Wireless Pers Commun (2015) 82:61–80
DOI 10.1007/s11277-014-2194-0

Improving Channel Assignment in Multi-radio Wireless Mesh Networks with Learning Automata

Mohammad Shojafar · Saeid Abolfazli · Habib Mostafaei · Mukesh Singhal

Published online: 13 January 2015

Abstract Wireless mesh networks (WMNs) consist of static nodes that usually have one or more radios or media. Optimal channel assignment (CA) for nodes is a challenging problem in WMNs. CA aims to minimize interference in the overall network and thus increase the total capacity of the network. This paper proposes a new method for solving the CA problem that comparatively performs more efficient than existing methods. The link layer in the TCP/IP model is a descriptive realm of networking protocols that operates on the local network link in routers discovery and neighboring hosts. TCP/IP employs the link-layer protocol (LLP) that is included among the hybrid states in CA methods, and learning automata are used to complete the algorithm with an intelligent method for suitable CA. We call this algorithm LLLA, which are short for *LLP and learning automata*. Our simulation results show that LLLA performs more efficient than ad hoc on-demand distance vector (AODV) types with respect to parameters such as packet drop, end-to-end delay, average goodput, jitter in special applications, and energy usage.

Keywords Wireless mesh network (WMN) · Multi-radio · Channel assignment (CA) · Multi-channel · Learning automata (LA) · Energy usage reduction

M. Shojafar (✉)
Department of Information Engineering, Electronics and Telecommunications (DIET),
Sapienza University of Rome, Rome, Italy
e-mail: m.shojafar@yahoo.com; shojafar@diet.uniroma1.it
URL: http://www.mshojafar.com

S. Abolfazli
Department of Computer System and Technology, The University of Malaya, Kuala Lumpur, Malaysia
e-mail: abolfazli@ieee.org

H. Mostafaei
Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran
e-mail: h.mostafaei@iaurmia.ac.ir

M. Singhal
Computer Science and Engineering, University of California, Merced, CA S&E 296, USA
e-mail: msinghal@ucmerced.edu

1 Introduction

Recent developments in micro-electro-mechanical-systems (MEMSs), wireless telecommunication, and digital electronics have made it possible to manufacture small, low-energy-consuming [1], cost-effective devices that are capable of wireless connections [2]. Wireless networks include infrastructure-based wireless networks and infrastructureless wireless networks. Infrastructure-based wireless networks have a central controller and service providers called *access points* that have the same function as routers in wired networks, i.e., nodes connect to one another through access points. In contrast, infrastructureless wireless networks have no central controller or access points; rather, every node acts as a final node as well as a router for other nodes in the network. Infrastructureless wireless networks include mobile ad hoc networks (MANETs), wireless sensor networks (WSNs), and wireless mesh networks (WMNs). WMNs are systems with high-speed broadband connections to clients. They are completely wireless and self-organized and guide traffic to or from the Internet with multi-hop and ad hoc methods. In these networks, some nodes are stable and static and usually have one or more radios or network interfaces which are full duplex [3]. WMNs use available channels that are supported by 802.11 protocols to reach their maximum capacity. Every node in a WMN connects to its neighboring nodes or to nodes in its transmission range when both nodes have at least one radio that uses a common channel through which a connection can be made [4].

Since a channel is a resource, assigning an optimal channel to nodes is an important problem in WMNs. Given a network of mesh nodes where each node has one/more radio interfaces, CA algorithm assigns a channel to each connection link so that the number of channels given to a node is no greater than the number of its radios. The purpose of CA is maximizing the number of channels in use in the network through properly choosing channels to avoid interferences. Interference means that links in the network share common channels with other links; these links are connected to each other by an edge in an interference graph. By minimizing interference in the overall network, CA increases its total capacity. The CA problem is NP-complete [5].

To avoid interference, channels may be assigned in such a way that nodes have the least number of common channels. This means decreasing the connections of nodes. A large number of connections in the network and low interference are not possible simultaneously. So there is a trade-off between the number of connections in the network and network interference. Interference in WMNs is inevitable and is the factor that limits their capacity. A channel assignment (CA) algorithm in WMNs aims to maximize capacity and decrease interference [6,7].

The quality of end-to-end communication in WMNs depends on good CA, efficient use of radios and network interface cards at the nodes, and the routing protocol. Interference between channels at a node is a function of both the traffic patterns of neighbor nodes and the channels that have been assigned to them. Therefore, an ideal approach will consider CA in addition to the routing protocol [8,46].

In this paper, we propose a new method to improve CA and decrease interference in multi-channel WMNs. In our approach we try to reduce the complexity and provide high performance based on mentioned parameters. Our Objective is to propose a new method for solving the CA problem that performs more efficiently than existing methods. As a result, we are given a WMN with hybrid channel assignment, fixed mesh routers, transmission range and interference range of the network interface controllers or NICs, number of available channels, a traffic profile specifying the expected load on each link. The objective is to assign a channel to each of the links subject to the interface constrains such as minimum amount of network interferences, dynamics of traffic selection for the mesh routers and numbers of hops for assignment. Our preliminary version of this paper, which is published in [9], introduced the algorithm, without considering the models in detail. The current prolonged version, we highlight the work and try to compare the proposed work not only with AODV [10] but also with two related and state-of-the-art methods in six important QoSs in this types networks, which are further narrated.

We compare our algorithm with the ad hoc on-demand distance vector (AODV) method [10,11], multiple-radio AODV (AODV-MR) [11] and multi-linked AODV (AODV-ML) method [12] in terms of channel interference and simultaneous connections of network nodes, showing that our proposed method yields better results. Section 2 presents background material. Section 3 describes related research on CA and routing in WMNs. Section 4 presents our proposed method, which is based on learning automata (LAs). Section 5 includes tables and graphs related to our simulation and efficiency estimation, comparing our proposal with other methods. Finally, Sect. 6 presents conclusions and directions for future work.

2 Background

In this section, we provide background material.

2.1 Overview of Time Division Multiplexing and Frequency Division Multiplexing

One of the methods that provide simple routing and CA for WMNs is the ad hoc AODV method. In the standard AODV, CA is done by carrier sense multiple access/collision avoidance (CSMA/CA). CSMA/CA has several problems, such as data congestion and hidden stations.

First, the complexity problems increase when nodes have high mobility in the network, because it is difficult to maintain the stability of the data communication path between a source and destination. Establishing routes consumes network time and energy, and when there is high mobility, energy and time will be wasted if an established path is broken. In our proposal, nodes retain the path or route adaptively by changing the instantaneous structure of the network; this is one of the most important issues that we are able to resolve by suitable channel allocation.

Second, channel assignment with multiple radios is another issue for the standard AODV. As we know, when we increase hardware capabilities, we are able to allocate channels easily and more reliably. Not only it is complex to properly assign channels to nodes which have several radios, but also the cost is higher compared to simple WMNs with single-radio nodes.

For transmitting data in a channel, we need to provide data multiplexing. There are several methods for this: one is time division multiplexing (TDM) and the other is frequency division multiplexing (FDM). We use both of these methods in our work, as will be explained in the remainder of the paper. In TDM, a channel is assigned based on the requested time or duration for each medium. Independent signals are transmitted and received over a common signal path by means of synchronized switches at each end of the transmission line, so that each signal appears on the line only a fraction of the time in an alternating pattern. For example, when we have some customers or nodes waiting to use common channels, we provide a divided time switch, and each node transmits its data on the channel based on this switch. FDM is used when the communication link bandwidth is greater than the total transmitted signal bandwidths. This technique divides the total bandwidth available in a communication medium into a series of non-overlapping frequency sub-bands, each of which is used to carry a separate signal.

