we denote $\bar{I}$, from our earlier observation that most of the variations in RSSI values stems from interference. If we assume that at least one packet across all senders was received when the external interference was almost zero, $\bar{I}$ can be estimated by the mean excess of the RSSI values from individual senders above their minimum observed values.

3.4. Learning automata

A learning automaton is an adaptive stochastic control mechanism that improves its action through repeated interactions with the dynamic environment. We focus on a more flexible learning automaton model called variable structure stochastic learning automata [25]. In this model, the action probabilities are updated at every stage using a reinforcement scheme. According to action probability distribution, one action is randomly selected. Then, based on the observed environment response, the automaton updates its action probabilities. In a repeated way, the learning automaton gradually learns which appropriate action should be selected when a specific stimulus is received from the environment.

The environment can be described by a triple $\boldsymbol{E} = \{\underline{\boldsymbol{\alpha}}, \underline{\boldsymbol{c}}, \underline{\boldsymbol{\beta}}\}$, where $\underline{\boldsymbol{\alpha}} = \{\alpha_1, \ \alpha_2, \ldots, \alpha_r\}$ denotes the finite set of inputs, $\underline{\boldsymbol{\beta}} = \{\beta_1, \beta_2, \ldots, \beta_m\}$ is the set of the reinforcement signal taken from the environment and $\underline{\boldsymbol{c}} = \{c_1, c_2, \ \ldots, c_r\}$ denotes the set of the penalty probabilities. Environments can be classified into P-model, Q-model, and S-model according to the nature of the reinforcement signal $\boldsymbol{\beta}$. The environments in which the reinforcement signal can only take binary values—0 and 1—are referred to as *P*-model environments. Another class of environment allows a finite number of the values in the interval [0, 1], which can be taken by the reinforcement signal. Such an environment is referred to as *Q*-model environment. In *S*-model environments, the reinforcement signal lies in the interval [*a, b*]. The random environment is said to be a non-stationary environment if the penalty probabilities vary over time, and is called stationary otherwise. Later we discuss more about our wireless network model as a non-stationary environment.

We define the learning environment of the system constructed with a network of *learning units* say LU_1, LU_2, … LU_n, so that each LU corresponds to a multi-radio wireless node. Each LU is equipped with an automaton as shown in Fig. 1. Formally, every learning unit can be described by $\mathcal{A}_i = (A_i, B_i, \mathcal{T}_i, p_i(k))$, where $A_i = \{\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{ir}\}$ is the finite set of actions, $B_i = \{\beta_i(k) \in [0, 1]\}$ is the set of possible input or reinforcement to the $\mathcal{A}_i$ as a random payoff, $\mathcal{T}_i$ is the learning algorithm for updating action probabilities, and $\boldsymbol{p}_i(k) = [p(k), p_{i2}(k), \ldots, p_{ir}(k)]^T$ is the action probability vector at time instant k, where $p_{ij}(k) = Prob[a_i(k) = \alpha_{ij}]$ and a_{ij} is the jth action of $\mathcal{A}_i$.

Let β_i be the payoff to the ith LU player. We define functions:

$$F^i : \prod_{i=1}^{N} A_i \to [0, 1], \ \ by$$

$F^i(\alpha) = E[\beta_i | i\text{th } LU\ choose\ \boldsymbol{\alpha_i}, \boldsymbol{\alpha_i} \in A_i] F^i(\alpha)$ is called the payoff for player i. The player receives the payoff signal (β_i).

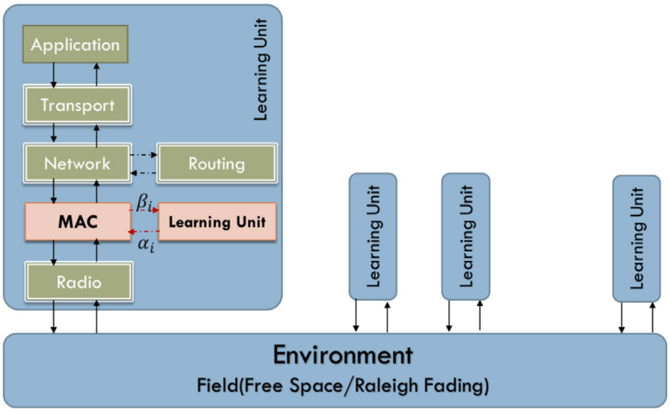

Fig. 1. A network of learning units constructed by a team of LA's.

