A Dynamic Channel Switching Policy Through P-learning for Wireless Mesh Networks

Md. Kamal Hossain[1, 2], Tan Chee Keong[2], Lee Ching Kwang[2], Yeoh Chun Yeow[2]
[1] Lab Analytics & Computation, Telekom Malaysia Research & Development
Cyberjaya, Malaysia
[e-mail: kamal@tmrnd.com.my]
[2] Faculty of Engineering, Multimedia University
Cyberjaya, 63100 - Malaysia
[e-mail: md.kamal.hossain@mmu.edu.my, tckeong@mmu.edu.my, cklee@mmu.edu.my]
[3] Lab Next Generation Access, Telekom Malaysia Research & Development
Cyberjaya, Malaysia
[e-mail: yeow@tmrnd.com.my]
*Corresponding author: Md. Kamal Hossain

Received June 25, 2015; revised September 17, 2015; revised November 8, 2015; accepted December 1, 2015; published February 29, 2016

Abstract

Wireless mesh networks (WMNs) based on IEEE 802.11s have emerged as one of the prominent technologies in multi-hop communications. However, the deployment of WMNs suffers from serious interference problem which severely limits the system capacity. Using multiple radios for each mesh router over multiple channels, the interference can be reduced and improve system capacity. Nevertheless, interference cannot be completely eliminated due to the limited number of available channels. An effective approach to mitigate interference is to apply dynamic channel switching (DCS) in WMNs. Conventional DCS schemes trigger channel switching if interference is detected or exceeds a predefined threshold which might cause unnecessary channel switching and long protocol overheads. In this paper, a P-learning based dynamic switching algorithm known as learning automaton (LA)-based DCS algorithm is proposed. Initially, an optimal channel for communicating node pairs is determined through the learning process. Then, a novel switching metric is introduced in our LA-based DCS algorithm to avoid unnecessary initialization of channel switching. Hence, the proposed LA-based DCS algorithm enables each pair of communicating mesh nodes to communicate over the least loaded channels and consequently improve network performance.

Keywords: Dynamic channel switching, machine learning, switching delay, WMN

http://dx.doi.org/10.3837/tiis.2016.02.010 **ISSN : 1976-7277**

1. Introduction

One of the major issues with WMNs is the capacity problem due to interference which is mainly caused by multiple simultaneous transmissions. Conventional approaches use mesh routers (MRs) with multiple interfaces which can be tuned to non-overlapping channels to alleviate interference. Since the number of channels are limited and co-located wireless networks may be tuned to the same channels, it is desirable to allocate and reallocate channels dynamically to reduce the effect of interference. An effective solution to manage network resources and mitigate the interference is to apply DCS which enables WMNs to avoid channels with serious interference by switching to channels with less severe interference [1][2]. The DCS in WMNs allows a wireless interface to operate in different channels during different time slots without interrupting network connections [2]. In addition, the DCS also enables a wireless MR to physically interact with multiple MRs where each of these wireless routers operates in different channels. This additional flexibility provided by the DCS yields two distinct advantages. Firstly, it greatly increases a WMN router's effective connectivity beyond the number of physical routers it originally has. Thus, it provides more room for optimization on routing protocols. Secondly, the DCS also enables each pair of interacting mesh nodes to communicate over the least loaded channels and improves the channel utilization subsequently. It is noteworthy that conventional DCS algorithms entirely neglect the past experience or data which can be obtained from the system. In addition, exhaustive search to find an optimal solution (optimal channel) followed by those algorithms are generally time-consuming which results in high inefficiency in DCS for WMNs. In this paper, an alternative approach to resolve the above-mentioned issues with DCS is proposed for WMNs. In this approach, the LA-based DCS obtains the optimal channel through *P*-model based reinforcement learning and learning is considered to be the effective approach to find optimal channel to this DCS problem [3][4].

One of the fastest reinforcement learning algorithm known as the Discretized Generalized Pursuit Algorithm (DGPA) [4], which is widely deployed for static environments [5] [6]. Due to non-static nature of wireless environment, mesh nodes may have to change their selected channels dynamically. Hence, modifications have to be made in the conventional DGPA, to adapt efficiently with the variations in stochastic environment and its channel conditions. The proposed changes are to track the considerable drop in channels' performance. In this way, any significant performance drop can be detected in order to clear the learned data (action probabilities); trigger the algorithm to re-learn and find the next available optimal channels for the node pairs. In addition, to avoid unnecessary initiation of channel switching, the proposed algorithm is included with a novel switching metric which considers switching delay to make decision whether switching to another channel will be beneficial or not. If switching to another channel improves network performance even though the switching latency and extra protocol overhead will be introduced, the system will proceed to switch the channel. It is believed that the proposed learning based DCS algorithm will enable WMNs to switch to more idle channels while unnecessary initialization of channel switching will be minimized.

The rest of the paper is organized as follows. First, Section 2 presents some related works. In Section 3, a short review on learning automata and Discretized Generalized Pursuit Algorithm (DGPA) are presented. Our proposed system model, the problem statement and the LA-based DCS algorithm are presented in Section 4. To effectively accommodate the proposed switching algorithm into HWMP, an enhancement is also described in this Section . Numerical and obtained simulation results are reported in Section 5. To complete, this chapter ends with some concluding remarks in Section 6.

2. Related Work

The interference problem which affects channel capacity in WMNs has been investigated. Recent studies reported show that the interference problem can be minimized significantly by equipping mesh nodes with multiple interfaces which are configured to different channels to allow multiple simultaneous transmissions [7] [8].

In multi-channel multi-interface WMNs, channel assignment (CA) is employed to minimize the interference and optimize the network performance by finding optimal channel. Several studies in WMNs assume static channel assignment where the algorithms switch channel periodically or permanently [9] [10]. However, those algorithms perform poorly in a network with dynamic traffic loads. Hence, current researches are mostly centered around dynamic channel assignment (DCA) to select channels dynamically in a distributed manner. The crucial design issue with these DCA algorithms is to decide when to carry out channel switching and which channel to switch to. Intuitively, an efficient DCS policy is essential for the routing protocols to resolve this problem.

In the conventional DCS algorithms, the channel is switched when the network load exceeds the link capacity or a predefined threshold [11] [12] [13]. However, it is difficult to obtain the accurate estimation of link/channel condition due to co-channel interference (CCI) and adjacent channel interference (ACI) in WMNs. In some cases, the inaccurate estimation of link capacity or channel condition may cause the frequent channel switching. Such frequent channel switching might increase network congestion and end-to-end packet delay. Moreover, unnecessary initialization of channel switching creates additional delay and protocol overheads which might lead to a significant performance degradation. Hence, it is necessary to take into account the trade-off between the delay introduced by the DCS and the improvement achieved by the DCS. Therefore, a control mechanism is essential for the DCS algorithm to enable efficient channel switching which can achieve an excellent tradeoff.

