LAMR: Learning Automata based Multicast Routing Protocol for Multi-Channel Multi-Radio Wireless Mesh Networks

Mohsen Jahanshahi[1]
Department of Computer Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran
mjahanshahi@iauctb.ac.ir

Mehdi Dehghan
Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran
dehghan@aut.ac.ir

Mohammad Reza Meybodi
Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran
mmeybodi@aut.ac.ir

Abstract

Multicast routing is a crucial issue in wireless networks in which the same content should be delivered to a group of recipients simultaneously. Multicast is also considered as a key service for audio and video applications as well as data dissemination protocols over the last-mile backhaul Internet connectivity provided by multi-channel multi-radio wireless mesh networks (MCMR WMNs). The multicast problem is essentially related to a channel assignment strategy which determines the most suitable channel-radio associations. However, channel assignment brings about its own complications and hence, solving the multicast problem in MCMR WMNs will be more complicated than that of traditional networks. This problem has been proved to be NP-hard. In the major prior art multicast protocols developed for these networks, channel assignment and multicast routing are considered as two separate sub-problems to be solved sequentially. The work in this article is targeted at promoting the adoption of learning automata for joint channel assignment and multicast routing problem in MCMR WMNs. In the proposed scheme named LAMR, contrary to the existing methods, these two sub-problems will be solved conjointly. Experimental results demonstrate that LAMR outperforms the LCA and MCM proposed by Zeng et al. (2010) as well as the genetic algorithm-, tabu search-, and simulated annealing-based methods by Cheng et al.(2011) in terms of achieved throughput, end-to-end delay, average packet delivery ratio, and multicast tree total cost.

Keyword: wireless mesh networks, multi-channel, multi-radio, multicast, channel assignment, learning automata

1. Introduction

Wireless mesh network (WMN) is a new technology basically targeted to provide wireless Internet access [1, 2]. WMN has many characteristics that distinguish it from the traditional networks; Unlike Ad-hoc networks, infrastructure of this network is usually stationary. Also contrary to Ad-hoc and wireless sensor networks there is no limitation on nodes power consumption. In this network, the cost of each node is not a concern and hence, every node may have many radio interfaces usually less than or equal to the number of available channels [3 and 4]. In this way, the nodes can transmit and receive data through different channels simultaneously [6]. Wireless mesh networks operating with multiple channels on multiple radio interfaces are henceforth referred to in this article as MCMR WMNs. Finally, opposed to wireless sensor and Ad-hoc networks, here, reduction of inter-channel interference is the main challenge.

[1] Corresponding author.
E-mail addresses: mjahanshahi@iauctb.ac.ir

One of the most important applications of MCMR WMNs is Multicast-based systems such as video conferencing, online games, webcast and distance learning, to name a few. Since the nature of wireless communication is based on broadcast characteristic of the air medium, the inter-channel interference in WMNs plays a vital role in determining the actual data rate attained for a multicast service. Using channel assignment strategy which effectively determines the most appropriate channel-radio associations, the interference can be reduced. However, channel assignment brings about its own complications; this means to make a communication properly, in addition to locating the neighboring nodes within the transmission range of each other, another requirement should be met as the existence of a common channel assigned to the radios of both nodes. Since multicast tree construction should be necessarily accomplished according to channel-radio associations in a way the overall interference in the network is kept at minimum, the second constraint essentially complicates the problem. The channel assignment problem has previously been tackled in the context of unicast routing [7-17]; unicast-based interference reduction schemes can be categorized into following classes:

- Disjoint: assignment of channels on a given routing topology [9, 12,15] or vice versa [8, 10, 11, 14]
- Joint: channel assignment and routing conjointly [13 and 17]

Clearly, unicast-based implementations cannot be used or at least are not scalable enough to be served in multicast communications. Moreover, considering the limited bandwidth of wireless networks, the existing wireline multicast solutions cannot be employed in mesh networks without essentially changing their behavior to reduce their overhead. Also, unlike the multicasting in MANETs and WSNs in which route recovery and energy concerns were important issues, in MCMR WMNs throughput and interference are the main concerns. Finally, since there exist several radios on each node which can dynamically be tuned on various channels, multicasting in MCMR WMNs has more difficulties.

Multicasting problem in single-channel single-radio WMNs has been investigated in [18, 19, 22-27, 45]. The work in [28], addresses the multicasting in multi-channel single-radio settings, which basically describes different network configurations; also, it highlights the throughput maximization in which the multicast tree is not constructed, in contrary to the interesting problem of this paper which is joint channel assignment and minimal interference multicast tree construction.

With regard to the various applications of multicast routing, a few works have specifically been aimed at MCMR WMNs. Trend of the existing researches is based on sequentially solving two sub-problems channel assignment and multicast routing [31, 32, 36-38 and 40]. As also discussed in [33-35 and 39], the solution for either of these two sub-problems is assumed to be already computed and considered as input to the other.