Definition 1. The vector $\alpha^* = (\alpha_1^*,\ \alpha_2^*,\ \ldots, \alpha_N^*)$ is an optimum strategy of the game if for each player i, $1 \le i \le N$ $F^i(\alpha^*) > F^i(\alpha)$ for all $\alpha = (\alpha_1^*, \ldots, \alpha_{i-1}^*, \alpha_i, \alpha_{i+1}^*, \ldots, \alpha_N^*)$, such that $\alpha_i \neq \alpha_i^*,\ \alpha_i \in A_i$.

Remark 1. In the aforementioned definition, α^* is a Nash equilibrium of the payoff matrix $F(.)$.

At every timeslot n, the learning unit (automaton) uses the reinforcement scheme to perform some action $(a(n))$ on the environment. The environment responds to each action with β_i, that which may be either positive or negative. This response serves as the input to the automaton and the probability that the action being performed is rewarded or penalized depends on the type of this response. According to the new probability vector of the actions, the learning unit computes the next action $(a(n+1))$ to be performed on the environment.

Let $\boldsymbol{p_i}(k) = [p_{i1}(k), \ldots, p_{ir_i}(k)]$, $1 \le i \le N$, denote the action probability distribution of ith automata, where $p_{ij}(k) = \text{Prob}(\alpha_i(k) = \alpha_{ij}]$ and a_{ij} is the jth action of ith automata. At cycle k, the ith automaton selects action $\alpha_j(k) \in A_j$ according to the probability distribution $\boldsymbol{p_i}(k)$. Then, the ith learning unit (containing automata A_l receives reinforcement $\beta_l(k)$ from the environment, which is the response to the action $\alpha_l(k)$. Finally, every automaton updates its action probability vector using the *Linear Reward-Penalty* (L_{R-P}) algorithm as follows:

$$p_i(k+1) = \begin{cases} p_i(k) - a(1-\beta(k))p_i(k) + b\beta(k)\left(\frac{1}{r-1} - p_i(k)\right), & a(k) \neq a_i \\ p_i(k) + a(1-\beta(k))(1-p_i(k)) - b\beta(k)p_i(k), & a(k) = a_i \end{cases} \tag{10}$$

where a and b are the reward and penalty parameters respectively and $\beta\,(k)$ is the response of the environment when the action is selected at the instance.

4. Channel assignment algorithm

In the multiple payoff game, each node acts as a player or learning unit that participates in the channel assignment game. According to the defined channel utility function as a satisfaction factor of the player, the objective of each player is to maximize its payoff, which can be measured by its channel utility function. The only information that is available for the player is its payoff after each play. Each learning unit as a player does not have any knowledge about the strategies used by other players or the environment response for possible play. Therefore, in non-cooperative channel assignment, each player (node) maximizes its channel utility in a fully distributed fashion, which can be expressed as:

$$\underset{S_i \in \boldsymbol{S_i}}{\text{Maximize}}\ u_i^c(S_i,\ \boldsymbol{I}_{-i}) \text{ for all } i \quad \in N$$

where S_i is the measured signal at the receiver and $\boldsymbol{I}_{-i}$ is the estimated interference imposed by the other sending nodes. The solution of this problem gives a Nash equilibrium. In a Nash equilibrium, given the assigned channels of another player, no player can improve its channel utility level by making individual changes in selected channels.

A distributed channel assignment algorithm is proposed that utilizes the learning automata able to adaptively assign channels to radio interfaces. In this algorithm, each learning unit takes an action on every timeslot τ. The action sets of each learning unit are constructed by a combination of M_i(The number of available radios for node i) element subset from a C (number of available channels) element set. A node may potentially switch channels after sending each packet, but this results in performance loss due to the overhead of channel switching delay and LEAVE/JOIN message overhead. Therefore, time is divided into two different interval scales that we called *Frame slots* and *Time slots* (Fig. 2). Channel assignment algorithms perform repeatedly in each frame slot. At the beginning of the frame slot, selected channels set for each node are updated according to observed, and estimated *SINR* as their environmental measured payoff.

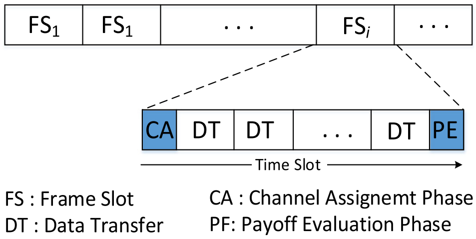

Fig. 2. Frame slot and time slot configuration for channel assignment, data transfer and payoff evaluation.

Algorithm 1 Distributed Channel Assignment (*LACA*).