3. Learning Algorithm

3.1 Learning Automata

In classical control theory, control mechanism is designed based on a priori information about the control process and can be modeled deterministically. Hence, the expected output of the process due to the current input can be predicted in a deterministic fashion. However, if the information is uncertain or incomplete, the assumptions made on uncertainties may be insufficient to successfully control the system. Therefore, later development of stochastic control theory takes into consideration the uncertainties in the system [14] [15]. However, the characteristics of the uncertainties may not be known completely in a priori if it changes frequently. Hence, it is necessary to observe the process in operation and acquire additional on-line information since a priori information is not sufficient. In such cases, learning is carried out progressively during real-time operation to obtain further updated information about the system. By using the past experience and corresponding outputs, learning algorithm known as LA is expected to determine an appropriate strategy of selecting suitable actions to improve the performance gradually. The idea of designing the learning system is to improve the choice of actions with respect to time towards an ultimate goal with or without the complete knowledge of the system. Mathematically, the goal of a learning system is the optimization of a function which is not known explicitly [15]. One of the main advantages of the reinforcement learning is that it does not require any priori information about the environment except reinforcement data over conventional techniques. Several approaches have been developed to model learning system to study their utilities [14] [15]. Among them, LA theory has gained significant attention and has been applied extensively in various aspects of networking and communication technology [16][3] [17][18].

LA is a general model which makes decision using the concept of an automaton where the automaton operates in an unknown stochastic environment. Generally, in a system where there are definite numbers of actions available in a random environment to be selected, the probability of selecting those actions depends on their internal states. If an action is selected by the LA, the system feedbacks a reinforcement signal. The reinforcement signal which is generated from the environment has a fixed unknown probability distribution. The LA then updates the internal states based on the received reinforcement signal. Simultaneously, the LA also evolves to several expected behaviors. Apparently, the random environment can be depicted as a black box as shown in **Fig. 1**. The decision maker (i.e., LA) selects an action according to certain rules of LA and waits for the feedback from random environment. Based on the feedback from random environment and past experiences, the LA updates the probability of internal states. One of the attractive features of LA which makes it popular for communication technology is that it can converge to ε-optimal solution with a reasonably fast pace in a random environment [15].

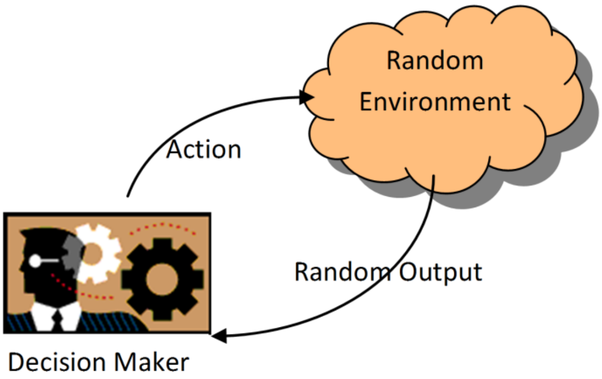


Fig. 1. Learning Mechanism of Learning Automaton

3.2 Behavior of Learning Automata and Reinforcement Scheme

The performance of LA is affected by the reinforcement schemes for the updating technique of the action probabilities. Therefore, it is necessary to relate the structure of reinforcement schemes and the performance of automaton using the schemes. In general, a learning automaton can be classified based on reinforcement schemes used or exhibited behaviour [14]. The learning paradigm of learning automaton interacts with the environment in a sequence of time steps. These steps can be referred as finite set of actions which are selected under a stochastic environment. After executing an action, the system generates an arbitrary reply which can be either favourable or unfavourable.

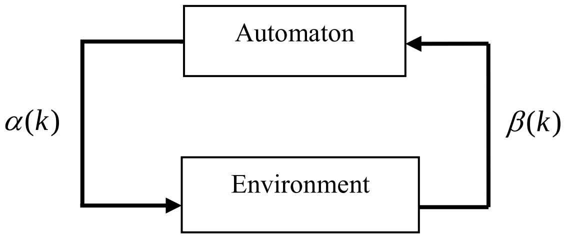


Fig. 2. Block Diagram of Automaton and Environment

At any specific time instant k, the automaton selects an action $\alpha(k)$ which is chosen arbitrarily depending on probability distribution. The input action receives a feedback (or outputs from the environment) $\beta(k)$ which acts as an input signal to the system. Subsequently, the LA updates its action probability $p(k+1)$ from $p(k)$ on the basis of allowable responses from the environment. Unfortunately, the convergence of these algorithms is relatively slow and hence constitutes a restraining factor in practical implementation. To improve the convergence rate, discrete probability space was applied in [4] by limiting the probability of selecting an input value within the interval [0, 1]. The key property of these algorithms is that they preserve running estimates for the reward probability of all possible actions, and use these reward probabilities to update the action probabilities.

A set of replies generated from system, is considered to be real numbers. In a *P*-environment model, environmental responses $\beta(k)$ is limited to only two binary values. Therefore, the *P*-model has a faster convergence rate and possesses lower complexity. In this environment model, $\beta(k)$ = $\{0,1\}$ while $\beta = 1$ usually refers to a favourable response which is considered as *reward* and $\beta = 0$ usually refers to as unfavourable response from the environment which is considered as *penalty*. There are two other environmental models presented in literature, i.e., *Q*-model and *S*-model respectively [14]. In the *Q*-model, the response from environment is a finite set within the interval [0, 1]. Unlike the *P*-model, the *Q*-model possesses more than two values. Besides, *S*-model's environment responses are continuous random variable of real numbers with equal difference. Due to its fast convergence rate [4] [15], the *P*-model is considered here where the reward probability d for a chosen action a is given as:

$$d(k) = \text{Prob}[\beta(k) = 1 \,|\, \alpha(k) = \alpha] \tag{1}$$

The DGPA is used as reinforcement scheme in the proposed LA-based DCS algorithm. DGPA is introduced as "estimator" algorithm to develop a faster converging learning algorithm where the past experience and reinforcement signal received are used to calculate the reward estimators where $d(k)$ is the estimates of the reward probability. It generalizes the perception of the pursuit algorithm by "pursuing" all the actions which possess higher estimates than the currently selected action. A detailed description about the algorithm can be found in [4] [19].

4. System Model and Problem Formulation

4.1 Network Assumptions

In this work, the DCS is modeled based on a learning automaton as described in Section 3. In this model, MRs in the WMN are stationary and possess multiple interfaces where *C* number of channels are available for every interface. Each interface of a node can be tuned into different channels; hence simultaneous transmission is possible without interfering each other in the WMN. To study the feasibility of the proposed LA-based DCS, data traffic is not directed to any specific gateway; instead any node can be sender or receiver for the traffics in this network.

The following assumptions are made for the proposed model:
1) First interface of the MR in WMN is tuned to a common control channel to exchange control messages and the control channel is opportunistically used for data traffic;
2) Adjacent nodes can communicate with each other if they are listening to a common channel;
3) Every node is capable of switching channel at the dynamic manner;
4) All the channels are operated in half-duplex mode.