Joint multicast routing and channel assignment problem in MCMR WMN has been proved to be NP [31-38]. Recently, learning automata [48,49] have been applied to a vast variety of science and engineering applications implying its aptitude to solve different kind of NP problems (e.g., [41,42,50]). This is because of inherent characteristics of learning automata that makes its application in networks as a reasonable control tool. For instance, it does not require prior knowledge about traffic characteristic [52], and thus it can be utilized online in different networks. In addition, learning automata doesn't not require to complex analyze of network during learning phase. Moreover, each learning automaton needs to keep just one action probability vector which exhibits its less memory demands. Accordingly, in this paper we adopt learning automata to present an integrated scheme, named LAMR (Learning Automata based Multicast Routing protocol) for solving two aforementioned sub-problems conjointly.

In our distributed solution, the learning automata residing on interfaces of each node specify using which channel, their associated radios should communicate with the neighbors. Clearly, the proposed design is composed of two phases; in the first phase, the minimal end-to-end delay paths between the multicast source to every multicast receiver are constructed in parallel and hence, speeds up to present the multicast solution. In the second phase, the residing learning automata refine the initial tree to achieve a minimal interference tree. LAMR also accounts for the *hidden channel problem* [40], which typically occurs when two-hop away nodes attempt to tune on the same channel. Experimental results demonstrate that LAMR outperforms the LCA and two versions of MCM proposed by Zeng et al. [31, 32]

as well as the genetic algorithm-, tabu search-, and simulated annealing-based methods by Cheng et al. [36-38] in terms of attained throughput, end-to-end delay, average packet delivery ratio, and multicast tree total cost.

The rest of this paper is organized as follows: In section 2, we survey the prior art multicast methods in MCMR WMNs and would highlight their advantages as well as the associated performance issues. The theory of learning automata is presented in section 3. Our proposed design for the joint channel assignment and multicast routing problem will be presented in section 4. In section 5, we will present the outcome of several performance measurement studies. A brief discussion comes in section 6. Section 7 concludes the article.

2. Related works

In [51], authors have utilized learning automata to solve multicast routing problem in MANET. As it is discussed in Introduction, the characteristics of MANET are very different from those of MCMR WMNs; in opposite to MANET, the nodes in MCMR WMNs are stationary, and hence there is no need to predict the nodes' trajectory in routing protocols. Another difference of MANET with the network of interest in this article is that, here energy is not constraint anymore and instead every node is equipped with several radio interfaces which can be tuned to the available channels. Therefore, unlike MANET multicast routing in MCMR WMNs is akin to channel assignment problem which brings about its own complications. As for similarity items, in [51] every node is equipped with just one learning automaton which its action is to select the next relay node on the multicast tree. In our solution, every node is equipped with several radios for each one a learning automaton is resided. The learning automata residing on interfaces of each node specify using which channel, their associated radios should communicate with the neighbors.

In [30], multicast throughput optimization in MCMR WMNs is modeled in terms of an integer linear programming (ILP) formulation. However, given that the pure LP model is not scalable to the size of the network, LP relaxation has been used along with two throughput optimization schemes. In the first scheme, channels are initially assigned to radios in a greedy manner via breadth-first search (BFS). The resultant associations are then given as input to the LP model for throughput computation. The second scheme, on the other hand, essentially is an iterative variant of the former; i.e. the channel-radio associations as obtained by the BFS are corrected by the LP model in each iteration. Unlike the case in our design, multicast tree construction does not form the mainstay of [30].

In our brief review of the relevant literature, we specifically focus on studies addressing both channel assignment and multicast tree construction in MCMR WMNs. Existing schemes may be roughly classified into the following three categories:

- Methods taking channel assignment for granted, thus treating multicast tree construction as the main problem [39].
- Methods assuming a given multicast tree, thus solving for channel assignment as the core problem [33-35].
- Methods sequentially solving for both multicast tree construction and channel assignment [31, 32, 36-38 and 40].

In [31] and its extended version in [32], two methods for multicast tree construction and channel assignment in MCMR WMNs have been proposed. In the first method, mesh nodes are initially visited by conducting a BFS starting from the multicast source. This way, nodes are placed at different levels from source to multicast group members. Multicast tree is actually constructed during the next step initialized with the sender node and all receiver nodes of the multicast group. Forwarding nodes in the multicast tree, on the other hand, are specified by taking on a bottom-up approach: if each receiver node v has several parents and one of their parents is on the multicast tree, this receiver is connected to that parent (f_v). Otherwise, one of the parent nodes is selected randomly and one link is established to that parent (f_v). The algorithm for node f_v would continue recursively. After constructing the multicast tree, a so-called 'LCA' (Level Channel Assignment) algorithm is used which assigns channels to nodes depending on to what level of BFS traversal tree they belong. In particular, channel i is assigned to nodes located at level i of the tree. While the main advantage of this scheme lies in its simplicity, it also comes with the following shortcomings; when presented with multiple candidates, a multicast receiver has to be content with a parent chosen at random, which may not always work out to be the most promising choice. Also, there is no discussion as to which receiver node should initiate tree construction, which is despite the implications it might have on the specifics of the resultant tree structure and on the total number of packet copies communicated throughout a session. These can be counted as the type of oversights which render the communication substrate prone to interference, hence proving inefficient in terms of throughput. As for channel assignment, the nodes placed at the same level are prone to interference. Also, if the number of channels is more than the number of levels, the pool of available channels will be left underutilized.