Step 1 Parameters definition

1. δ: Threshold for terminating algorithm
2. C: Number of available channels
3. M_i:Number of available radios for node i
4. A_i: Action set for channel assignment automaton in LU_i

Step 2 Initialization

1. $A_i =$ Set of M_i-subset of C, $|A_i| = \binom{C}{M_i}$,
2. $p_i(0) = [p_{i,j}(0) = \frac{1}{|A_i|} : j \in \{1, 2, \dots, |A_i|\}]$

Step 3 Search

1. Repeat for timeslot $k = 1,\ 2,\ \dots$
2. Pick an action $\alpha_i(k)$ according to its action probability vector $\boldsymbol{p}_i(k)$.
3. Each player assigns the corresponding channel set to its radios.
4. Each node j as a receiver compute *SINR* according to estimated signal strength $\widetilde{S}_j^i$ and $\widetilde{I}_j^i$ and send back them via *ACK* message to sender i as β_j^i.
5. Each player obtains a payoff $\overline{\beta}_i(k)$ on the all of selected actions, and then normalize $\overline{\beta}_i$ as

$$\boldsymbol{u_i}(\boldsymbol{k}) = \begin{cases} \overline{\boldsymbol{\beta}}_{\boldsymbol{i}}(\boldsymbol{k})\ \boldsymbol{if}\ \max(\overline{\boldsymbol{\beta}}_{\boldsymbol{i}}) = \min(\overline{\boldsymbol{\beta}}_{\boldsymbol{i}}) \\ \frac{\overline{\boldsymbol{\beta}}_{\boldsymbol{i}}(\boldsymbol{k}) - \min(\overline{\boldsymbol{\beta}}_{\boldsymbol{i}})}{\max(\overline{\boldsymbol{\beta}}_{\boldsymbol{i}}) - \boldsymbol{min}(\overline{\boldsymbol{\beta}}_{\boldsymbol{i}})}\ \boldsymbol{o}.\boldsymbol{w}. \end{cases} \tag{11}$$

6. Each player updates its action probability vector for automata $\mathcal{A}_i$ for each element j of its probability vector such that:

$$\boldsymbol{p_j}(\boldsymbol{k}+1) = \begin{cases} \boldsymbol{p_j}(\boldsymbol{k}) - a\boldsymbol{u_j}(\boldsymbol{k})\boldsymbol{p_j}(\boldsymbol{k}) + b(1 - \boldsymbol{u_j}(\boldsymbol{k}))(\frac{1}{\boldsymbol{r}-1} - \boldsymbol{p_j}(\boldsymbol{k})),\ a(k) \neq \boldsymbol{a_j} \\ \boldsymbol{p_j}(\boldsymbol{k}) + a\boldsymbol{u_j}(\boldsymbol{k})(1 - \boldsymbol{p_j}(\boldsymbol{k})) - b(1 - \boldsymbol{u_j}(\boldsymbol{k}))\boldsymbol{p_j}(\boldsymbol{k}),\ a(k) = \boldsymbol{a_j} \end{cases} \tag{12}$$

7. If $\varepsilon = (\|\boldsymbol{p}_i(k+1) - \boldsymbol{p}_i(k)\|_1) < \delta$ stop the algorithm; else go to step 3.1
8. end

The minimum and maximum values of utility function are determined dynamically for normalizing the utility function to lie in the interval (0, 1), but the realistic value for the maximum and minimum cannot be determined in advance. In other words, the current $\max(\overline{\beta}_i), \min(\overline{\beta}_i)$ are approximation values for actual maximum and minimum. Alternatively, a zero can be chosen for $\min(\overline{\beta}_i)$ and a large constant value for $\max(\overline{\beta}_i)$, but choosing a very large value for maximum compared to average utility values will significantly decrease the speed of convergence.

In the proposed algorithm, each node tries to improve its utility function by maximizing the normalized payoff via current assigned channels. This algorithm implements a non-cooperative game that nodes gradually learn which channel-set is the best to improve their utility function. According to linear reward-penalty scheme (L_{R-P}) reflected in the update Eq. (12), when a node picks a channel set as its action, and when a chosen channel set results in a reward, then the probability of choosing that action in the next time slot is updated, if not, no change takes place.