4.2 Problem Formulation

The conventional CA algorithms assign an optimal channel for communicating node pairs to achieve best network performance. Recently, there is considerable attention to explore node's ability to switch channel dynamically. If the link's traffic load exceeds link capacity or performance drops significantly, the corresponding nodes either reduce the data rate or switch to another channel to facilitate the traffic demand. If the link's capacity is degraded due to interference (e.g., CCA and ACI), it is preferable to switch to other channels with less interference. However, how to switch the channel and which channel to switch is crucial to decide. Switching latency due to synchronization overhead is also another important issue which should be considered while designing an efficient DCS algorithm. Most of the conventional DCS algorithms tend to ignore switching delay overhead, assuming that switching delay will become shorter overtime with technological advancements [1][2]. As it is feasible to think that switching delay will be shorter progressively; the transmission speed is also expected to get faster by then. Therefore, the aggregate loss of bandwidth due to switching delay cannot be tackled completely [20]. Hence, an efficient DCS must be aware of the trade-offs between benefit of channel switching and the drawbacks due to switching latency introduced by channel switching.

Most of the existing works analyze the DCS problem based on a static model. In practice, due to dynamic nature of wireless environment those assumptions may not hold in real WMNs. In this work, an LA-based DCS is proposed to learn the real environment and the algorithm is trained to decide when to switch the channel taking the channel's performance and switching delay into account.

4.3 System Overview

The proposed LA-based DCS algorithm has three key modules: 1) a channel learning module to find the optimal channel for interacting nodes by learning real environment; 2) a performance tracking module to track significant drop of performance of the currently assigned channel; and 3) a switching metric module to decide whether to switch to a new channel or not. In the Performance Tracking module as shown in **Fig. 3**, a control loop is used to feedback the current channel performance. In a specific time frame, if there is a consecutive drop of performance which implies that there is a significant change in the channel capacity. This scenario could be due to some of the neighboring nodes or co-located wireless network which start to use the same channel where the strong interference source comes into existence within its transmission range. Hence, it is necessary to switch to another better channel to avoid interference and further performance degradation. In this context, an efficient channel switching mechanism is essential to assist every node to switch to the optimal channels. If channel switching is triggered, the algorithm will reset the learning parameters in order to clear the previously updated action probabilities of the LA-based DCS algorithm and re-learn the environment to find the next optimal channel [6]. In this model, switching metric is equipped with the ability to make the trade-off decision whether it will be beneficial to execute the channel switching. The LA-based DCS algorithm will initialize channel switching only if network performance improves.

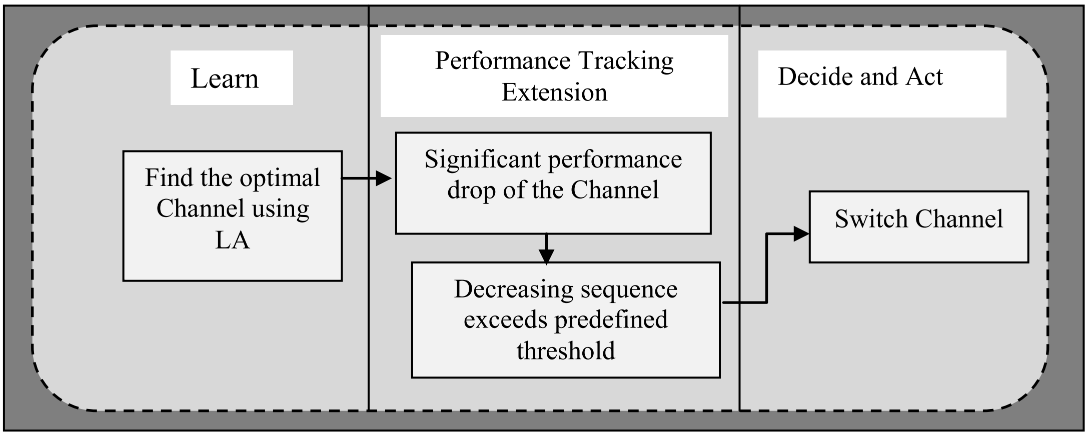

Learn
Find the optimal Channel using LA
Performance Tracking Extension
Significant performance drop of the Channel
Decreasing sequence exceeds predefined threshold
Decide and Act
Switch Channel

Fig. 3. System Overview

4.4 Learning based Channel Switching Algorithm

The DCS problem is considered for the multi-channel multi–radio WMNs. The DCS in this context involves assigning and switching channels (if necessary) to the radio interfaces with the objective of achieving optimal channel utilization and reduce the interference. It is very challenging to achieve the optimal performance because the number of radio interfaces per node is relatively smaller than the number of available channels and also due to channel inter-dependency among the nodes [21]. Hence, finding an optimal channel for the communicating nodes is pivotal for the performance improvement and reliable connectivity [22][23]. Essentially, learning is envisioned as one of the effective ways to find the best channels in this vicinity [3]. The proposed *Learn* module of the system finds the optimal channel by directly interacting with the environment in a distributed manner.

The 802.11b radio-based WMN is considered in this work where the available spectrum is divided into C channels. Each node in the WMN can adjust the channel index denoted as $i \in \{1, ..., C\}$. The traffic generated at the source node is transmitted to the destination node in a subsequent time slots. Each mesh node, n keeps a probability vector $\bar{P}_n = \left[p_n^1, p_n^2, ... p_n^C \right]$ and an estimator vector $\bar{D}_n = \left[d_n^1, d_n^2 .... d_n^C \right]$ in their routing table entry where p_n^i is the probability of selecting channel i and $d_n^{\,i}$ is the estimator probability based on successful transmission if channel i is selected. Initially, a pair of communicating nodes picks a channel according to probability vector $\bar{P}_n$. If the selected channel is used by other neighbouring nodes, it will select next available channel based on $\overline{P}_n$ and starts to transmit data. Based on the received $\beta(k)$ value, the LA updates the probability vector $\overline{P}_n$ and estimator vector $\overline{D}_n$ for its next action. The details of the proposed learning based DCS is summarized below:

BEGIN ALGORITHM LA-BASED DCS:

Parameter:

Notation	Description of the notation
R	resolution parameter
I	minimum number of times a channel has been selected
C	total number of channels available
$H(k)$	number of channels that have higher estimated value than that of the currently selected channels
$\Delta = \frac{1}{CR}$	the smallest step size of adjusting probability vector
$W_n^i(k)$	total number of times transmission with channel $i \in \{1, 2, ...C\}$ is successful up to step k
$Z_n^i(k)$	total number of times the channel $i \in \{1, 2, ...C\}$ is selected up to step k
$\beta_n^i(k)$	$\beta_n^i(k) = 1$ if transmission is successful using channel i on node n, otherwise $\beta_n^i(k) =$ 0 (the action is penalized)

Initialization:

- Set the probability of action to be $\overline{P}_n(0) = \left[p_n^1(0), p_n^2(0), ..., p_n^C(0) \right]$ and $P_n^i(0) = {}^1\!/_C$ which means that initially all the channels have equal probability, where $i \in \{1, 2, 3 ....C\}$ and $\sum_{i=1}^{C} P_n^i(0) = 1$
- Choose an available channel to transmit data according to $\overline{P}_n(0)$, record the response of β_n^i and update $W_n^i(0)$ and $Z_n^i(0)$ until each channel is selected for I times.
- Initialize $\overline{D}_n(0) = [d_n^1(0), d_n^2(0), ...d_n^C(0)]$ where $d_n^i(0) = \frac{W_n^i(0)}{Z_n^i(0)}$

***Update_LA*:**

Step 1.