The second method, namely MCM (Multi-Channel Multicast) also starts with placing the tree nodes at different levels using BFS. It then seeks to omit the edges which happen to be between the nodes at the same level. The minimum number of relay nodes (RNs), which form the multicast tree, would be determined according to the following approximation algorithm:

1. Parents can be chosen as RN if one of their children has a fewer number of parents.
2. When presented with a number of RNs, the node with the largest number of children is selected.
3. The elected RN together with its children is removed from the tree and the previous two steps would be repeated until all nodes at level $i+1$ are removed.

Channel assignment for MCM, on the other hand, can be performed by either of the following two algorithms reported in [32]: The top-down 'Ascending Channel Assignment'(ACA) assigns channels to levels starting from zero. When running out of channels, the algorithm re-assigns channel zero to the nodes of the next level and this process repeats. The shortcomings associated with ACA are as follows; ACA is prone to *Hidden channel problem* [40]. Additionally, for almost diagonal trees, ACA is subject to the same shortcomings as those of LCA's [32].

The second channel assignment scheme, referred to as 'Heuristic Channel Assignment' (HCA) draws on the channel separation concept which indicates the disparity between two channel numbers. For example, the separation between channels 2 and 5 is 3. In this method when a channel is assigned to node u, it should minimize the sum of squares of interference factor [47] between node u and all nodes v in its neighborhood. Probably, the most significant advantage associated with this initiative lies in the factoring of throughput and delay into the evaluations. However, since it only accounts for single-hop neighbors, the *hidden channel problem* may occur. In this reference, authors also explained the case that there are more interfaces per each node. In this configuration, MCM algorithm is applied to construct the multicast tree. As for the channel assignment strategy, for a given network with $2k$ interfaces and C channels, it is assumed initially there exist just 2 interfaces. Then, aforementioned channel assignment algorithm for MCM is applied by restricting the number of available channels to $\frac{C}{k}$. After the process, supposing the assigned channel to receiving and sending interfaces are i and j respectively, interface p, (where $1 \leq p \leq k$) is assigned to channel $i + \frac{C(p-1)}{k}$, and interface q (where $k + 1 \leq q \leq 2k$) is assigned to channel $j + \frac{C(q-k-1)}{k}$.

The *hidden channel problem* of [32] has motivated the work in [33] extended later in [34]. In order to reduce interference, a rational function has been proposed which evaluations of the assignment of channel c to node v. The *hidden channel problem* is dealt with by factoring the channel information of nodes within two hops into the objective function. However, it is assumed in [33 and 34] that the multicast tree is readily available.

The evaluation function technique is also utilized in [35]; in particular, the assignment of channel c to node v is evaluated based on the probability of packet transmission by neighboring nodes of v on channel c. However, the specifics of the computation of this probability are not elaborated in [35]. It also works under the assumption that the multicast tree is already constructed.

A genetic algorithm-based scheme has been proposed in [36 and 37], extended also in [38]. Here, each multicast tree is represented by a two dimensional array (chromosome) the rows of which determines a path from multicast source *S* to the receiver R_i. Basically, a chromosome has *K* rows for *K* receivers. The number of columns in the chromosome is equal to the maximum path length from *S* to the multicast group members. The representation of the chromosomes is based on the IDs of the nodes lying along the path from sender *S* to the multicast group members. In this method, chromosomes are created as follows: The algorithm begins from the multicast source node *S*. A one-hop neighbor of *S* is randomly selected and its ID is inserted into the chromosome. This process continues till it reaches a receiver R_i. Thus, a row of chromosomes is made. The same process also runs for the next receiver (within the next row). After constructing the multicast tree for each row of the chromosome, which is essentially a path from *S* to R_i, the channel (3 is the number of available channels) is sequentially assigned to the edges. The initial value of *n* is zero and it is incremented by one after passing each edge. The following disadvantages can be pointed out for [36, 37, and 38]; The assignment of the channels to each row of the chromosome is subject to similar shortcomings as those of the LCA algorithm's; i.e. the *hidden channel problem* is not accounted for. Further, If the number of channels is more than the number of multicast tree levels, some channels will not be used at all. Also, the overhead induced from the chromosome construction together with the GA's cross-over and mutation operator scan be significant.

The methods proposed in [31-38] either construct a multicast tree or have assumed the tree is already constructed and then channels are assigned. In [39], however, the channel assignment is supposed to be done a priori and instead, the multicast tree should be constructed. More specifically, a centralized multicast tree algorithm, namely 'Minimum Number of Transmission' (MCMNT), has been proposed which aims to minimize the number of packets copied on to different channels in each node. The objective is to construct a tree with minimum cost using either the Dijkstra or the Bellman_Ford algorithm.

In [40], a distributed bottom-up approach has been proposed to construct a multicast tree and to establish channel-radio associations. Initially, an approximation algorithm is used to generate the minimum RN set and to construct the multicast tree:

1. Each node identifies its two-hop neighbors with the least number of parents.
2. Some of the parents of two-hop neighbors with the fewer children are candidate. Candidates with the best link quality will be selected.
3. The selected nodes and all of their children are removed from the single-hop and two-hop nodes' list.
4. The algorithm continues until both single-hop and two-hop neighbors' list would be empty.