It should be noted that the learning automata based channel assignment acts as a distributed algorithm such that each node independently select an action in each timeslot and do not need any knowledge about other nodes and their actions. Therefore the proposed algorithm does not depend on the network topology or network size. The computational complexity of the algorithm is inversely proportional to the learning rate, in order to reduce the high cost of the network, the learning rate should be maximized. Therefore, to make a balance between the computational complexity and the optimal channel allocation, the learning rate must be carefully selected. This can be chosen empirically or based on the theoretical framework. Narendra and Thathachar [25], using Martingale's convergence theorem, specified a lower bound on the probability of convergence to the optimal action for each learning rate. They showed that by appropriate choice of learning rate, the lower bound on the probability of the convergence to the optimal action is close to unity. According to similar framework for learning, it can be shown that the same results hold true for our proposed channel assignment algorithm.

5. Experimental results

In order to study the quality and performance of the proposed channel assignment algorithm, several scenarios were considered through detailed numerical studies. Some wireless network parameters in the physical layer were considered as follows. We used free space path loss model for the attenuation of the signal strength. Also, Rayleigh distribution was

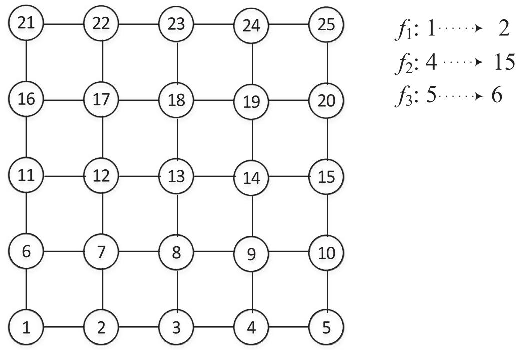

Fig. 3. Grid style topology used in simulation.

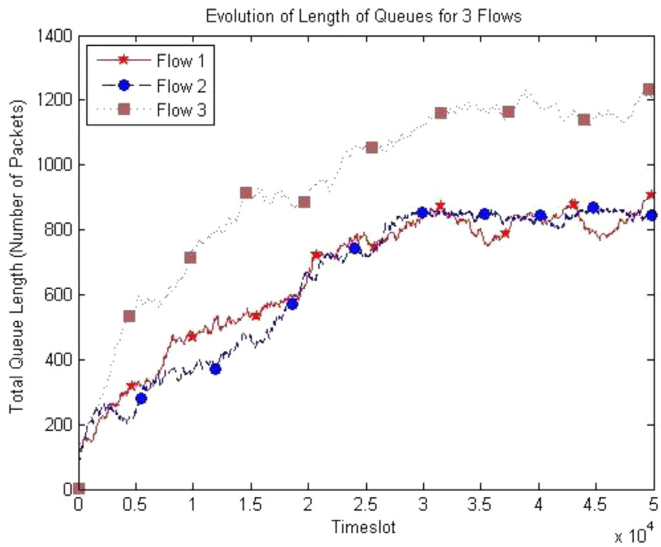

Fig. 4. Queued backlog evolutions for the flows.

assumed to be a statistical model for the effect of a propagation environment. All of the nodes in the mesh backbone are fixed, while client nodes may be mobile. Default signal frequency, signal reception sensitivity, and signal reception threshold were subsequently assumed as 2.4 GHz, −91 dBm, and −81 dBm. For simplicity, background noise was set to −50 dBm.

We conducted four series of experiments to evaluate different aspects of our algorithm. In the first series, we evaluated the stability of the algorithm through study of evolution of queued backlog. For these experiments, we used a topology in which all nodes were placed at the same distance from each other. For this proposal, 25(5×5) wireless nodes were placed in a square region in the area of 2500×2500 m square. The network topology used in the present experiment is a grid style topology, as depicted in Fig. 3. Each node was equipped with a learning automaton, which evaluated the response of the environment according to measured utility, as defined by Eq. (1).

In the first scenario, we studied the behavior of three concurrent flows. It was assumed that there were three simultaneous flows: F_1 from node 1 to 2, F_2 from 4 to 15, and F_3 from 5 to 6. Each node, through its learning automaton and according to actions probability distribution, chooses its action by selecting $m = 2$ channel from $K = 10$ available orthogonal channels at the beginning of frame slot. After channel assignment, nodes start packet transmission according to the three specified traffic flows in some consecutives timeslot. Normalized utility as environment payoff is measured by the nodes at the end of every frame slot. Subsequently, nodes evaluate the response of the environment to last channel-set allocation and reward/penalize their automata by updating action probabilities. First, we are interested in the evolution of the average length of the queues and throughput for each flow. The simulation result for the evolution of the averaged backlog and throughput improvement for each flow was evaluated in this experiment. Figs. 4 and 5 show that the total number of the packets and throughput for each flow tends to a constant value in steady state; this result corroborates the stability of the system. The second result of the observation is that according to separation of the source and destination, different levels of throughput in steady state are observed; the throughputs of the flows are converged to completely different values. For