1: Select an available channel $i \in \{1, 2, 3, ...C\}$ according to probability 2: vector $\overline{P}_n(k) = [p_n^1(k), p_n^2(k), ...p_n^C(k)]$

Step 2.

```
Update the probability vector accordingly:
$\forall\ j \neq i,\ j \in \{1, 2, .....C\}$
$p_n^j(k+1) = \min\left\{ p_n^j(k) + \frac{\Delta}{H(k)}, 1 \right\}, if\ d_n^j(k) > d_n^i(k)$        (2)
$p_n^j(k+1) = \max\left\{ p_n^j(k) - \frac{\Delta}{C - H(k)}, 0 \right\}, Otherwise$        (3)
And then $p_n^i$ is calculated as:
$p_n^i(k+1) = 1 - \sum_{j=1,\ j \neq i}^{C} p_n^j(k+1)$        (4)
```

Step 3.

```
Update the estimator probability for the action selected using following equations:
$W_n^i(k+1) = W_n^i(k) + \beta_n^i(k)$        (5)
$Z_n^i(k+1) = Z_n^i(k) + 1$        (6)
$d_n^i(k+1) = \frac{W_n^i(k+1)}{Z_n^i(k+1)}$        (7)
```

Step 4.

```
Realize if the system convergence point exists at $\bar{k}$ and optimal channel $i$
by checking if $p_n^i(\bar{k}) = 1$ exists.
```

Step 5.

```
If $Q_s(k) < Q_s(k-1)$, start tracking the performance drop of the channel
  and record the length ($l$) of decreasing sequence of $Q_s(k)$
    If $l \geq l_0$, where $l_0$ is decreasing threshold factor
        GoTo Step 6
    Else
        GoTo Step 5
    End If
End If
```

Step 6.

```
If equation (9) = = true
    GoTo Initialization
  Else
    GoTo Update_LA
End If
```

End ***Update_LA***

END ALGORITHM LA-BASED DCS

In *Step* 4, $\bar{k}$ is denoted as the starting point of the convergence. According to the first four steps, the LA-based DCS algorithm shows that if the action probability of one of the channels reaches 1, it is considered as optimal channel and the algorithm converges to a steady state.

Step 5 is the *Performance Tracking* module. The function of this module is to recognize the performance drop by tracking packet loss probability of the current chosen channel. Assuming that after k steps, channel i is selected as an optimal channel for the interacting nodes, where probability of action $,\alpha_i(\bar{k})=1 \;\; \bar{k}>k$. After a certain period T, performance of channel i drops significantly and hence channel $\hat{i}$ is the next best channel for the communicating nodes. Therefore, it is desirable for communicating nodes to switch from channel i to channel $\hat{i}$ to satisfy network traffic demand. In order to achieve it, the DGPA algorithm is modified to track the channel performance [6]. The proposed modification recognizes a considerable performance drop of the channel due to collision and channel error. According to the proposed method in [9], the probability of a packet successfully transmitted can be defined as Q_S which is the inverse of packet loss probability and it is used as performance metric to track the channels performance in this work. In general, Q_S can be expressed as:

$$Q_S=(1-Q_C)(1-Q_e) \tag{8}$$

where, Q_C is probability of collision and Q_e is probability of channel error. In equation (8), Q_C and Q_e are considered to be mutually independent. When two nodes start to transmit simultaneously or overlaps in time where the interacting nodes are within the transmission range, a collision may occur. However, the occurrence of channel error is due to low signal to noise ratio (SNR) of the received packet which is caused by large path loss or severe interference. Authors in [9] claim that it is not necessary to differentiate between collision and channel error probability; rather their proposed technique to approximate probability that the future packets will collide is more effective to estimate channel conditions. Hence, the proposed collision probability estimation technique is adopted to track the channel performance.

The final stage of the proposed LA-based DCS algorithm is switching metric (*Step 6)* module where the communicating nodes will decide whether to switch channel. According to *step* 5, if decreasing sequence (l) of a channel is greater than or equals to the l_0, it will proceed to *step 6*, i.e., the switching metric module. To obtain an optimal channel, the LA-based DCS algorithm learns the environment which may require a certain number of channel switching and this generates redundant protocol overheads. Moreover, unnecessary channel switching can increase latency and degrade overall network throughput. Hence, the switching metric is proposed to approximate whether channel switching will be beneficial or not. Under certain network condition, if the time difference to transmit m data packets using currently degraded channel capacity and the current channel's optimal capacity is greater than the total switching latency as shown in equation (9), the algorithm will be triggered to switch channel. The proposed switching metric is devised as:

$$mf_i C\left(\frac{1}{c_{\text{current}}(n,i)}-\frac{1}{c_{\text{optimal}}(n,i)}\right)>\delta N \tag{9}$$

where m is number of data packets to be transmitted on each channel for the initialization of LA-based DCS, f_i is the size of the data packet/frame in bits, δ is channel switching latency, N is total number of channel switching required to initialize the proposed LA-based DCS, $c_{\text{current}}(n,i)$ and $c_{\text{optimal}}(n,i)$ are the link capacities for node n on channel i at the current and optimal channel condition, respectively. The link capacity can be given as (Shannon's capacity theorem):

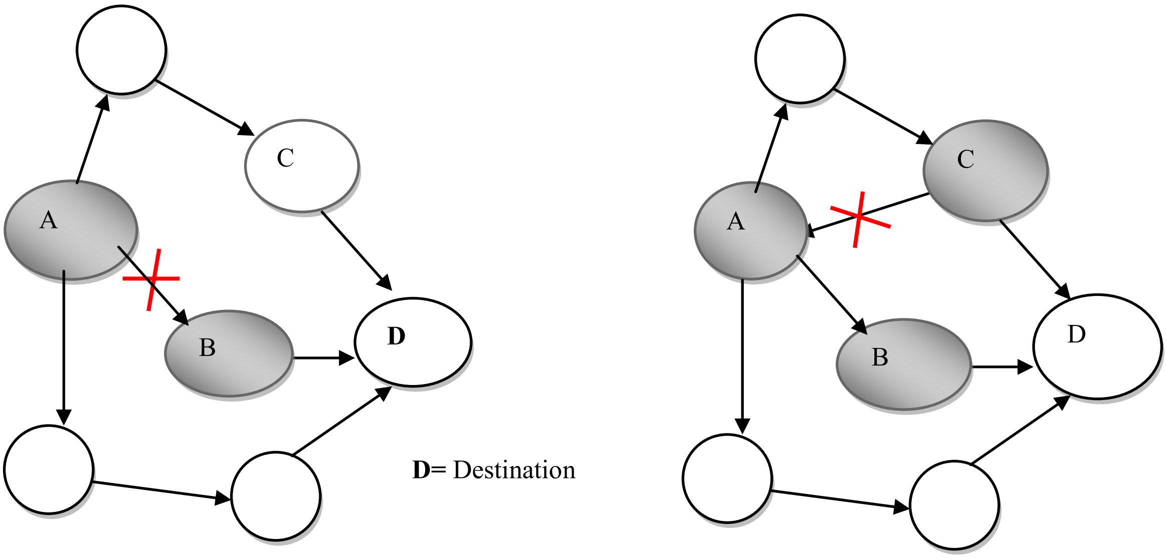
$$c(n,i)=w\log_2(1+\gamma_{n,i}) \tag{10}$$

where w is the bandwidth of the channel selected and $\gamma_{n,i}$ is SNR value of the node n on channel i. If channel switching is triggered, the algorithm will proceed to '*Initialization*' stage to re-learn and obtain the next optimal channel.