After determining the set of RNs, a node, willing to join the group, sends a unicast joint_REQ message to its parent. The parent node, in turn, selects a node from its parent candidate list and passes the information to its parent as well. Every node that receives joint_REQ should send joint_REP in response. Embedded within these messages is the information on channel settings. In case the node sending the joint_REQ message happens to be the first child node of its parent, the 'channel adjustment information' is equal to the fixed channel of the sender node. In this case, if the fixed channel of the sender and receiver are the same, the parent node selects another channel to prevent interference. On the other hand, if the node that sends joint_REQ is not the first child of its parent, then channel adjustment entails those channels used by the majority of the children of the current node. Over the course of the subsequent step, joint_RPL is sent to the requesting node. Upon receiving the joint_RPL message, the node switches its fixed channel according to the channel adjustment information and keeps this channel fixed during the multicast session. This method is, however, subject to *hidden channel*.

Recently other successful researches on multicast routing problem have been reported which their underling networks are basically different of MCMR WMNs (e.g., [41-43]).

In sum, the major drawback with the existing methods is that they have considered channel assignment and multicast tree construction in the form of two independent issues and thus have taken on an essentially sequential approach to a joint problem [31, 36-38 and 40]. In [33-35 and 39], it is assumed that the solution for either of these two sub-problems is pre-calculated. This is while cross-layer design forms an integral part of a successful WMN-based implementation, as has been extensively and methodically argued in [27]. Moreover, all the reviewed schemes, except of course for [40], are built around a centralized perspective. Considering aforementioned shortcomings, in what follows we present learning automata which its concept is essential to present proposed multicast routing protocol named LAMR.

3- Theory of Learning Automata

In this section Learning Automata [48,49] will be briefly reviewed.

A learning automaton is an adaptive decision-making entity that enhances its functionality trough learning how to choose the optimal action from a finite set of permitted action by repeated interactions with a random environment. In this case, the environment evaluates this taken action and responds by a reinforcement signal. The learning automaton then updates its internal information according to both its chosen action and then received reinforcement signal. Subsequently, the learning automaton adjusts its action repeatedly until a termination condition is satisfied. Every environment is represented by $E=\{\alpha,\beta,c\}$, where $\alpha=\{\alpha_1,\alpha_2,...,\alpha_r\}$ is a set of inputs, $\beta=\{\beta_1,\beta_2,...,\beta_r\}$ is a set of outputs, and $c=\{c_1,c_2,...,c_r\}$ is a set of penalty probabilities. The environments depending on the nature of the reinforcement signal β can be classified into P-model, Q-model, and S-model. Whenever the set β has just two members, the environment is said to be of a *p*-model. In this environment, $\beta_1=1$, $\beta_2=0$ represent the penalty and reward respectively. Similarly, the Q-model environment contains a finite number of values in the interval [0,1]. The *S*-model environment, on the other hand, has an infinite number of members. Finally, c_i denotes the penalty probability of each taken action α_i.

Learning Automaton is classified into fixed structure and variable structure. Learning automaton with variable structure is represented by $\{\alpha,\beta,p,T\}$, where $\alpha=\{\alpha_1,\alpha_2,\cdots,\alpha_r\}$ is a set of actions, $\beta=\{\beta_1,\beta_2,\cdots,\beta_r\}$ is a set of inputs, $p=\{p_1,p_2,\cdots,p_r\}$ is the action probability vector, and $p(n+1)=T[\alpha(n),\beta(n),p(n)]$ is its learning algorithm. The learning automata randomly chooses an action in accord with its probability vector (p_i) . Supposing that the chosen action is α_i, the learning automaton, upon receipt of a reinforcement signal from the environment, updates its action probability vector according to formula (1) in case that the received signal is taken as desirable and to (2) otherwise. *a* and *b* In formulae (1) and (2) denote reward and penalty parameters respectively. If *a* and *b* are set equal, the algorithm is referred to as L_{R-P} ; whereas, for $b \ll a$ the algorithm is called $L_{R\varepsilon P}$. Finally, if $b=0$ then the algorithm is said to be L_{R-I}.

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

$$\begin{aligned} p_i(n+1) &= (1-b).p_i(n) \\ p_j(n+1) &= \frac{b}{r-1}+(1-b)p_j(n) \qquad \forall j \quad j\neq i \end{aligned} \tag{2}$$

Recently, learning automata has been used as a powerful tool to solve multicast problem in different networks [41-43,51].

4. Proposed method

Lets $\mathrm{G}=(\mathrm{V},\mathrm{E})$ be isomorphic to MCMR WMN. Node $\mathrm{s}\in\mathrm{V}$ is going to start a multicast session to the receiver set $R=\{R_1,R_2,R_3,\dots,R_r\}$, where $R_i\in V$. The problem is how to find a minimal interference multicast tree. The required assumptions and preliminaries followed by the specifications of the proposed scheme are presented in next two sub-sections.