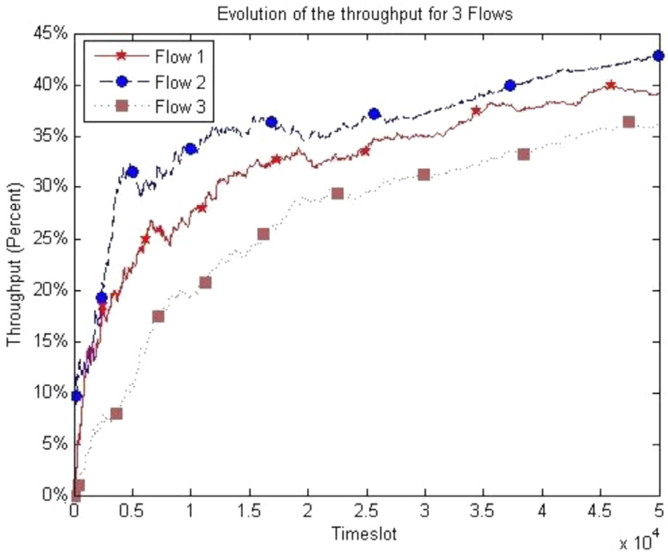

Fig. 5. Throughput evolutions for the flows.

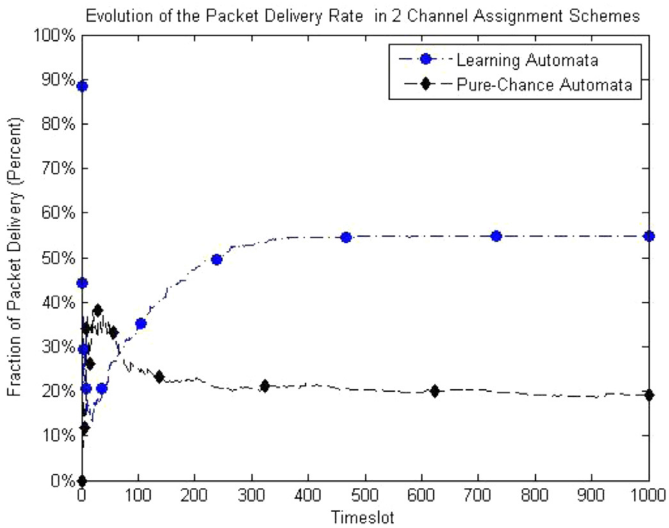

Fig. 6. Evolution of the packet timeslot rate across the timeslot.

example, the minimum level of throughput is attained for the longest hop flow F_3 compared to F_2, which achieved the highest throughput with hop length of only 3.

Despite the short time fluctuation in the beginning of the algorithm, network elements effectively learn how to pick suitable channels set to their radio interfaces, which leads to more connectivity and low interference, which in turn results in more packet delivery in the network. Considering these observations, it can be concluded that the channel assignment scheme based on the learning automata can insure the stability of total backlogs related to associated packet queues.

In the second case, the performance of our proposed channel assignment scheme based on the learning automata and the pure-chance automata were compared. Despite the pure-chance approach which blindly assigns channels to radios, the proposed algorithm gradually learns from the environment to assign appropriate channel. The efficiency of the algorithm is depicted in Figs. 6 and 7. According to Fig. 6, the packet delivery rate reaches to above 50% of total sent packets in the learning automata based channel assignment, but this is about 20% in the pure-chance scheme. As well as the Fig. 7 showed that the total length of the queues changed quickly and converged to a steady state value as compared to the pure-chance algorithm. Although the total backlog value in the steady state behavior was significantly lower than the pure-chance automata scheme, indicating that in comparison with this scheme, the greater number of packets to the destination was reached. This was due to the full utilization of all resources against blind assigning of channels to radios in pure-chance automata

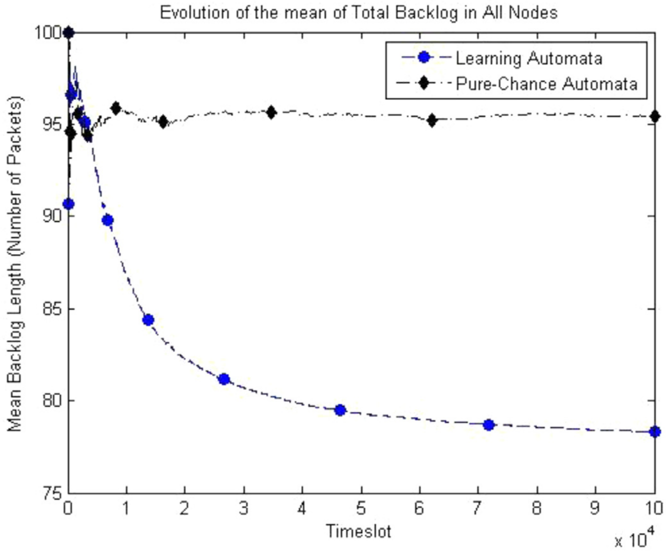

Fig. 7. Evolution of the mean of backlog across the timeslot.