2.2 Overview of Learning Automata

AI algorithms are currently applied in WMNs to decrease the time needed for CA and to simulate real systems, modeling various features and monitoring their status to resolve any problems that may exist. One of these algorithms is the stochastic LA [13,14], which can be considered to be a single object that executes several actions. An LA's performance at any time involves selecting an action from a set of actions and evaluating the action in a random environment, using the response from the environment to select its next action. Hence, an LA gradually identifies optimal actions. The method that an LA uses to select the next action is deterministic. LAs can have fixed or variable structures. A variable LA is a set consisting of quadruples $\{\alpha,\ \beta,\ p,\ T\}$, where α is an action set, β is an input set, p is a probability for the action, and T is learning algorithm. LA algorithms can be classified as S and P models.

In a P model, or a standard model, the value of a reward (when a desirable response is received) or penalty (when a non-optimal or non-desirable response is received) is constant and is the same for each action. The general form of a standard-model LA is given by Eqs. (1) and (2). Equation (1) applies when a desirable response is received from the environment, while Eq. (2) applies when a non-desirable response is received. In Eq. (1), the probability of the action is increased by the reward rate while the probabilities of all other actions decrease. In Eq. (2), the probability of the action is decreased by the punishment rate while the probabilities of all other actions increase. The parameter r indicates that the system has at most r actions and a is the reward parameter and b is the punishment parameter.

$$\begin{aligned} p_i(n+1) &= p_i(n) + a[1 - p_i(n)] \\ p_j(n+1) &= p(1-a)p_j(n); \quad \forall j;\ j \neq i \end{aligned} \tag{1}$$

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

In an S model, the response of the environment, defined as a function of the automaton, is added to the standard model to improve the automaton's learning. There are three types of S models, based on the reward and punishment parameters a and b: when a = b the S model is called $S\text{-}L_{RP}$, when $b \ll a$ the S model is called $S\text{-}L_{ReP}$, and when $b = 0$ the S model is called $S\text{-}L_{RI}$ [15,16].

For $S\text{-}L_{RP}$ model learning automata, a linear algorithm is given in Eq. (3):

$$\begin{aligned} p_i(n+1) &= p_i(n) + a \cdot (1 - \beta_i(n)) \cdot (1 - p_i(n)), \\ p_j(n+1) &= p_j(n) - a \cdot (1 - \beta_i(n)) \cdot p_j(n), \quad \forall j,\ j \neq i. \end{aligned} \tag{3}$$

For $S\text{-}L_{RI}$ model learning automata, a linear algorithm is given in Eq. (4):

$$\begin{aligned} p_i(n+1) &= p_i(n) + a \cdot (1 - \beta_i(n)) \cdot (1 - p_i(n)) - a \cdot \beta_i(n) \cdot p_i(n), \\ p_j(n+1) &= p_j(n) - a \cdot (1 - \beta_i(n)) \cdot p_j(n) + a \cdot \beta_i(n) \cdot \left[\frac{1}{r-1} - p_j(n)\right] \\ &\quad - a \cdot (1 - \beta_i(n)) \cdot p_j(n), \quad \forall j,\ j \neq i. \end{aligned} \tag{4}$$

For $S\text{-}L_{ReP}$ model learning automata, a linear algorithm is given in Eq. (5):

$$\begin{aligned} p_i(n+1) &= p_i(n) + a \cdot (1 - \beta_i(n)) \cdot (1 - p_i(n)) - a \cdot \beta_i(n) \cdot p_i(n), \\ p_j(n+1) &= p_j(n) - a \cdot (1 - \beta_i(n)) \cdot p_j(n) + a\beta_i(n) \cdot \left[\frac{1}{r-1} - p_j(n)\right] \\ &\quad - a \cdot (1 - \beta_i(n)) \cdot p_j(n), \quad \forall j,\ j \neq i. \end{aligned} \tag{5}$$

Learning automata update the probability vectors for r actions. If action α_i is selected in the nth iteration and the environment's response is $\beta_i(n)$, then the automaton's probability vector is updated based on the different S Models (i.e., aforementioned) equations. More information about learning automata can be found in [15–17].

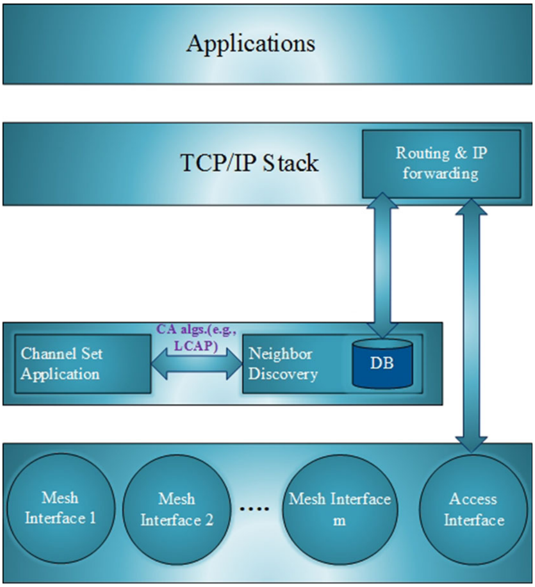
3 Related Work

This section reviews recent work in routing and CA protocols. Previous work on channel assignment schemes can be broadly classified into three categories: *static*, *dynamic*, and *hybrid* assignment. In *static channel assignment*, each interface of the mesh network is permanently assigned a channel. The goal of static channel allocation is to enhance the performance of the overall network. Research in this category includes [18–25]. Specifically, a method proposed by Das [18], based on the carrier sense multiple access (CSMA) algorithm with RTS/CTS/ACK in the MAC layers of WMNs, dedicates channels constantly and is therefore a constant CA method. The paper studies the problem of CA for WMNs with multi-channel multi-radios. It focuses on static WMNs in which multiple channels are available for each wireless medium without interference, with the purpose of finding a constant CA that has the most mutual links that can be activated simultaneously by considering the limitation of interferers. This paper proposes two Integral Linear Programming (ILP) models to solve the problem of constant CA with multiple radios. The numerical results show that it is very beneficial to increase the number of radios in each node and the number of channels in the network. The benefits of static channel assignment are that the channel switching overhead is low because of the fixed channel allocation. Therefore, it does not consider adapting to changing traffic and the allocation is centralized, where the main factors that are considered are routing and network connectivity, and the allocation is heuristically optimal [26]. In *dynamic channel assignment*, nodes must switch their interfaces dynamically from one channel to another for successive data transmissions. Several good methods have recently exploited this technique in their allocation of channels and routing in mesh networks, including [27–35]. In simple terms, when two interfaces need to communicate, they must switch to the same channel. Hence, some synchronization or control protocols are needed to enable interfaces to negotiate a common channel and simultaneously switch to the channel [26].

Pediaditaki [27] presented the first method, named LCAP (Learning-based Channel Allocation Protocol), for dynamic assignment. The algorithm is based on LAs and is similar to the Asymmetric and Distributed graph Coloring algorithm (ADC). Every node can do CA based on its neighbor's conditions, and neighboring nodes are able to automatically find their neighbors for information transmission in the channel by considering information about each channel. This method increases channel efficiency for high traffic. This method is similar to Ko [28]. The method in [28] uses the channel selection algorithm presented below to heuristically specify a channel. The aim is to decrease the work of interferers in different channels and to decrease surplus generation in a channel by adding one or more new nodes in data transmissions. The method not only tolerates various errors but also effectively adapts to changes in channel performance caused by external interferers. The adaptability is achieved by using an intelligent LA algorithm. A new architecture is presented in detail for WMNs and CA. The proposed algorithm is added to the network layer as an effective algorithm for CA. An example of this architecture is shown in Fig. 1 which is similar to other CA methods.