4.1 Assumptions and Preliminaries

Our study in this paper is based on some reasonable assumptions as follows. As mentioned in Introduction, each mesh node in MCMR WMN is assumed to be equipped with multiple radio interfaces not more than the number of available channels. All mesh routers are supposed to have omni-directional antennas with identical transmission / interference ranges. $P(v)$ denotes a set of one-hop downlink neighbors which have a potential to be actual neighbor of v. The members of $P(v)$ can be an actual neighbor if at least a common channel is assigned to their radios. The interference of Link l, referred to as $I(l)$, is measured by the sum of all one- and two-hop away contentions. $I(T)$ corresponds to the overall interference in multicast tree and equals to sum of all $I(l), l \in L$, where L is the set of all links forming the multicast tree.

As elaborated in [32] and [38], because of the broadcast nature of wireless nodes, all the links initiating from the same node are assigned to the same channel. That is, a single transmission by a node can be received by all other nodes within its transmission range. We make this point clear in our study where all the links initiating from the same interface, are assigned to the same channel.

On the other hand, as discussed in [54], to prevent the *ripple effect* of channel changing in MCMR WMNs, the network interfaces cards of the mesh nodes should be divided into Down-NIC (DNIC) and Up-NIC (UNIC) while each node is only responsible to assign channels to its DNIC. Accordingly, in our model, the learning automaton residing at the *i*-th DNIC of each mesh node *j*, referred to as LA_{ij}, decides on which channel, the DNIC R_{ij} should send data to UNIC R_{in} , with *n* being a member of the node *j*'s one-hop downlink neighbors (see Fig. 1). That is, associated with each LA is an action probability vector (APV) $\Gamma_{1\times c}$, where c denotes the number of available channels. Each element γ_{1c} is initialized with $\frac{1}{c}$ and represents the probability of picking the action α_{1c} , i.e. assigning channel *c* for connection to one-hop neighboring nodes.

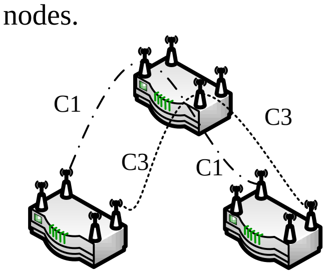

Figure 1: A MCMR WMN with 2 UNICs and 2 DNICs per each node

The aim of each LA is to minimize the two-hop channel correlation coefficient for its associated interface which, in turn, depends on the inter-channel separation. The correlation coefficient between two channels is known for all possible channel separations [28]. For example, for the IEEE 802.11, the correlation coefficient between channels 1 and 2 equals to 0.7906.

In LAMR, a customized control message is used that Table 1 shows its structure. The field *Type* specifies what type the message is. In our protocol, three types of control messages have been used as follows; *Route_Request* which is relayed across the network to reach multicast receivers. Hereafter, we refer to this message as *RReq*. *Route_Reply* which is sent back by multicast receiver nodes. Hereafter, this message is referred to as *RRep*. Finally, *Reward* which is sent by multicast source to all the nodes forming the new constructed path.

Table 1: Data structure of the customized control messages

Type	*Enumerative field that specifies the type of message; 0 for RReq, 1 for RRep, and 2 for Reward.*
MS	*Determines ID of the multicast source*
MRList	*Array-structured field contained all multicast receivers' ID*
DR	*Specifies the ID of discovered multicast receiver*
TTL	*Time-To-Live of the message.*
IdList	*Array-structured field to store the nodes ID during the path discovery*
ChannelList	*Array-structured field to register the selected channels in each node*

4.2 LAMR: Learning Automata-based Multicast Routing Protocol

Selecting the relay nodes forming the multicast routing tree is known as NP. Therefore, in this section we utilize learning automata to present LAMR as a novel multicast routing protocol for MCMR WMN. LAMR protocol is run on different NICs independently in parallel. The operation of LAMR for each NIC is composed of two phases; in the first phase, the minimal end-to-end delay multicast tree is constructed and in the second phase the initial tree is optimized from overall interference point of view. In what follows, both phases are discussed.

A. First Phase

The operation of the first phase is organized into several steps which described as follows;
Initially, the learning automata residing at the DNICs of multicast source node which is indexed by *j*, are activated in parallel. The activated learning automaton LA_{ij}, picks an action according to its APV. As mentioned in sub-section 4.1, the chosen action specifies that using which channel, R_{ij} should communicate with UNIC R_{im}, where $m \in P(j)$. Afterwards, a *RReq* message is generated by multicast source and then the chosen channel by LA_{ij} and index of the associated node are inserted in fields *ChannelList* and *IDList* of the message, respectively. The generated message is then sent by R_{ij} to all R_{im} . This process is done by all learning automata residing on all DNICs of multicast source node in parallel.

The LA_{im} residing on *i*-th DNIC of every intermediate nodes *m* is activated upon receiving *RReq* message at *i*-th UNIC R_{im}. Then it chooses one of its actions according to its APV. The chosen channel and ID of the corresponding node, say *m*, are appended to fields *ChannelList* and *IDList*, respectively. Also the value of *TTL* field of the message is decreased by one and finally the incoming message is sent to all radios R_{ik}, where $k \in P(m)$.