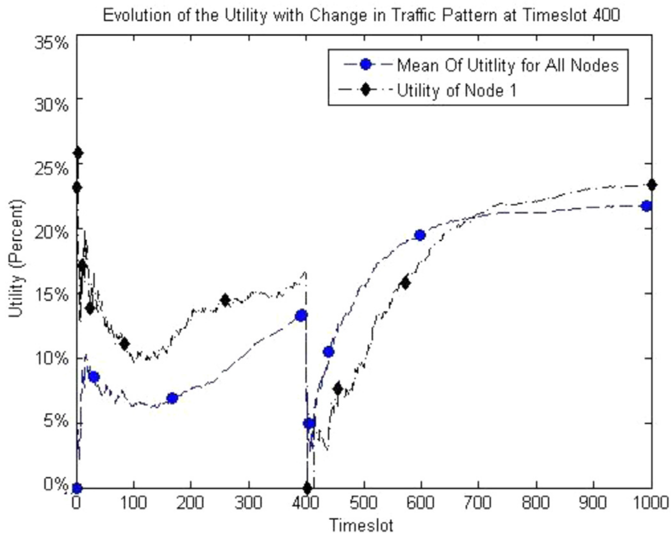

Fig. 8. Evolution of the utility with change in traffic pattern.

algorithm. So it can be inferenced that the average penalties for choosing the wrong action in the learning automata, gradually decreases in comparison with pure-chance automata. In other word, each node equipped with learning automata, decrease the chance of wrong channel selection and learn how to choose appropriate channels to its radio based on past experiences.

In the third scenario, we investigated adaptability of learning automata based channel assignment against sudden changes in traffic pattern. In this experiment, after the system reaches steady-state behavior in channel assignment, some new flows start data transmission with the new senders and receivers. As shown in Fig. 8, bulk traffic injected in some nodes leads to more interferences and causes a sharp decline in the mean utility function. Learning automata receive estimated interference from RSSI as an environmental response and change their channel assignment strategy according to the new situation. This mechanism of the learning-automata-based channel switching increases flexibility of adaptation, especially when the network dynamics vary in time.

Finally, the performance of the algorithm was investigated in a random topology such that the nodes were placed in a square region with the area of 5000×5000 meter-square. In this area, nine wireless nodes were spread randomly, as depicted in Fig. 9. Two separate flows—F1 from Nodes 1 to 8 and F2 from Nodes 2 to 7—were considered. Fig. 10 shows the

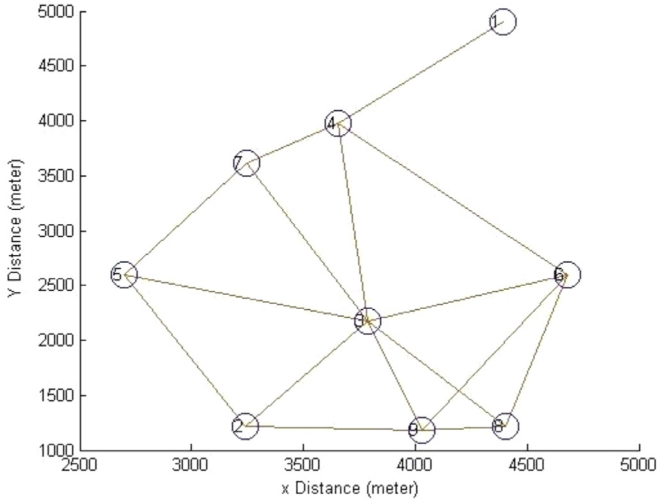

Fig. 9. Random topology for 9 nodes.