The authors in [28] considered several parameters to study their proposal. They used the difference between the maximum and minimum number of NICs involved in a channel to determine channel efficiency. A smaller difference means that nodes in the network approach

Fig. 1 CA architecture such as LCAP



homogeneity, which leads to greater channel efficiency [28]. In [27,28], the channel switching overhead is *medium* and adopts by changing the traffic is fair for most cases. These are distributed methods for CA between nodes that use environmental interfaces as a major factor in channel allocation and routing, and they are mathematically and heuristically optimal compared to other methods of dynamic allocation [26].

Finally, *hybrid channel assignment* applies a static or semi-dynamic CA method to a network's fixed interfaces and a dynamic assignment method to the switching interfaces. The goal is to merge the advantages of the two methods to model real systems more precisely. This idea was first proposed in [6,10]. At each node, some interfaces are assigned fixed channels, while the remaining interfaces can frequently switch channels. Each node informs its neighbors of its fixed channels by broadcasting a message on all the channels to its neighbors. If a node needs to transmit some packets to a neighbor, it will first switch one of its switchable interfaces to a channel which one of the neighbor's fixed interfaces is working on, and then send the packets through the switchable interface [20]. Researchers are currently working to enhance their models based on hybrid allocation. Further discussion of this topic can be found in [6,10–12,26,36–40].

In [11,12], the authors assesses the ad hoc performance of AODV in some radio WMNs. In [11], simulation results show that in high-load traffic, AODV-MR can use the extended spectrum in the best way and that this is much better than single-radio AODV. In [12], a AODV-ML method is presented. This is a variation of the AODV router protocol that is responsible for discovering and using multiple links between multiple-homed nodes in wireless networks. The proposed protocol can be considered to be a broad spectrum of multiple-hopped networks consisting of multiple-homed nodes. Both of these hybrid approaches reduce channel switching overhead compared to dynamic CA methods, and they can assign channels adaptively to changing traffic, unlike static channel allocation methods. However, each link in [11] is constructed between a static interface and a dynamic interface, so there is an inevitable delay in data transmission because of the channel switching, similar to [6]. In comparison, the method in [12] ensures low-delay paths with static links, which are constructed between static inter-

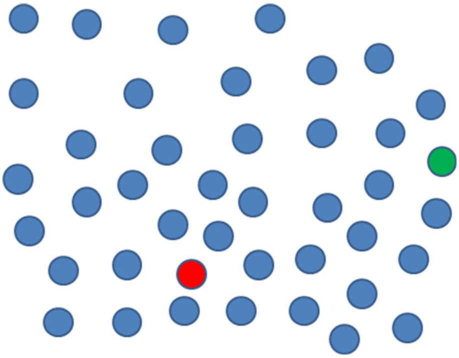


Fig. 2 Initial stage in LLLA

faces. Therefore, this method is preferable for use when there are both delay-sensitive and delay-non-sensitive data flows [26]. Our proposed method belongs to the hybrid methods, and we explain in the next section how it uses coordinated static and dynamic assignment for channels; it is distributed like the other hybrid methods, but with better performance results.

4 Our Proposed Method (LLLA)

In this section, the purpose is to present an algorithm that improves CA with learning automata (LAs). We use the AODV wireless protocol. For CA, we apply the link-layer protocols (LLPs) of hybrid states in a channel selection, and LAs are used to complete the algorithm with an intelligent method for proper CA. We therefore call our algorithm LLLA, which stands for a combination of LLP and LA.

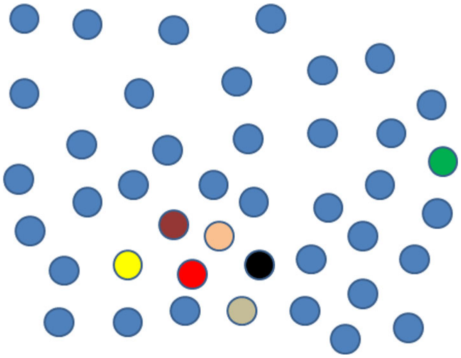
The purpose of our intelligent algorithm is to find the best states that satisfy several criteria. In Fig. 2, it is assumed that each node has at least three radio interfaces and can transmit information by simultaneous time-division multiplexing (TDM) and frequency-division multiplexing (FDM). We assume that the radio interfaces are the same, so that they can all exchange information in the frequency ranges. This assumption is reasonable, because many medium cards made by different companies use the same standards.

Our goal is to create an intelligent method using an LLP method, LAs, and TDM and FDM multiplexing. Each node has two functions [e.g., according the LA probability types, Eqs. (3–5)]. The intelligence of this method is derived from the LA, which can transform radio interface from static to dynamic states and conversely, according to environment feedback. We want to develop an algorithm that can adapt itself to network requirements, network topology, and applications in the network. At first, the medium state is chosen according to the volume of data. Then, based on the feedback received from the network, the medium changes its state.

The goal in Fig. 2 is to transmit information between the red node and the green node. The LLLA algorithm is implemented on all nodes. CA is based on our proposed LLLA algorithm, in which one third of the nodes are always considered static and the remainders are initially considered dynamic. It is assumed that the red node has three radio interfaces, so one of its radios is considered to be a static medium and the other two radios are considered to be dynamic radio interface.

In the implementation, static interface use TDM for multiplexing and dynamic radio interfaces use FDM. On the outset, the transmitter node (i.e., the node who wants to transmit

Fig. 3 LLLA's transmission stage



information) goes through the initial stage of the AODV protocol. This node transmits a "Hello" packet to its neighbors. Neighbors who receive the "Hello" packet send their response. The next stage implements the second phase of the AODV protocol (i.e., when a node wishes to send a packet to some destination, it checks its routing table to determine if it has a current route to the destination). If Yes, it forwards the packet to next hop node. If No, it initiates a route discovery process. Route discovery process begins with the creation of a route Request (REQUEST) packet; therefore, source node creates it. Some REQUEST packets are sent, and neighbors who receive the message send a list of their neighbors as a response (echo) to the requested node. It is assumed that the red node has simultaneous data transmission with five of its neighbors (see Fig. 3).

As mentioned earlier, it is assumed that the red node has three radio interfaces, one of them in a static state and the other two in dynamic states. The red node should therefore have five connections with its three radio interfaces. All the nodes in the transmission stage should send the size of their packets to their neighbors. In addition, nodes that are transmission radio interfaces should receive the packet sizes from the transmitters and send them to the next nodes. In this way, all nodes will be aware of the sizes of the packets that are to be transmitted.