4.5 Enhancement in Routing Protocols

Even though the default routing protocol (i.e., HWMP) of WMNs supports multiple interface; it was typically designed for multi-home based protocols instead of multi-interface network [1]. Hence, it does not have the capability to adapt the frequent channel fluctuations. Moreover, the proposed DCS algorithm requires the mesh nodes to switch channel in a dynamic manner. Therefore, some modifications are introduced in the original default routing protocol. Firstly, the reasons are demonstrated here, why the existing routing protocol cannot be used effectively for the proposed DCS algorithm.

In WMNs, all the mesh nodes keep the connectivity by exchanging "*Hello*" information; for example, node A has sent the "*Hello*" message; the neighbours who receive the "*Hello*" message will add node A into their routing table entries. The entries for the routes will be remained for future use, until that expiry. Two possible scenario are discussed here: 1) If node A changes its channel, but the corresponding neighbour node, say B does not switch to the new channel; node A and B will not be able to communicate with each other after the switch (**Fig. 4** (a)). 2) If nodes A and B change the corresponding channel to a new channel, the neighbour node, say node C is not aware of the new channel, thus, node C cannot connect with A or B (**Fig. 4** (b)).



(a) Node A has changed channel, but node B did not switch yet

(b) Nodes A and B have changed channels other than the current channel

Fig. 4. The Routing Table Entry Remains Valid but the Actual Link is Broken

However, according to the routing table entries in those nodes, the corresponding routes are still valid. Unfortunately, the packets cannot be transmitted because the actual link is broken. Therefore, after maximum number of retries HWMP will initiate route discovery. It takes time to discover a new route; subsequently packets will be queued and delayed. Therefore, the default HWMP is modified to acquire channel index information of the neighbouring nodes. To resolve the issue, an additional field of channel index is introduced in the entry of routing table and also in RREQ message. Whenever a node switches to another channel, RREQ message will be broadcast with the new channel index information and the corresponding neighbouring nodes which receive the RREQ will update the routing table entries with the new channel index info. Therefore, even if neighbouring nodes change channels, it still able to find right route and channel index by referring to routing table and establishes a working link without initiating route discovery. The proposed modifications have been introduced into the default HWMP (routing table and RREQ packet format) to effectively accommodate the proposed LA-based DCS algorithm. This modified routing protocol is named as Enhanced-HWMP (E-HWMP) in this paper.

5. Performance Evaluation

In this section, NS-3 simulator is used to evaluate the performance of the proposed algorithm [24]. The performance of the E-HWMP (which is incorporated with the proposed LA-based DCS algorithm) is compared with the default IEEE 802.11s based HWMP routing protocol [25]. The default IEEE 802.11s based HWMP routing protocol is modified to possess random channel switching (HWMP-RCS) capability and also exhaustive searching approach to find an optimal channel to switch (HWMP-ESCS). In HWMP-RCS, channel switch is executed randomly as its traffic on that channel is above a certain threshold and immediate neighbour nodes are not tune to same channel. For HWMP-ESCS, if channel condition drops below a certain SNR threshold, the current node searches all the available channels and switch to the next optimal channel based on highest SNR value. In this paper, the performance of the E-HWMP is compared with both of these modified protocols to demonstrate the superiority of our algorithm.

The E-HWMP uses a common channel for exchanging control messages to overcome connectivity problems in WMNs. The corresponding channel has also been used for data traffic opportunistically. A network of 30 nodes is placed randomly in 1000 m × 1000 m area and performance comparison of E-HWMP is observed in a multi-channel multi-interface WMN. The transmission range of every mesh node is adjusted to be 100 m; hence two mesh nodes which are within radius of 100 m can communicate with each other. However, those nodes which are within 200 m apart and tuned to the same channel will experience interference due to overlapping channels. Each interface is assumed to have 11 channels (as IEEE 802.11b radio is used) where the first 10 channels are used for only data traffic and channel 11 is used as common channel to exchange control messages and opportunistically is used for data traffic. All transmissions are based on User Datagram Protocol (UDP), the packet size is fixed to 128 bytes and data rate is 25 packets per second. The channel switching delay is assumed to be 80 μ sec and the NS-3 default log-distance path loss model for the channel is considered. Each node has up to two interfaces; one interface can switch channels dynamically for data traffic and the other one is fixed to default control channel. The number of interfaces for data transmission can be changed easily and can accommodate other routing protocol without any significant modifications.

Now, a simple example is provided to prove the effectiveness of the proposed LA-based DCS algorithm in WMNs. In the network scenario, every mesh node is allowed to switch channel dynamically within 10 available channels. Therefore, the LA has 10 set of action components. The corresponding probabilities of action components are $\overline{P}_n = [p_n^1, p_n^2, ..., p_n^C]$. The other simulation parameters to initialize LA are:

- Resolution Parameter, $R = 5$
- Iteration or minimum number a channel has been selected, $I = 7$
- Total number of frequency channels available, $C = 10$
- The smallest step size for adjusting probability vector, $\Delta = 0.02$

For illustrative purpose, let's assume that the probability vector of selecting channels for any specific node after some iteration steps is $\overline{P}_n = [0.06, 0.08, 0.1, 0.30, 0.16, 0.02, 0.04, 0.14, 0.08, 0.02]$ where channel 4 (probability distribution of the channel, p_n^4 is 0.30) has the highest chance to be selected at this time. Hence, based on probability distribution, channel 4 is the optimal channel considering interference and link quality. However, LA updates the probability of actions continuously and, it gradually converges to the optimal channel for the interacting nodes. Therefore, initially channel 5 was not optimal channel but is converges as optimal at the end as shown in **Fig. 5**. The algorithm also has been tested for a lower number of available channels (5 frequency channels) to check the convergence rate as shown in **Fig. 6**. Based on the observations, LA converged here to an optimal channel within first 120 step size in the WMN.

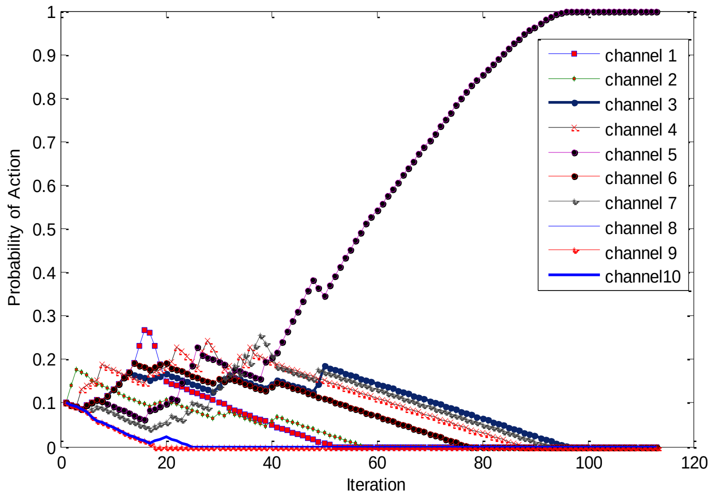


Fig. 5. The Probability of Action Updating History of LA-Based Channel Switching Algorithm (10 Channels)