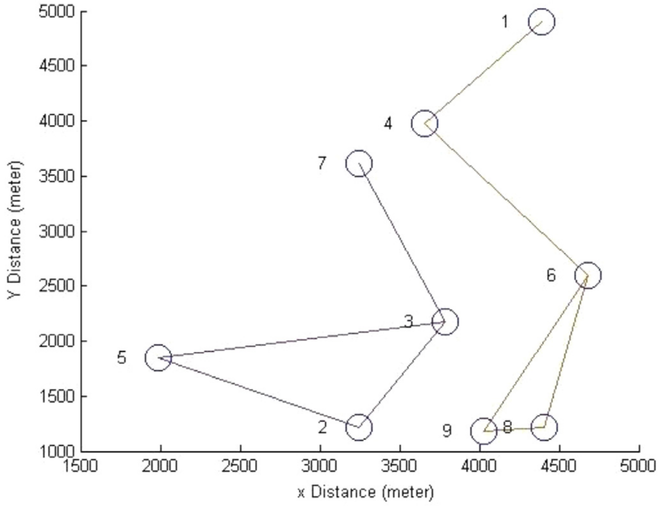

Fig. 10. Assigned channels for two flow in the random topology.

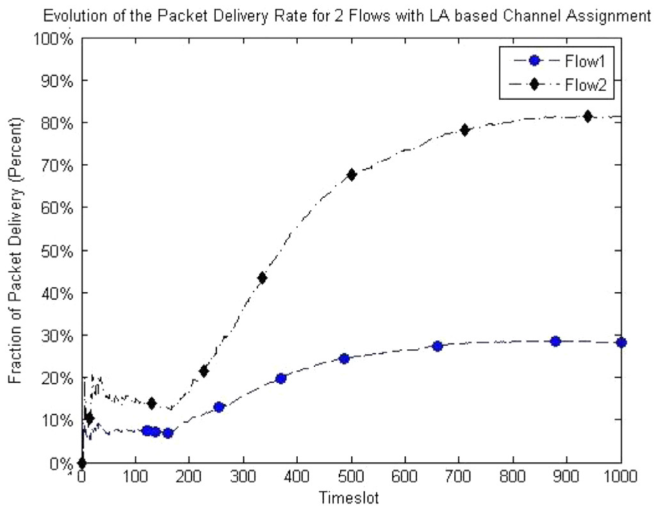

Fig. 11. Packet delivery rate for two flow in the random topology.

simulation results of the channel assignment algorithm, adaptation of selected channels according to their flow of traffic, and the position and distance of nodes from one to another. According to the algorithm, nodes gradually learn to choose different channels in terms of their traffic path from source to destination. Fig. 11 shows the packet delivery rate for two specified flows. As represented in the figure, flow F2 has a better delivery rate compared to F1 because of smaller hop-count than that of F1, such that a larger portion of the generated packets in F2 is delivered to the destination. Based on the proposed algorithm, the allocation of the channels based on learning automata acts in such a way that, although the connectivity of the network associated with each data flow is preserved, but by assigning appropriate channels to radio connections, minimizes interference to reach the better packet delivery rate.

Algorithm 1

6. Conclusions

The paper models wireless mesh network channel assignment problem as a sub-problem of the general wireless resource optimization problem and proposes a distributed learning-automata-based algorithm to perform channel allocation. Learning-automata-based algorithm has the advantage of topology-independent solution, which can be implemented in real-world physical interference model of wireless networks. The proposed scheme can be used in multi-channel, multi-radio wireless systems; the automata gradually learn the optimal strategy of channel selection to achieve maximum utility. Based on this model, the L_{R-P} algorithm is used in a multi-automata system to determine channel selection probabilities when the network dynamic characteristics are unknown. This algorithm-adjusted action probabilities according to environment response which is reflected in observed utility. The simulation results for different scenarios show that the channel allocation permits the network to attain an optimum level of achievable network throughput. Another important achievement of the algorithm is high adaptability when the network dynamics, especially traffic pattern change across time. This mechanism of the learning-automata-based channel switching increases flexibility of adaptation, especially when the traffic pattern randomly changes according to network users' demand. Also, these results show that the proposed scheme provides significant improvement in network behavior and can establish good balance for network connectivity and throughput.