Now suppose that the gray, black, and yellow nodes have small packets and the orange and brown nodes have large packets. Each LA in our algorithm is a standard LA [41]. Also, since desired and undesired states of criteria are considered, our environment is considered to be a P environment (*P-Type* or a P environment characterizes one of the types of LA) and S-types (i.e., S_{ReP}, S_{RI}, S_{RP}). In such an environment, feedback from the environment determines whether the environment is *desired* or *not desired* (*undesired*). For example, if minimal end-to-end delay times, packet drops, and so on are *desired*, this means that this is the desired performance. However, if the performance is not desired according to the criteria, the environment is considered to be *undesired*. In the LLLA algorithm, the red node has five functions, as it makes a connection channel for each of the neighboring nodes. In the initial state, large packets use static radio interface and small packets use dynamic radio interface (due to the fact that, large or heavy packets need fixed or static radio to transfer the quota of their data continuously, but the dynamic one can transfer the data while the range will change in transmission easily. In the next stage, the probability for the connection type selection is determined by receiving information about the routing performance. For example, the brown node first sends a packet to its destination. This node uses a static medium to make a connection with the red node. If the red node receives information that there was an acceptable time between transmission of the brown node's packet and its acceptance it send it to the next hop align to reach to the destination. The probability of a connection with static radio interface for the brown node is increased and this connection is maintained for some time. Nevertheless, if the red node sees that the elapsed time is greater than past times for similar transmissions, it increases the probability of selecting dynamic radio interface for the brown node. In general, if a medium is first considered static for a node and the subsequent performance is undesired, its medium is then considered dynamic with a high probability, and vice versa. In this way, node radio interfaces types are adapted to network conditions.

Two main states that can occur in the algorithm are considered in the following. A very important issue is the interference area among radio interface. For example, consider neighboring nodes such that one has three radio interfaces, one has four, and one has two. At first glance, if each of these nodes is in the others' range or Channel, interference is inevitable. This problem can be solved by noting several points. The first is that before making any connection, the frequency range of a connecting node should be made known to its neighbors. Neighbors receiving the frequency range of the node can consider themselves to have wider frequency ranges. This continues until we are beyond the frequency range of a node or set of nodes. Then, when the next connections are requested in other parts of the network, the narrowest frequency range is again used for the connection. For example, if the red node considers its frequency range to be X to X + 300, three scenarios are possible for the brown node. The *first* is that all of the brown node's connections are connections with the red node, so it uses the same frequency range. *Second*, some of the brown node's connections may be with the red node and others with other nodes. In this case, the brown node uses the same frequency range for its connections with the red node but makes its other connections with frequencies greater than X + 300. In the *third* scenario, the brown node has no connections with the red node, and so, from the beginning, it selects other connections with frequencies greater than X + 300. The point is that all neighbors must save used frequencies in their memories in order to make correct decisions in this regard. The following state is also possible. A neighbor wants to make a connection and wants to communicate this. In this state, the highest frequency range is used for transmitting information so that no interference with existing connections is created. Also, for TDM, it should be noted that when a frequency range is determined for some connections, the frequency switches among the connections every 300 µs. Of course, radio interfaces will always search for the highest frequency range and can find it through a search cycle.

As previously noted, dynamic interfaces, use FDM and static interfaces use TDM for multiplexing. Now consider a state in which all interfaces are dynamic. This usually occurs when small packets are exchanged in the network and feedback concerning criteria such as drop, delay, etc., is therefore not desired. It is worth noting that if there is a neighbor whose transmitted packet size is large and all interfaces have switched to the dynamic state, then a request is sent for transforming a dynamic medium to a static medium. After studying this request, the neighbor node initially transmits large packets through dynamic radio interface. Then it gradually disconnects its radio interface from other nodes and finally transforms one of these radio interfaces to a static medium. It is important to note that if the reaction to this change in packet transmission is not proper, the static medium access is again transformed to a static medium access. A state in which all radio interfaces have switched to a static state because of network conditions can occur when most network packages are large it is easy to calculate and find the proper assignments for the channels. Of course, if small packets are in the network for transmission, there is no need for dynamic radio interfaces because they create a great surplus for the network.

In the proposed algorithm, first, each vertex of the graph $\langle V, E\rangle$ [42] (V or v_i; vertex is mesh nodes, Edge, or E is the considered probability for selecting next hop for each node) is equipped with a learning automaton, and as a result, a network learning automata isomorphic to the graph is initially constructed. The set of learning automata is defined by tuple $\langle$ A,$a\rangle$, where A $\equiv \{\mathrm{A}_1, \mathrm{A}_2, \ldots, \mathrm{A}_\mathrm{n}\}$ is the set of learning automata corresponding to the set of vertices, and $a \equiv \{a_1, a_2, \ldots, a_n\}$ denotes the set of actions in which $a_i \equiv \{a_i^1, a_i^2, \ldots, a_i^{r_i}\}$ describes the set of actions that can be taken by learning automaton A_i. In the proposed algorithm, the action set of each learning automata (i.e., A_i) is choosing one of the neighbors which are initialized equally as Eq. (6):

$$p_{ij} = \frac{1}{d(v_i)} \tag{6}$$

where $\mathrm{d}(\mathrm{v}_\mathrm{i})$ is the degree of vertex (node which includes mesh clients or mesh routers) v_i.

At the beginning of the algorithm, all learning automata are considered inactive. Randomly, a vertex is selected and activated (e.g., A_i), the activated automaton selects one of its neighbors (A_j) according to the probability vector A_i. Then, p_ij updates based on types of LAs and also the traffic rate, radio numbers which are taken into account and find the proper nodes for routing, scheduling and assigning the appropriate channel for the node.

Algorithm 1 briefly describes the LLLA approach pseudocode.

Algorithm 1. Linked-layer learning automata protocol for Channel Assignment in multi-layered WMN

```
Input: Graph G=<V,E>, maximum iterations T, convergence threshold tt, maximum available traffic rate MTR,
Output: The set of sample vertices (S) which are find the assigned channel with proper rout
Begin
    Disable all learning automata for nodes
While stopping criteria are not meet (T or tt or MTR) Do
Begin
    Enable $A_i$
    Choose an action by $A_i$ according to the eqs (for each LA calculates (1) to (5))
    If  (Ai visited before)
        The chosen action by the learning automata is rewarded, other actions punished
    EndIf
    $i \leftarrow v_i$
    Disable $A_i$
    Sort vertices
    $S \leftarrow k - top(verticeslist)$
End While
End Algorithm
```

5 Simulation Results

AODV code in the OPNET 14 [43] software package was used to implement the algorithm. The AODV algorithm is the default for OPNET. In the LLLA algorithm implementation, the LLP concept was entered in the second layer and the LA algorithm was implemented on the nodes. Also, OPNET 14 code was used for the FDM and TDM [43]. All our experiments are conducted on a desktop with Intel core 2 dual 2.6 GHz COPU and 4.0 GB RAM size.

5.1 Simulation Scenarios

In order to study the accuracy and quality of our proposed LLLA algorithm, several scenarios were considered and studied separately, based on different criteria. In all states of the LA, the reward rate was 0.1 and the punishment rate was 0.05. In the following subsections, we will describe the QoS parameters that were computed for comparing our proposed LLLA method with AODV, AODV-MR, and AODV-ML. We made a few assumptions in the scenario: All radios are statically tuned to a channel. All Mesh Clients and Mesh Routers have a radio tuned to a common 802.11b channel. The remaining radios on the Mesh Routers are tuned to orthogonal 802.11b channels. All antennae are Omni-directional. We summarize simulation parameters (network metrics, channels parameters) in Table 1.

5.2 Network Environment

Nodes in the network follow the Random waypoint mobility model, with some nodes transmitting large packets and other nodes transmitting small packets. To simulate this scenario, a 1,000 × 1,000 (m^2) environment with 70 nodes was used, and the nodes were distributed randomly in the network environment. The simulation running time was 25 min, and the purpose was to transmit data from ten nodes to their considered destinations with the AODV algorithm (i.e., numbers of mesh routers: 10, and numbers of mesh clients: 60). Each of the routers nodes began random transmission of information starting at the 30th second. Of the ten nodes, three nodes created high-volume traffic (512 byte) constant bit rate (CBR) traffic and seven created light traffic (128 byte) CBR traffic. This scenario was run four times, with results obtained for different states of the LAs (i.e., we test the system in P and S model (all there types) of LA). The IEEE 802.11 Distributed Coordination Function (DCF) [37] is used at the MAC layer. All packets are transmitted using the un-slotted Carrier Sense Multiple Access protocol with Collision Avoidance (CSMA/CA). In CSMA/CA each broadcasting node waits for a vacant channel by sensing the medium. If the channel is vacant, it makes the transmission. In case of a collision, the colliding stations wait using the Ethernet binary exponential back off algorithm [44]. Virtual Carrier Sensing (RTS/CTS) is disabled during the simulations.