The *RReq* messages are relayed across the *i*-th Up and Down NICs of network nodes until reach one of multicast receivers. Upon receiving this message in a multicast receiver, the receiver performs two tasks; First, it acts like an intermediate node. That is, it decreases the *TTL* of the arriving message and relays it to its one-hop neighbors. The reason is that maybe there exist another receiver near that. In this case, many hops from source to these two receivers may be common and hence, the extra duplication of packets is avoided. The second task is that the receiver node duplicates the message and changes its type to *RRep*. Also, its *ID* as a discovered receiver node is inserted into field *DR* of duplicated message and then duplicated message is sent to *i*-th UNIC of multicast source using dijkstra algorithm.

As its feedback mechanism, upon receiving the *RRep* message on *i*-th UNIC, the multicast source examines the *DR* field of the arriving message. Incoming message is discarded in case the *RRep* message has been previously received from the receiver node whose *ID* is inserted into field *DR*. The reason is that round trip time of the previously *RReq* message is less than recently arriving message which implies less end-to-end delay. Otherwise, the multicast source sends a *Reward* message to all of the nodes whose IDs have been stored in field *IDList* of the *RRep* message which form a new discovered path from the source to the receiver.

Intermediate nodes update their routing tables as well as their APVs according to the learning rule L_{RI} upon receiving the *Reward* message. In this step, the radio-channel association using the field *ChannelList* of the *Reward* message (i.e. the channels previously chosen by the LAs) is tuned too.

Multicast source then waits *t* time steps to receive *RRep* messages from other multicast receivers on its *i*-th UNIC.

Once the multicast source receives a *RRep* message from all multicast receivers on all its k UNICs, where 2k is the whole number of NICs per each node, above process is terminated. In this case, the initial multicast tree has been constructed and the second phase starts.

B. Second phase

The minimal-end-to-end-delay multicast tree constructed in the first phase would be optimized with respect to inter-channel contention. The operation of this phase organized into a number of rounds. In continue, specification of one round is presented.

At first, the messages *RReq* are sent by the multicast source on all DNICs across the network using the routing tables constructed in the first phase. In each node, the residing LA on the each DNIC makes a decision about the proper channel-radio association to communicate with corresponding UNIC of its downlink neighbors. The *RReq* messages are relayed until reach multicast receivers. The receiver nodes then send back the *RRep* to associated UNIC of the node that sends the *RReq* message to it. The *RRep* messages are sent back until reach multicast source. It should be noted that this path is minimal-delay path constructed in the first phase. Along the reverse path to source, the radio of nodes is tuned to chosen channels by LAs. Once a *RRep* message is returned by all the receivers on all *k* UNICs of source node, the newly constructed tree is evaluated in terms of total tree contention. If the contention is less than that of before initiation of *RReq* by source, it sends a *Reward* message to all of the nodes forming the new tree. Then, intermediate nodes update their routing tables as well as APVs according to learning rule L_{RI} upon receiving this message. It should be noted that in case the resultant interference aggravates, chosen actions by LAs is "penalized" by getting no reward.

The above algorithm iteratively runs until it provides a stable multicast tree.

4.3 Reliability in LAMR

It may occur a situation in which a path from the multicast source to one or more receivers is not discovered after *t* time steps waiting. In this situation, the probability is that the receiver is out of order. This situation can be examined by the *central limit theorem* in probability theory [49]. *Central limit theorem* asserts that if there are *n* independent random variables with the same distribution in which μ and σ denote mean and standard deviation respectively, the probability of being the $(n+1)$-th variable in the range of $\mu \mp 4\sigma$ is 0.999936657516. Availability of each multicast receiver within our protocol can effectively be checked using a similar initiative. Clearly, to cope with the mentioned problem in proposed scheme, using the BFS algorithm, mesh nodes are placed on different levels. A unicast message is then sent to the nodes placed in the lowest level *n* times. Afterwards, μ and σ of their corresponding *n* round trip times (RTT) is computed. Now, if the waiting time for any receiver is greater than $\mu \mp 4\sigma$, it implies that the receiver with probability close to 1, is out of order and thus, the algorithm continues regardless of this receiver. Otherwise, it may the value of TTL of the messages is less than required. In this case, the value of TTL is increased and then the algorithm is initiated again.

The corresponding algorithms for multicast source initialization, multicast source handle message, and intermediate nodes handle message are shown in Figures 2, 3, and 4, respectively. Figures 5 and 6 depict the flow charts of the algorithms ran on multicast source and intermediate nodes.