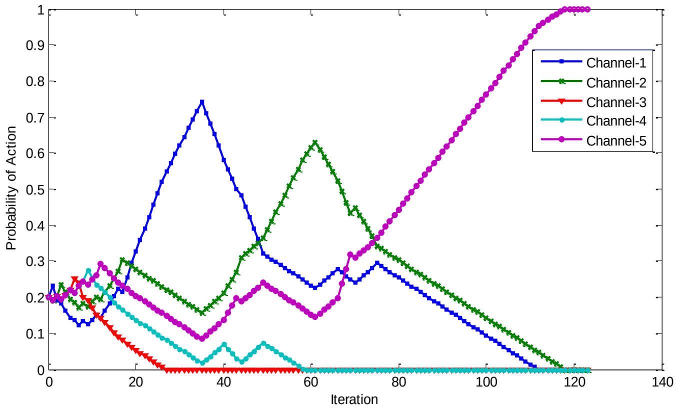


Fig. 6. The Probability of Action Updating History of LA-Based Channel Switching Algorithm (5 Channels)

After obtaining an optimal channel, the mesh interface continues to use the selected channel. In practice, due to dynamic nature of the wireless environment, the channel quality fluctuates quite frequently. Moreover, some other mobile devices or network may tune to the same channel within the transmission range and it results in interference. Hence, it is necessary to switch to other channels for performance improvement and to meet the traffic demand. Conventional DGPA is not capable of adapting dynamic changes of wireless environment [6]. Hence, the DGPA which is used in the proposed LA-based DCS algorithm; is modified to track the channel performance using equation (8) and switch channel if there is consecutive drop in the probability of successful transmission and the drop sequence exceeds the consecutive decrement factor (l_0). In the proposed model, consecutive drop of performance is considered as an indication that there is a significant change in the channel quality.

As seen in **Fig. 7**, after about $K = 70$ steps there are some consecutive drop points of the performance metric at 74, 84, 90 and 96 but performance metric at $K(90) < K(96)$. Hence, those four steps can only form a decreasing sequence of 3: $\{K(74), K(84), K(90)\}$ where l=22 only. However, if the figure is observed, at $T_0 > k$, the number of drop points form a longer decreasing sequence than $\{K(74), K(84), K(90)\}$ and there is a significant drop in performance metric. To recognize the changes in the performance, the drop points and the length of decreasing sequence of the consecutive drops are recorded. If the length of consecutive decreasing sequence (l) is greater than or equals to decreasing threshold factor (l_0), this indicates that there is a significant changes in current channel's performance. Hence, channel switching will be triggered to avoid further performance degradation. In this experiment, simulations are conducted for different l_0 to obtain the optimal value for l_0 using the aforementioned WMN scenario. It has been observed that when $20 \le l_0 \le 30$, it shows the best performance in improving throughput and end-to-end delay. Hence, l_0=25 is chosen in our simulation. If the *Performance Tracking* module detects significant degradation, it will trigger the switching metric (*Decide and Act*) module to make the final decision on channel switching.

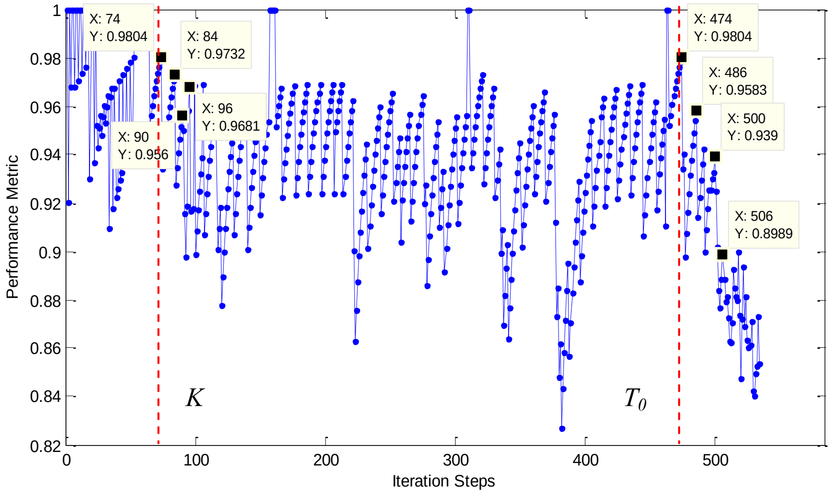


Fig. 7. The Optimal Channel's Performance of a Dynamic WMN Environment

To measure the effectiveness of the switching metric module, the network is simulated with and without the switching metric module. When switching metric module is not included, the LA-based DCS algorithm initializes channel switching as soon as *Performance Tracking* module detects significant performance drop. **Fig. 8** depicts the comparison results of the proposed LA-based DCS algorithm with and without the switching metric module. From the figure, it is evident that the algorithm performs better and approximately 30% of improvement is reported when the switching metric module is included in the proposed LA-based DCS algorithm. This is because switching metric module reduces unnecessary initialization of channel switching. Hence, less network congestion and packet delay in the network are obtained.

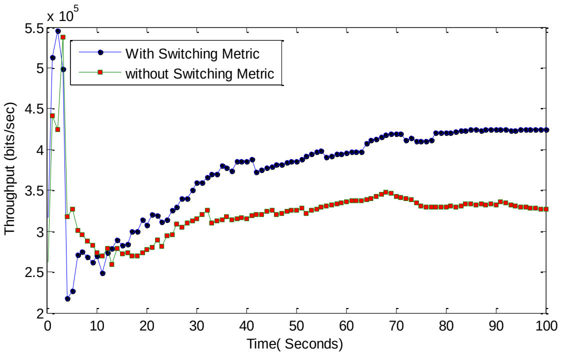


Fig. 8. Performance Comparison of LA-Based Switching Algorithm (with and without Switching Metric)

The performance of the E-HWMP is compared with HWMP-RCS and HWMP-ESCS. In this analysis, each mesh node is equipped with multiple radios; the throughput comparison is presented for two radios in this context. **Fig. 9** shows that the performance of the E-HWMP compared to the HWMP-RCS and HWMP-ESCS. There is a notable difference in aggregate throughput, because the proposed LA-based DCS finds an optimal channel for communication links by learning real environment which implicitly considers all the interference and hidden node problems. Moreover, if the performance of current channel drops significantly, the LA-based DCS algorithm switches to an idle or vacant channel to improve the network performance. The proposed novel switching metric also plays a vital role to reduce unnecessary initialization of channel switching, hence reduce overhead and ultimately improve the throughput. Although HWMP-ESCS approach searches all the available channels to obtain the best one in terms of capacity and link quality; during exhaustive searching process certain channel might be good and recorded as best channel for the communicating nodes. But after the searching process, the channel may not be the best channel anymore due to dynamic nature of wireless environment. According to HWMP-ESCS, it will choose the channel as ranked by exhaustive searching approach even though it may not be the best available channel at that time. Besides, every time HWMP-ESCS search all the available channels to obtain optimal channel to switch which creates additional overhead and degrade the performance. In HWMP-RCS, it chooses channel without being aware of the channel condition. Hence, it might end up choosing a highly congested and low capacity channel, consequently which results in low throughput performance. From the figure, it also shows lowest throughput performance compare to other techniques.