5.2.1 End-to-End Delay (Average Latency)

As can be seen in Fig. 4, the proposed LLLA method results in a 55 % reduction in the end-to-end delay compared to the standard AODV method and approximately 5–6 % compared to ADOV-MR By increasing the speed of the clients in meter per seconds denoted m/s. the values for all protocols decrease after a while because system understand how to find the proper channel for routing and assignment based on LA. The reason is twofold: first, adding

Table 1 Simulation parameters

Type	Value
Examined protocols	LLLA, AODV, AODV-MR, and AODV-ML
Simulation time	1,500 s
Channel numbers	1, 3
Simulation area	1,000 m × 1,000 m
Propagation model	Two-ray ground reflection
Mobility model for mesh clients	Random waypoint
Max. speed of mesh clients	20 m/s
Transmission range	250 m
Traffic type	CBR (UDP)
Traffic size (payload size) per second	128, 512 byte
Packet rate	10 pkt/s
Initial energy	1,000 mJ

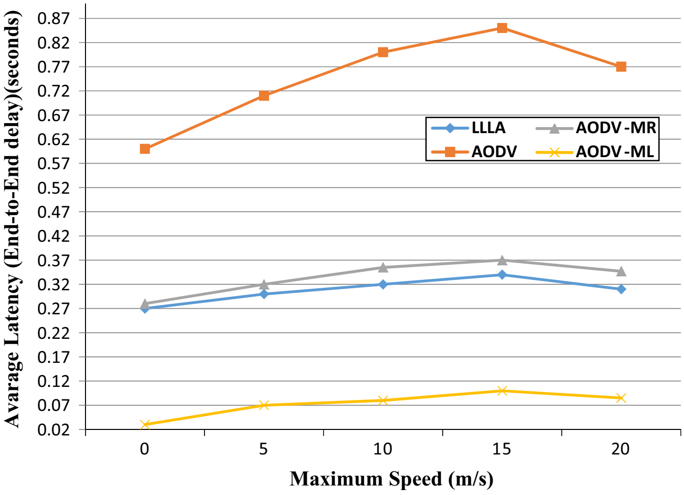


Fig. 4 End-to-end delay time (s) versus maximum speed (m/s)

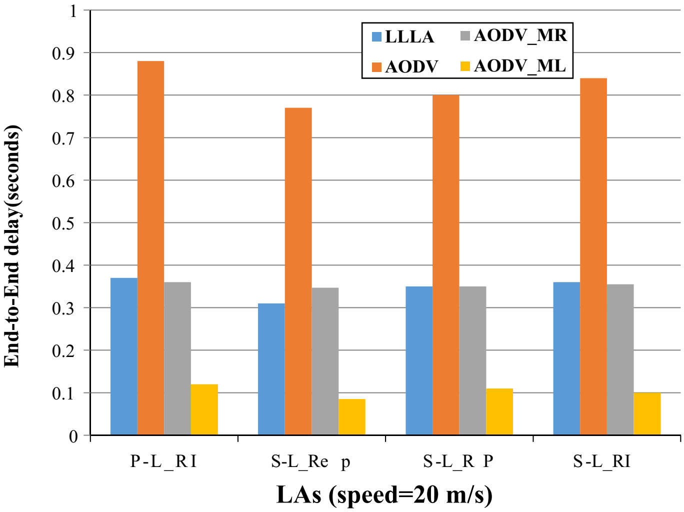


Fig. 5 End-to-end delay time (s) versus LAs (i.e., we consider alpha equal 0.1 and beta equal 0.05)

medium radios greatly increases the number of connections and enables a large volume of data to be transmitted. Second, the network is adapted to the data, so that nodes correctly use the radio interface according to the volume of data. Since there is almost no interference between nodes, every packet is delivered quickly, without any drops, and this greatly decreases the delay time. LLLA values are approximately 10–12 % less than AODV-ML.

Figure 5 shows end-to-end delay based on seconds for different Learning automata in the final state of clients' speed (speed equal to 20 m/s) for all the protocols. In the whole scenario, alpha (a); the reward rate is 0.1 and beta (β); the punishment rate is 0.05. It shows that the S-L$_{\text{ReP}}$ model of LA has better results compared to others. In the rest of simulations, we use this type of LA with its parameters.

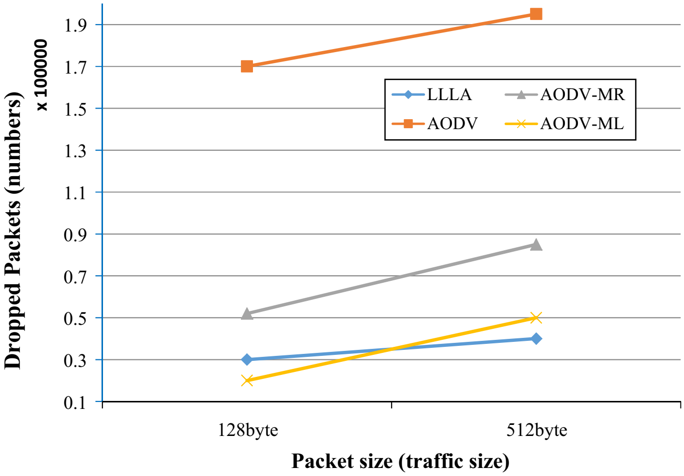


Fig. 6 Lost packets based versus packet sizes (bytes) (dynamic traffic)

5.2.2 Packet Loss (Dropped Packets)

The measure of dropped packets depends on each node's buffer memory. If the buffer memory is smaller than is necessary for packet buffering, nodes will drop a received packet and request the packet at another time. Of course, this increases network traffic. On the one hand, control messages will flow in the network, and on the other, the same packet will also flow through the network again, and this increases network congestion. Generally, full buffers occur when there is mobility in the route and destination node, or medium nodes have moved or there is a failure, so that a node cannot direct its data forward but its buffer is full and it must drop incoming packets. Moreover, the packet loss rate also influences the aggregate goodput of the network. It can be seen in Fig. 6 that, in our simulated scenario, standard AODV is clearly not able to handle the offered load and shows degraded performance with increased data packets and LLLA dropped packets are between AODV-MR and AODV-ML for the first group of traffic. When traffic increases in the channels, LLLA is able to adopt itself with the new situations and the dropped packets are increased with lower slop compared to AODV families and it reaches to the values less than AODV-ML and AODV-MR. One of the reasons is that when there is less interference, there are less packet failures and these less packets can see less congestion, resulting in smaller probability that the buffer at a node is full. Usually one of the reasons for having nodes buffer packets is the uncertainty of receiving packets without failure, because if medium nodes do not do any buffering, routing should be done from the source to the destination even for a small packet. However, AODV-MR and AODV-ML cope better with high loads and achieve significantly higher goodput compared to standard AODV. An approximately 300 % decrease in interference is obtained, and accordingly, the number of lost packets is decreased.