MulticastSource_Initialize ()

1- All LAs residing on all DNICs are activated in parallel.
2- Each activated LA, chooses an action according to its APV.
3- For each DNICs a RReq message is created.
4-The ID of the multicast source and the chosen channel are inserted in the fields IDList and ChannelList of RReq message, respectively.
5- The messages are sent to associated UNIC of their one-hop downlink neighbors.
6- Detected [all interface][all receivers]=false;

Figure 2: Initialing algorithm of multicast source

MulticastSource_handlemessage (*message msg*, *interface i*, *phase p***)**

```
Switch (p)
{
Case first:                                   // Means the first phase
      if (Detected[i][msg.DR]==true)      // the msg is repetitive
      Then
            message.Discard() ;
      Else
            {
              Detected[i][msg.DR]=true;
              msg.kind=Reward;
              For all nodes n in msg.NodeList
                  msg.Send(n, i);         // arriving msg is sent to i-th UNIC of mesh node n
            }
            For x from all interfaces
                For r from all receivers
                      If (Detected[x][r]==true)
                                  Then
                                        {
                                          P=second;            // the second phase should be started
                                          Detected [all interface][all receivers]=false;
                                          Return;
                                        }
            Wait (t);
            receiver. alive = CLT() ;   /*Check the availability of un-discovered receiver using Central Limit Theorem.*/
            If (receiver. alive)
                   Then
                       Increment TTL and initiate a new RReq message
                   Else
                    {
                          P=second;
                          Detected [all interface][all receivers]=false;
                          Return;
                    }
Case second:
      Detected[i][msg.DR]=true;
      If (Detected[all interfaces][ all receivers]==true)
          Then
                {
                    Contention=Tree.evaluate();
                    If (Contention ≤ PContention)
                         {
                            msg.type = Reward;
                            For all nodes n forming tree
                                msg.Send(n, i);
                            Detected [all interface][all receivers]=false;
                            PContention = Contention;
                         }
                }
}
```

Figure 3: Handle message algorithm of multicast source

Relay nodes_handlemessage (*message msg*, *interface i*, *phase p***)**

```
Switch (msg.kind){
Case RReq:
        If the node is one of the multicast receivers
        Then
        {
                dmsg=msg.duplicate();
                dmsg.type= RRep;
                dmsg.DR= ID;
                dmsg.send(multicast source, i);
        }
        If (msg.TTL== expired)
        Then
                Msg.discard();
        Else
        {
                LA residing on i-th DNIC is activated;
                channel= LA.choose();
                msg.TTL--;
                msg.IDList= ID;
                msg.ChannelList=channel;
                If ( p==first)
                        For all nodes n placed in its one-hop downlink neighbors
                                msg.send(n, i);
                else
                        For all nodes n appeared in its routing table entries as next hop
                                msg.send(n, i);

        }

Case Reward:
        if (p==one)
          {
                LA.update();
                RoutingTable.update();
                i.channel=channel;        // interface i is tuned to the chosen channel by LA
         }
        Else
                LA.update();
Case RRep:
        msg.Backward();
        i.channel=channel;
}
```