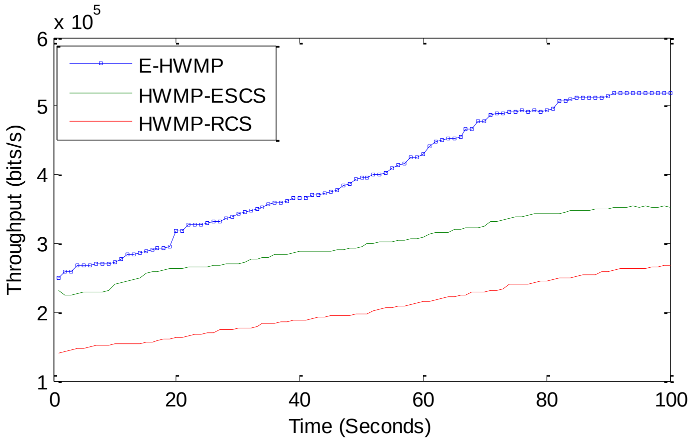


Fig. 9. Throughput Comparison between HWMP, HWMP-ESCS and HWMP-RCS

The simulation is also conducted to analyse the end-to-end delay and packet loss rate in a WMN where each mesh node is equipped with two radio interfaces. The end-to-end delay and packet loss rate comparison are illustrated in **Fig. 10** and **Fig. 11** respectively. In terms of end-to-end delay, the E-HWMP reduces unnecessary initialization of channel switching which subsequently reduce packet delay. HWMP-ESCS search all the available channels to find optimal channels and if the channel does not meet the threshold requirement, it will initiate another exhaustive search to find next optimal. After the expensive searching approach the selected channel might not be optimal one due to dynamic nature of the wireless environment which was assumed to be the best channel at the time of searching. Hence, it shows higher end-to-end packet delay compared to E-HWMP. In HWMP-RCS, channel is selected randomly and hence it might choose highly congested and loaded channel while switching channel. So, it has highest end-to-end packet delay compare to other techniques.

In terms of packet loss rate, the E-HWMP finds optimal channel for communicating node pairs and changes the channel if there is a significant drop of performance using current channel. Thus, the channels have higher SNR which implies that the BER is low. Hence, lower packet loss rate in WMN is achieved. From the **Fig. 11**, it can be observed that the E-HWMP performs much better in multi-channel multi-radio WMN environment in terms of packet loss rate compared to the HWMP-ESCS and HWMP-RCS.

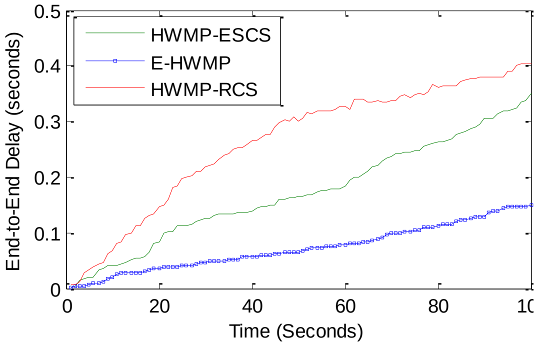


Fig. 10. End-to-End Delay

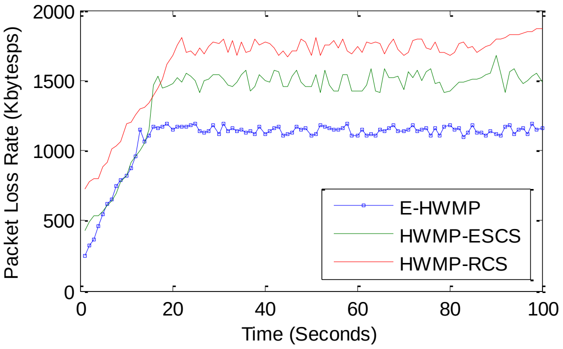


Fig. 11. Packet Loss Rate

6. Conclusion

In this paper, a novel approach to solve the DCS problem has been presented for multi-channel multi-radio WMNs. The proposed LA-based DCS algorithm assigns the optimal channels to the communicating nodes by using a reinforcement learning scheme based on *P*-learning to minimize network interference and also allows the mesh nodes to dynamically switch to other channels if necessary. The real-time simulations are conducted using NS-3 simulation tool. The benefits gained by the LA-based DCS approach are described as follows.

Firstly, the LA-based DCS provides a realistic, systematic and easy approach to obtain near-optimal channels by using past history and real time online learning. Secondly, the proposed learning is performed online; any unforeseen event or change occurring due to significant variations in channels conditions, such as traffic and interference can be tracked. In this way, the significant performance drop can be detected in order to trigger channel switching and the algorithm can converge to next optimal channels. Thirdly, the proposed LA-based DCS explicitly considers the delay overhead which is incurred due to channel switching. A novel switching metric is designed to decide whether to switch to a different channel with the objective to reduce unnecessary initialization of channel switching.

A modification also has been proposed in default HWMP to effectively accommodate the proposed LA-based DCS algorithm which known as E-HWMP in this paper. The proposed modification can help to find correct routes and establish a working communication link even if they switch channels, without initializing route discovery. Finally, the effectiveness of the proposed LA-based DCS has been demonstrated and comparative studies between E-HWMP, HWMP-ESCS and HWMP-RCS have shown that the E-HWMP can perform much better in terms of network throughput and end-to-end delay in WMNs.

References

[1] X. Li, J. Wu, S. Lin, and X. Du, "Channel switching control policy for wireless mesh networks," *J. Parallel Distrib. Comput.*, vol. 72, no. 10, pp. 1295–1305, Oct. 2012. Article (CrossRef Link).

[2] G. Wu, S. Singh, and T. Chiueh, "Implementation of dynamic channel switching on IEEE 802.11-based wireless mesh networks," in *Proc. of the 4th Annual International Conference on Wireless Internet*, pp. 39:1–39:9, 2008. Article (CrossRef Link).

[3] J. Nie and S. Haykin, "A dynamic channel assignment policy through Q-learning," *IEEE Trans. Neural Networks*, vol. 10, pp. 1443–1455, 1999. Article (CrossRef Link).

[4] M. Agache and B. J. Oommen, "Generalized pursuit learning schemes: New families of continuous and discretized learning automata," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 32, pp. 738–749, 2002. Article (CrossRef Link).

[5] R. Bellman and R. Kalaba, "On Adaptive Control Processes," *IEE Trans. Autom. Control*, vol. 4, pp. 1–9, 1959. Article (CrossRef Link).

[6] Tuan, T. A., Tong, L. C., & Premkumar, A. B, “An adaptive learning automata algorithm for channel selection in cognitive radio network,” in *Proc. of IEEE International Conference on Communications and Mobile Computing (CMC),* Vol. 2, pp. 159-163, 2010. Article (CrossRef Link).

[7] P. Kyasanur and N. H. Vaidya, "Capacity of multi-channel wireless networks: impact of number of channels and interfaces," in *Proc. of the 11th annual international conference on Mobile computing and networking* , pp. 43–57, 2005. Article (CrossRef Link).