5.2.3 Network Throughput

Network throughput is the most important criterion for evaluating a network protocol. This criterion measures the traffic that can be handled correctly in a time period. When network

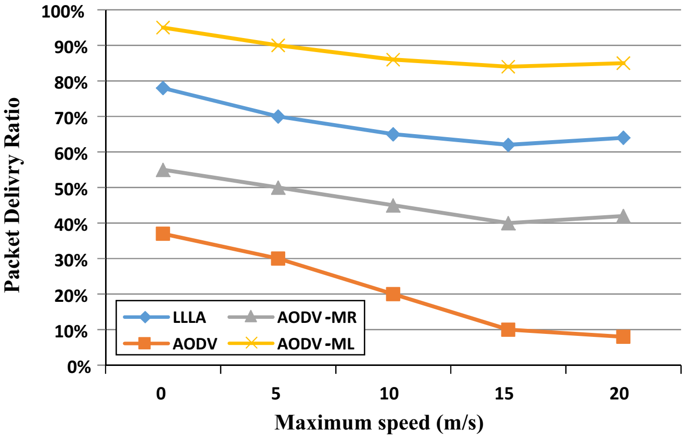


Fig. 7 Network throughput based versus client maximum speed (m/s)

delay is decreased, more data can be handled in a given time period, and more data traffic can be exchanged in the same time period. Simulation results show that, the LLLA algorithm provides greater network throughput than the standard AODV algorithm (i.e., an approximately 45 % improvement was obtained) but less than AODV-ML (i.e., an approximately 15 % less than AODV-ML). There are three reasons for LLLA's improvements. First, there is just one radio medium (interface) in the standard AODV algorithm, while the proposed LLLA algorithm uses a number of radio interface, and this increases the performance. Second, the LLLA algorithm considers states in which traffic is different, but the standard AODV algorithm does not make such distinction (i.e., changing the traffic takes into account in LLLA) (LLP statement). Third, the network adapts itself according to the type of traffic and traffic handling through the usage of LAs, and this has a significant effect on data transmission. In the simulation, it was shown that an LRI LA has the best performance, shown in Fig. 7, because of the constant state with respect to mobility and failures.

5.2.4 Jitter in Specific Applications

Jitter randomly modifying timing of packet transmissions to avoid simultaneous packet transmission by neighboring routers—something that might result packet losses due to link-layer collisions in routing/CA methods such as AODV families and LLLA. Jitter was also considered as a criterion for comparing LLLA with the standard AODV algorithm. Jitter is usually used as an evaluation criterion for specific applications with high volume, and one type of application in which jitter seems especially important is multimedia applications. Buffers play a key role in multimedia applications. For example, delay or failure in TV programs is responsible for more user dissatisfaction than the poor quality of the programs. Our simulation shows that the proposed LLLA algorithm produces an approximately 75 % improvement with respect to jitter after 15 min of running the program while AODV-ML decrease its jitter only 25 %. The reason is the fast transmission and non-filling of buffers. There is always a direct relation between jitter and end-to-end delay. As a result, a mechanism that is able to decrease end-to-end delay will also decrease jitter. The worst one is AODV, because it has just one radio interface and it is unable to fix its jitter while running. The simulation results for the LLLA algorithm are shown in Fig. 8.

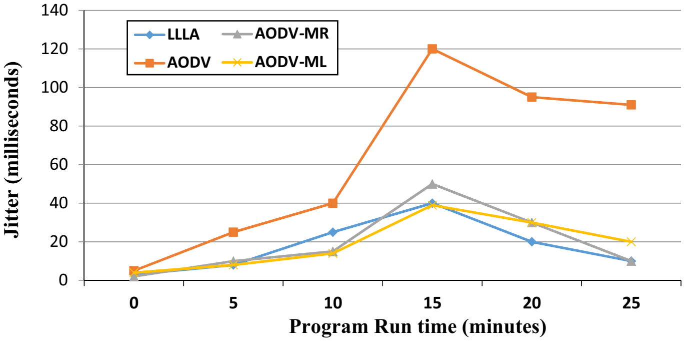


Fig. 8 Jitter (ms) based on run time (min)

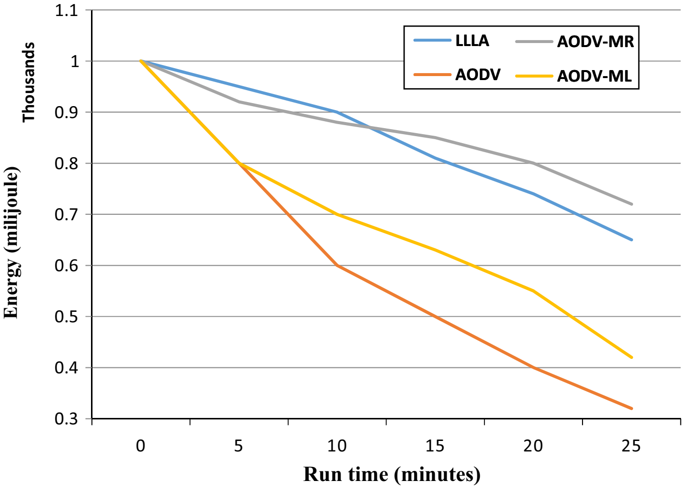


Fig. 9 Energy consumption based on time

5.2.5 Energy Consumption

As is clear from the definition of Algebra wireless routing protocol (AWNs [45]), their energy source is usually batteries. Since Battery life is limited, savings in energy is one of the important criteria in these networks. Each node consumes energy in four states: receiving data, transmitting data, audio, and awareness. The more data packets received and sent, the greater the energy consumption. If a node in a network sends data but the data do not reach their destinations, the energy used for this transmission is wasted. The node then tries to resend the data, and this causes further energy discharge. Consequently, for saving energy, an important issue is creating more stable networks (i.e., provide the network flexible and available for a long time without changing and active to work). Figure 9 compares the discharge of battery at nodes in the LLLA and standard AODV algorithms. As can be observed, LLLA had an approximately 35 % decrement in energy consumption in the simulation period compared to the standard AODV algorithm which is 66 % decrement, AODV-ML (25 %) is worse than AODV-MR (55 % reduces). Due to the fact that, multi-linked uses energy much more compare

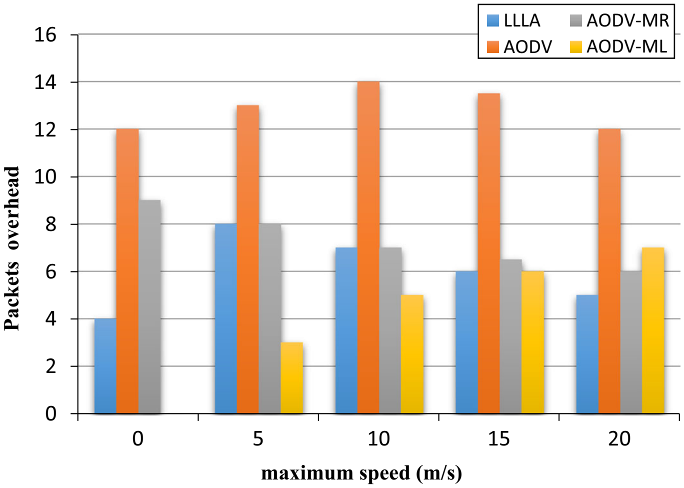


Fig. 10 Overhead rate (numbers) based on speed (m/s)

to multi-radio protocols, so, the energy residual for AODV-MR is more than AODV-ML and approximately more than 40 % after finishing the implementation.