References

[1] Jain K, Padhye J, Padmanabhan VN, Qiu L. Impact of interference on multi-hop wireless network performance. Wirel Netw 2005;11(4):471–87.
[2] Ishmael J, Bury S, Pezaros D, Race N. Deploying rural community wireless mesh networks. Internet Comput IEEE 2008;12(4):22–9.
[3] Beheshtifard Z, Meybodi MR. Online channel assignment in multi-radio wireless mesh networks using learning automata. Int J Soft Comput Softw Eng 2013;3(3):726–32.
[4] Beheshtifard Z, Meybodi MR. Maximal throughput scheduling based on the physical interference model using learning automata. Ad Hoc Netw 2016;45:65–79.
[5] Torkestani JA, Meybodi MR. Mobility-based multicast routing algorithm for wireless mobile Ad-hoc networks: a learning automata approach. Comput Commun 2010;33(6):721–35.
[6] Si W, Selvakennedy S, Zomaya AY. An overview of channel assignment methods for multi-radio multi-channel wireless mesh networks. J Parallel Distrib Comput 2010;70(5):505–24.
[7] Skalli H, Ghosh S, Das SK, Lenzini L, Conti M. Channel assignment strategies for multiradio wireless mesh networks: issues and solutions. Commun Mag IEEE 2007;45(11):86–95.
[8] Raniwala A, Gopalan K, Chiueh T-C. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. In: ACM SIGCOMM computer communication review, 8; 2004. p. 50–65.
[9] Marina MK, Das SR, Subramanian AP. A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks. Comput Netw 2010;54(2):241–56.
[10] Tang J, Xue G, Zhang W. Interference-aware topology control and QoS routing in multi-channel wireless mesh networks. In: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing. ACM; 2005.
[11] Bahl P, Chandra R, Dunagan J. SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In: Proceedings of the 10th annual international conference on Mobile computing and networking. ACM; 2004.
[12] Raniwala A, Chiueh TC. Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In: Proceedings of the 24th annual joint conference of the IEEE computer and communications societies, INFOCOM. IEEE; 2005.
[13] Avallone S, Banchs A. A channel assignment and routing algorithm for energy harvesting multiradio wireless mesh networks. IEEE J Sel Areas Commun 2016;34(5):1463–76.
[14] Zhao X, Zhang S, Li L, Qu Z, Zhang Y, Ding Y, Liu J. A multi-radio multi-channel assignment algorithm based on topology control and link interference weight for a power distribution wireless mesh network. Wirel Personal Commun 2018;99(1):555–66.
[15] Pediaditaki S, Arrieta P, Marina MK. A learning-based approach for distributed multi-radio channel allocation in wireless mesh networks. 17th IEEE international conference on network protocols, ICNP. IEEE; 2009.
[16] Selvakumar K, Revathy G. Escalating quality of services with channel assignment and traffic scheduling in wireless mesh networks. Cluster Comput 2018:1–7.
[17] Shajaiah H, Abdelhadi A, Clancy C. Utility functions and resource allocation for spectrum sharing. In: Resource allocation with carrier aggregation in cellular networks. Springer; 2018. p. 19–24.
[18] Zheng X, Cai Z, Li J, Gao H. A study on application-aware scheduling in wireless networks. IEEE Trans Mob Computing 2017;16(7):1787–801.
[19] Bertsimas D, Tsitsiklis JN. Introduction to linear optimization, 6. MA: Athena Scientific Belmont; 1997.
[20] Athanasopoulou E, Bui LX, Ji T, Srikant R, Stolyar A. Back-pressure-based packet-by-packet adaptive routing in communication networks. IEEE/ACM Trans Netw 2013;21(1):244–57.
[21] Gupta P, Kumar PR. In: The capacity of wireless networks, 46; 2000. p. 388–404.
[22] Foschini G, Salz J. Digital communications over fading radio channels. Bell Syst Tech J 1983;62(2):429–56.
[23] O'Neill D, Thian BS, Goldsmith A, Boyd S. Wireless NUM: Rate and reliability tradeoffs in random environment. In: Proceedings of the Wireless communications and networking conference. WCNC. IEEE; 2009.
[24] Reis C, Mahajan R, Rodrig M, Wetherall D, Zahorjan J. Measurement-based models of delivery and interference in static wireless networks. ACM SIGCOMM computer communication review. ACM; 2006.
[25] Narendra KS, Thathachar MA. Learning automata: an introduction. Courier Corporation; 2012.

Ziaeddin Beheshtifard received his B.S. and M.S. degrees in Computer Engineering from University of Tehran, Iran in 1996 and Tarbiat Modares University, Iran in 1999, respectively. He is currently pursuing his Ph.D. degree in Computer Engineering at Azad University of Qazvin Branch, Qazvin, Iran. His research interests include wireless networks, ad hoc networks, soft computing, learning systems and distributed algorithms.

Mohammad Reza Meybodi received his M.S. and Ph.D. degrees from Oklahoma University, USA during 1980 and 1983, respectively in Computer Science. Presently, he is a Full Professor in Computer Engineering Department, Amirkabir University of Technology, Iran. His research interests include wireless networks, fault tolerant systems, learning systems, parallel algorithms, soft computing and software development.