[8] H. Skalli, S. Ghosh, S. Das, and L. Lenzini, "Channel Assignment Strategies for Multiradio Wireless Mesh Networks: Issues and Solutions," *IEEE Communications Magazine*, vol. 45. pp. 86–95, 2007. Article (CrossRef Link).

[9] M. N. Krishnan and A. Zakhor, "Throughput improvement in 802.11 WLANs using collision probability estimates in link adaptation," in *Proc. of IEEE Wireless Communications and Networking Conference*, WCNC, 2010. Article (CrossRef Link).

[10] A. Raniwala, K. Gopalan, and T. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8. p. 50, 2004. Article (CrossRef Link).

[11] M. Shojafar, S. Abolfazli, H. Mostafaei, & M. Singhal, "Improving Channel Assignment in Multi-radio Wireless Mesh Networks with Learning Automata," *Wireless Personal Communications*, 82(1), 61-80, 2015. Article (CrossRef Link).

[12] K. Ramachandran, I. Sheriff, E. Belding and K. Almeroth, "Routing Stability in Static Wireless Mesh Networks," in *Proc. of the 8th international conference on Passive and active network measurement*, pp. 73–83, 2007. Article (CrossRef Link).

[13] So, Jungmin, and Nitin H. Vaidya, "A routing protocol for utilizing multiple channels in multi-hop wireless networks with a single transceiver," *University of Illinois at Urbana-Champaign*, 2004.

[14] K. S. Narendra and M. A. L. Thathachar, "Learning Automata - A Survey," *IEEE Trans. Syst. Man. Cybern.*, vol. SMC-4, 1974. Article (CrossRef Link).

[15] K. Narendra and M. Thathachar, "Learning automata: an introduction," *IEEE Trans. Syst. Man. Cybern. B. Cybern.*, vol. 32, pp. 711–22, 2012.

[16] M. A. Haleem and R. Chandramouli, "Adaptive downlink scheduling and rate selection: A cross-layer design," *IEEE Journal on Selected Areas in Communications*, vol. 23. pp. 1287–1296, 2005. Article (CrossRef Link).

[17] Y. Song, Y. Fang, and Y. Zhang, "Stochastic Channel Selection in Cognitive Radio Networks," *IEEE GLOBECOM 2007-2007 IEEE Glob. Telecommun. Conf.*, pp. 4878–4882, Nov. 2007. Article (CrossRef Link).

[18] X. Yiping, R. Chandramouli, and C. Cordeiro, "Price dynamics in competitive agile spectrum access markets," *IEEE J. Sel. Areas Commun.*, vol. 25, pp. 613–621, 2007. Article (CrossRef Link).

[19] M. Thathachar and P. Sastry, “Networks of Learning Automata: Techniques for Online Stochastic Optimization,” *Springer*, vol. 49, no. 0, 2004. Article (CrossRef Link).

[20] R. Chandra, P. Bahl, and P. Bahl, "Multinet: Connecting to multiple IEEE 802.11 networks using a single wireless card," in *Proc. of IEEE INFOCOM*, vol. 2, pp. 882–893, 2004. Article (CrossRef Link).

[21] E. Hossain and K. Leung, Wireless mesh networks: Architectures and protocols, pp. 1–333, 2007. Article (CrossRef Link).

[22] L. Chen, S. Iellamo, and M. Coupechoux, "Opportunistic Spectrum Access with Channel Switching Cost for Cognitive Radio Networks," *2011 IEEE Int. Conf. Commun.*, pp. 1–5, 2011. Article (CrossRef Link).

[23] Yang, J., Wang, Y., Hua, K., & Wang, W., "Fairness based dynamic channel allocation in wireless mesh networks," in *Proc. of IEEE International Conference on Computing, Networking and Communications (ICNC)*, pp. 556-560, 2014. Article (CrossRef Link).

[24] K. Andreev, P. Boyko, and P. B. Andreev, Kirill, "IEEE 802.11s Mesh Networking NS-3 Model," in *Proc. of NS-3 Workshop*, p. 8, 2010.

[25] IEEE P802.11s/D1.08, "Draft STANDARD for Information Technology- Telecommunications and information exchange between systems- Local and metropolitan area networks- Specific requirements- Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Lay," no. January, 2008.

Hossain, **Md. Kamal** received his B.Eng. degree (Electronics majoring in Telecommunication) and M.Eng.Sc (research) degree in Wireless Communication from Multimedia University, Cyberjaya, Malaysia in 2012 and 2015 respectively. He is currently working as Assistnt Researcher (Junior Data Scientist) at Lab Analytics and Computation, Telekom Research & Development, Malaysia. His current research interests include Big Data Network Analytics, machine learning algorithm, general communication theories, cooperative communication, cognitive radios, and wireless mesh networks. He is a student member of IEEE, IEICE and IEEE power & Energy Society.

Chee Keong Tan received his B.Eng (Electronics majoring in Telecommunication), M.Eng.Sc (wireless communications) and Ph.D (wireless communications) from Multimedia University, Malaysia in 2006, 2009 and 2014 respectively. He is currently a senior lecturer in Faculty of Engineering, Multimedia University, Malaysia. His research interests include radio resource management in WiFi, WiMax, mesh networking, wireless ad hoc networks and cognitive radio.

Ching Kwang LEE graduated with B.Sc. and PhD degrees from the University of Kent at Canterbury, United Kingdom, in 1982 and 1987 respectively. He was a research fellow in the areas of microwave antenna; majoring in frequency selective surface, at the above university between 1988 and 1990. He then joined the Electro-Optic Group, Division of Radiophysics (Now renamed as Telecommunications and Industrial Physics), at Commonwealth Scientific Industrial Research Organisation (CSIRO), in Australia as a research scientist from October 1990 to July 1991 working on near field range. He then joined the education sector at the school of Electrical and Electronic Engineering, Nanyang Technological University, Singapore from 1991-2010. Joint Faculty of Engineering been at Multimedia University in Malaysia since October 2010 and Director of Graduate Institute of Engineering since March 2014.

Chun Yeow Yeoh (M’99–SM’01) born in Georgetown, Penang. He received the B.Eng. (Hons.) (First Class) degree in Electrical Electronic and M.Sc degree in Electrical Engineering from the Universiti Teknologi Malaysia, Johor, Malaysia, in 2001 and 2003. His master thesis is TCP/IP stack implementation in Embedded System. He is currently Principal Researcher with TM Research & Development Sdn. Bhd., Malaysia. He has previously worked with Cozybit. Inc. in SanFrancisco Bay Area as staff engineer on 802.11s mesh networking andhas contributed to Linux kernel. He has also worked with Mimos Berhad and has successfully implemented TCP/IP stack for Pesona, the first Malaysian 16-bit microprocessor chip. His research interests include the MAC layer of wireless communication technologies, embedded system, software-hardware co-design and Linux system. Chun Yeow is a registered member of Board of Engineer Malaysia. He is also a member of 5G sub-WG under Malaysian Technical Standard Forum Berhad (MTSFB), a standard writing organization recognized by Malaysian law. He was treasurer of IEEE ComSoc/VT Malaysia Chapter and currently holding the position as ExComm. He has received the best paper award in IEEE ICIT, Shenzhen, China on 2007.