5.2.6 Overhead of Produced Packets

Another important criterion for networks is production overhead. Production overhead is usually the result of greater control of the network, network debugging, and the occurrence of errors. In a standard AODV network, overhead related to CA is more related to interference problems. When interference occurs in the network, control packets are exchanged, and data packets are also reproduced, both of which add to network overhead. Since the proposed algorithm has a number of medium radios, it produces specific overheads. However, because of the logical mechanism that is created in the network, most of the overheads due to interference are resolved. In Fig. 10, the simulation results show that the LLLA algorithm reduces the network overhead approximately by 55 % compared to the standard AODV algorithm, by 40 % to AODV-MR and 25 % to AODV-ML. This overhead factor is almost the same for all types of LAs (Fig. 10).

6 Conclusion and Future Works

This paper applies learning automata for channel assignment in multi-radio wireless mesh networks (WMNs). States in the link-layer protocols are considered for channel selection by nodes. The authors conducted simulations to compare the proposed approach with AODV; a classical routing algorithm, AODV-MR, and AODV-ML for MANET that does exploit multiple channels. This paper studied the criteria of end-to-end delay, packet drop rate, network throughput, jitter, energy consumption, and packet-produced overhead for WMNs. The simulation results show that network-based implemented CA algorithms are able to adapt themselves to their underlying environment based on their functionalities.

In future research, we will consider a state in which nodes have movement to improve the proposed algorithm. Network nodes are considered mostly dynamic, and some nodes transmit large packets in the network while others transmit light packets. The LLLA method has a more intelligent performance than AODV but also has its own overhead and creates a time complexity proportional to itself. In addition, failure states should be considered and networks with special applications should be studied.

References

1. Cordeschi, N., Shojafar, M., & Baccarelli, E. (2013). Energy-saving self-configuring networked data centers. *Computer Networks*, *57*(17), 3479–3491.
2. Xu, S., & Saadawi, T. (2001). Does the IEEE 802.11MAC protocol work well in multi hop wireless adhoc networks. *IEEE Communications Magazine*, *39*, 130–137.
3. Ahmadi, A., Shojafar, M., Hajeforosh, S. F., Dehghan, M., & Singhal, M. (2014). An efficient routing algorithm to preserve k-coverage in wireless sensor networks. *The Journal of Supercomputing*, *68*(2), 599–623. doi:10.1007/s11227-013-1054-0.
4. Wanli, D., Kun, B., & Lei, Z. (2008). Distributed channel assignment algorithm for multi-channel wireless mesh networks. *International Colloquium on Computing, Communication, Control, and Management (CCCM)*, *2*, 444–448.
5. Sridhar, S., Guo, J., & Jha, S. (2009). Channel assignment in multi-radio wireless mesh networks: A graph-theoretic approach. In *First international conference on communication systems and networks (COMSNETS)*, Bangalore, India.
6. Kyasanur, P., & Vaidya, N. (2005). Routing and interface assignment in multi-channel multi-interface wireless networks. In *Proceedings IEEE conference on wireless communications and networks conference* (pp. 2051–56).
7. Gao, L., & Wang, X. (2008). A game approach for multi-channel allocation in multi-hop wireless networks. In *Proceedings of ACM MobiHoc* (pp. 303–312). doi:10.1145/1374618.1374659.
8. Pal, A., & Nasipuri, A. (2011). JRCA: A joint routing and channel assignment scheme for wireless mesh networks. In *IEEE IPCCC* (pp. 1–8). doi:10.1109/PCCC.2011.6108059.
9. Shojafar, M., Pooranian, Z., Shojafar, M., & Abraham, A. (2014). LLLA: New efficient channel assignment method in wireless mesh networks. In A. Abraham, P. Krömer and V. Snášel (Ed.), *Innovations in bio-inspired computing and applications* (Vol. 237, pp. 143–152). Berlin: Springer.
10. Kyasanur, P., & Vaidya, N. (2006). Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks. *Mobile Computing and Communications Review*, *10*(1), 31–43.
11. Pirzada, A. A., Portmann, M., & Indulska, J. (2006). Evaluation of multi-radio extensions to AODV for wireless mesh networks. In *Proceedings of the 4th ACM international workshop on mobility management and wireless access* (pp. 45–51). doi:10.1145/1164783.1164791.
12. Pirzada, A. A., Wishart, R., & Portmann, M. (2007). Multi-linked AODV routing protocol for wireless mesh networks. In *IEEE global telecommunications conference* (pp. 4925–4930). doi:10.1109/GLOCOM.2007.934.
13. Boppana, R., & Halldorsson, M. M. (1992). Approximating maximum independent sets by excluding subgraphs. *BIT Magazine*, *32*(2), 180–196. ISSN:0006-3835. doi:10.1007/BF01994876.
14. Bui, T. N., & Eppley, P. H. (1995). A hybrid genetic algorithm for the maximum clique problem. In *Proceedings of 6th international conference on genetic algorithms* (pp. 478–484).
15. Narendra, K. S., & Thathachar, M. A. L. (1989). *Learning automata: An introduction*. Englewood Cliffs, NJ: Prentice-Hall.
16. Poznyak, A. S., & Najim, K. (1997). *Learning automata and stochastic optimization*. Berlin: Springer. ISBN:3-540-76154-3.
17. Sayyad, A., Shojafar, M., Delkhah, Z., & Ahamadi, A. (2011). Region directed diffusion in sensor network using learning automata: RDDLA. *Journal of Advances in Computer Research, 2*(1), 71–83.
18. Das, A. K., Alazemi, H. M. K., Vijayakumar, R., & Roy, S. (2005). Optimization models for fixed channel assignment in wireless mesh networks with multiple radios. In *SECON* (pp. 463–474).
19. Subramaniam, A. P., Gupta, H., & Das, S. R. (2008). Minimum-interference channel assignment in multi-radio wireless mesh networks. *IEEE Transactions on Mobile Computing*, *7*(12), 1459–1473.
20. Raniwala, A., Gopalan, K., & Chiueh, T. (2004). Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, *8*(2), 50–65.
21. Alicherry, M., Bhatia, R., & Li, L. (2006). Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks. *IEEE Journal on Selected Areas in Communications*, *24*(11), 1960–1971.

22. Tang, J., Xue, G., & Zhang, W., (2005). Interference-aware topology control and qos routing in multi-channel wireless mesh networks. In *MobiHoc* (pp. 68–77). doi:10.1145/1062689.1062700.
23. Ding, Y., Huang, Y., Zeng, G., & Xiao, L. (2008). Channel assignment with partially overlapping channels in wireless mesh networks. In *WICON* (pp. 1–9).
24. Yang, J., Jiang, Q., Manivannan, D., & Singhal, M. (2005). A fault-tolerant distributed channel allocation scheme for cellular networks. *IEEE Transaction on Computers*, *54*(5), 616–629.
25. Cao, G., & Singhal, M. (2000). Distributed fault-tolerant channel allocation for cellular networks. *IEEE Journal on Selected Areas in Communications*, *18*(9), 1326–1337.
26. Ding, Y., & Xiao, L. (2011). Channel allocation in multi-channel wireless mesh networks. *Computer Communications*, *34*(7), 803–815.
27. Pediaditaki, S., Arrieta, Ph., & Marina, M. K. (2009). A learning-based approach for distributed multi-radio channel allocation in wireless mesh networks. In *ICNP* (pp. 31–41). doi:10.1109/ICNP.2009.5339701.
28. Ko, B., Misra, V., Padhye, J., & Rubenstein, D. (2007). Distributed channel assignment in multi-radio 802.11 mesh networks. In *Proceedings of IEEE WCNC* (pp. 3978–3983). doi:10.1109/WCNC.2007.727.
29. Raniwala, A., & Chiueh, T. (2005). Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network. In *IEEE INFOCOM* (Vol. 3, pp. 2223–2234). doi:10.1109/INFCOM.2005.1498497.
30. Aryafar, E., Gurewitz, O., & Knightly, E. W. (2008). Distance-1 constrained channel assignment in single radio wireless mesh networks. In *IEEE INFOCOM*. doi:10.1109/INFOCOM.2008.127.
31. Ramachandran, K. N., Belding, E. M., Almeroth K. C., & Buddhikot, M. M. (2006). Interference-aware channel assignment in multi-radio wireless mesh networks. In *IEEE INFOCOM* (pp. 1–12). doi:10.1109/INFOCOM.2006.177.
32. Wu, S.-L., Lin, C.-Y. Tseng, Y.-C., & Sheu, J.-P. (2000). A new multi-channel mac protocol with on-demand channel assignment for multi-hop mobile ad hoc networks. In *I-SPAN* (pp. 232–237). doi:10.1109/ISPAN.2000.900290.
33. So, J., & Vaidya, N., (2004). Multi-channel mac for ad hoc networks: Handling multichannel hidden terminals using a single transceiver. In *MobiHoc* (pp. 222–233). doi:10.1145/989459.989487.
34. Bahl, P., Chandra, R., & Dunagan, J. (2004). Ssch: Slotted seeded channel hopping for capacity improvement ieee 802.11 ad-hoc wireless networks. In *MobiCom* (pp. 216–230). doi:10.1145/1023720.1023742.
35. Dhananjay, A., Zhang, H., Li, J., & Subramanian, L. (2009). Practical distributed channel assignment and routing in dual-radio mesh networks. *ACM SIGCOMM Computer Communication Review*, *39*(4), 99–110.
36. Ding, Y., Pongaliur, K., & Xiao, L. (2009). Hybrid multi-channel multi-radio wireless mesh networks. In *Workshop IWQoS* (pp. 1–5). doi:10.1109/IWQoS.2009.5201403.
37. Jahanshahi, M., Dehghan, M., & Meybodi, M. R. (2011). A mathematical formulation for joint channel assignment and multicast routing in multi-channel multi-radio wireless mesh networks. *Journal of Network and Computer Applications*, *34*(6), 1869–1882.
38. Chianga, C., Chenb, H., Liaoc, W., & Shiha, K. (2014). A decentralized minislot scheduling protocol (DMSP) in TDMA-based wireless mesh networks. *Journal of Network and Computer Applications*, 206–215. doi:10.1016/j.jnca.2013.02.017.
39. Chen, T., Bansal, A., & Zhong, S. (2011). A reputation system for wireless mesh networks using network coding. *Journal of Network and Computer Applications*, *34*(2), 535–541. doi:10.1016/j.jnca.2010.12.016.
40. Prakash, R., Shivaratri, N., & Singhal, M. (1999). Distributed dynamic fault-tolerant channel allocation for mobile computing. *IEEE Transactions on Vehicular Technology*, *48*(6), 1874–1888.
41. Thathachar, M. A. L., & Sastry, P. S. (2002). Varieties of learning automata: An overview. *IEEE Transactions on Systems, Man, and Cybernetics*, *32*(6), 711–722.
42. Omranpour, H., Ebadzadeh, M. M., Barzegar, S., & Shojafar, M. (2008). Distributed coloring of the graph edges. In *IEEE CIS* (pp. 1–5). doi:10.1109/UKRICIS.2008.4798944.
43. http://www.opnet.com/.
44. Tanenbaum, A. S. (2002). *Computer networks* (4th ed.). Englewood Cliffs, NJ: Prentice Hall.
45. Fehnker, A., Glabbeek, R. V., Hofner, P., Mciver, A., Portmann, M., & Tan, W. L. (2013). A process algebra for wireless mesh networks used for modelling, verifying and analysing AODV. Technical report 5513, NICTA.
46. Shamshirband, Sh., Amini, A., Anuar N. B., Kiah, M. L. M., Wah, T. Y., Furnell, S. (2014). D-FICCA: A density-based fuzzy imperialist competitive clustering algorithm for intrusion detection in wireless sensor networks. *Measurement, 55*, 212–226.

Mohammad Shojafar is currently a Ph.D. student in Information and Communication Engineering at DIET Dept. of the "La Sapienza" University of Rome. He Received his Msc in Software Engineering in Qazvin Islamic Azad University, Qazvin, Iran in 2010. Also, he Received his Bsc in Computer Engineering-Software major in Iran University Science and Technology, Tehran, Iran in 2006. His current research includes wireless communications, distributed computing and mathematical/AI optimization. He is an author/co-author of 30+ peer-reviewed publications (h-index = 7) in well-known IEEE conferences (e.g., PIMRC, ICC, ISDA, HIS) and journals in IEEE, Elsevier and Springer publishers. Since 2013, he is the membership of IEEE Systems Man and Cybernetics Society Technical Committee on Soft Computing and a Distinguished Lecturer of IEEE Computer Society representing Europe. Recenetly, he served as a reviewer and international program committee member in NaBIC2014, SoCPaR 2014, WICT 2014. He has received fellowship award of foreign nationals educated abroad academic year 2012–2015 for "La Sapienza" University of Rome. In addition, Mohammad was a Programmer and Analyzer in Exploration Directorate Section at N.I.O.C in Iran from 2012–2013. Email: shojafar@diet.uniroma1.it and follow more information at: www.mshojafar.com

Saeid Abolfazli has newly received PhD from faculty of Computer Science and IT, University of Malaya, Malaysia. His PhD was fully funded by Malaysian Ministry of Higher Education, Malaysia under High Impact Research project entitled Mobile Cloud Computing: Device and Connectivity from 2011–2014. Dr. Abolfazli received M.Sc. (Information Systems) in 2008 from India and B.E. (Software Engineering) in 2001 from Iran. Saeid has been serving as CEO of Espanta Information Complex during 1999–2006 in Iran when he was a part time lecturer to the ministry of education and Khorasan Technical and Vocational Organization. Dr Abolfazli has been a TPC member for several international conference and reviewer of number of international journals of computer science such as IEEE TPDS. He is a member of IEEE society and IEEE CS Cloud Computing STC. His main research interests include Parallel and Distributed Computing Systems, Mobile Cloud Computing, and Mobile Data Analytics.. Visit him at http://mobilecloudfamily.com/saeid

Habib Mostafaei received the B.S. degree from the Islamic Azad University Khoy branch, in 2006 and the M.S. degree from the Islamic Azad University Arak branch, in 2009, both in software engineering. He joined the faculty of the Computer Engineering Department at Urmia Azad University in 2009. He is a reviewer for Journal of Network and Computer Applications (JNCA), Computer & Electrical Engineering (COMPELECENG), Wireless Networks (WINET), and Computing (COMP). His research interests include learning systems, sensor networks, soft computing, and cloud computing. Email: h.mostafaei@iaurmia.ac.ir

Mukesh Singhal is the Professor and Gartner Group Endowed Chair in Network Engineering, Dept. of Computer Science, The University of Kentucky, Lexington, USA; and Chancellor's Professor School of Engineering of University of California, Meced, USA. He has authored and published over 250 research papers and four textbooks which 189 of them are indexed in DBLP. The books on emerging topics that Dr. Singhal edited include, Distributed computing, Cloud computing, Security and Networks. He is Journal Editorships of five outstanding journals such as IEEE Trans. on Dependable and Secure Computing and IEEE Trans. on Parallel and Distributed Systems. He is one of the highly cited authors in computer science and software engineering worldwide (h-index = 45, Citations = 7171).