Figure 4: Handle message algorithm of intermediate nodes

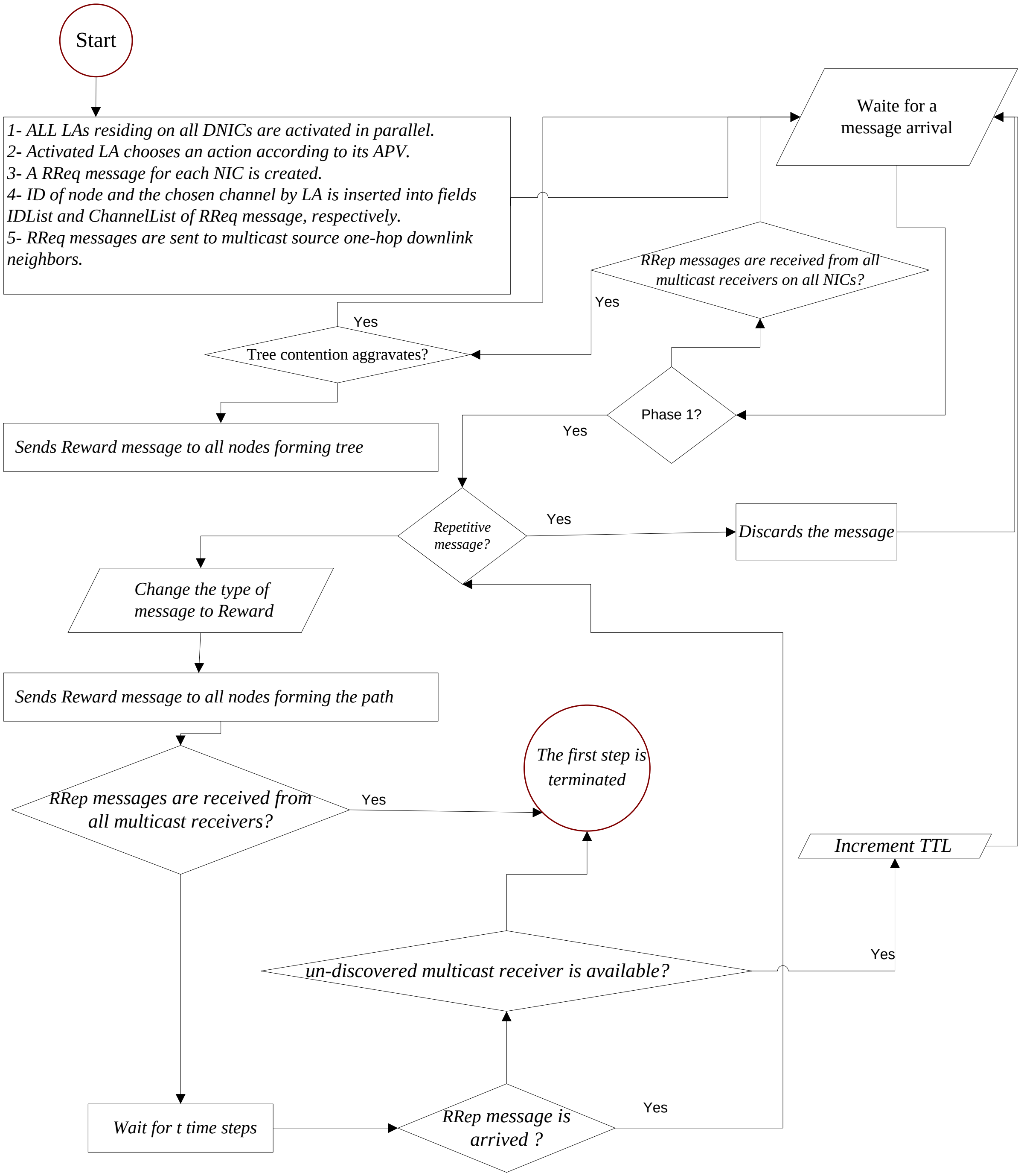

Figure 5: Flow chart of multicast source algorithm

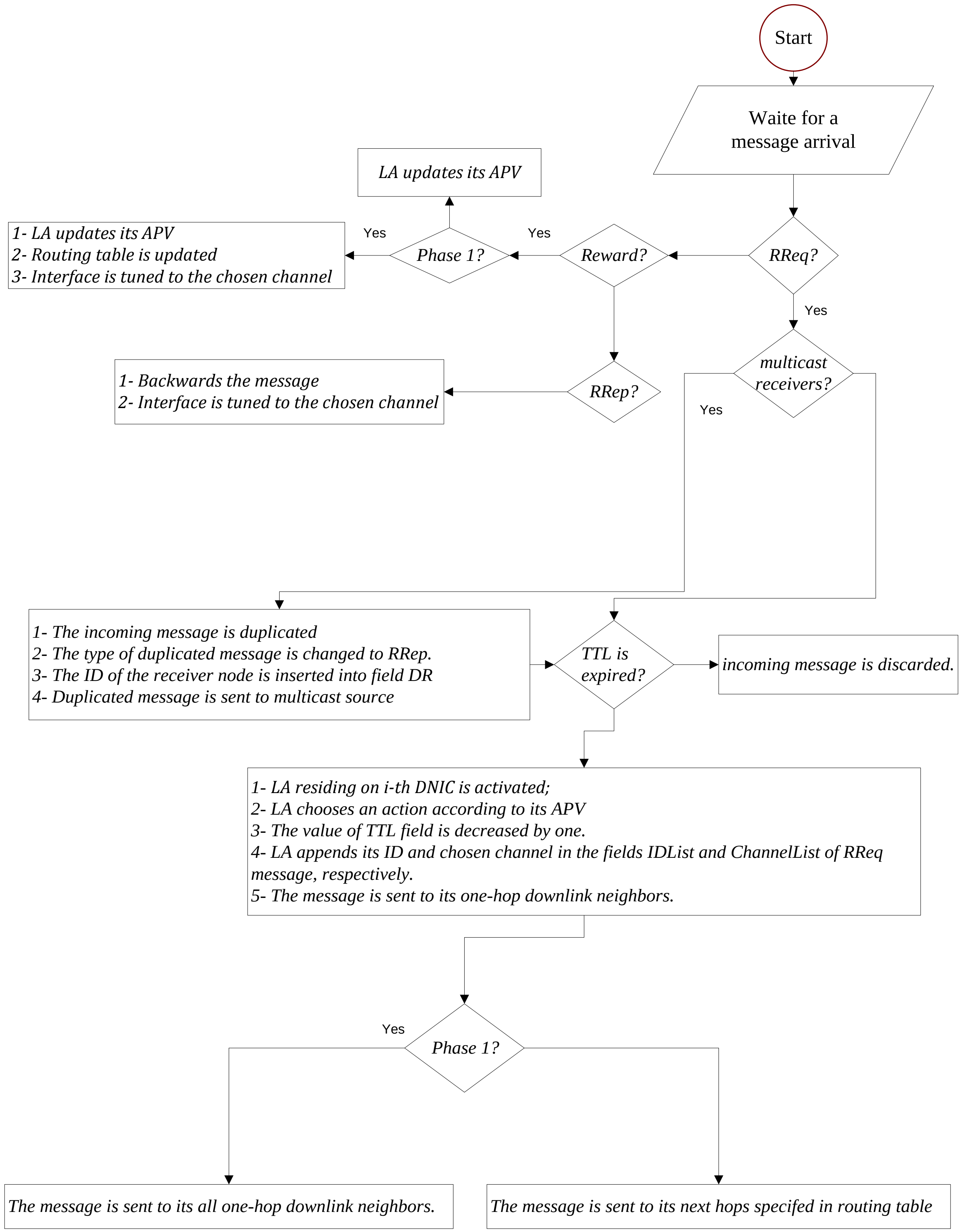

Figure 6: Flow chart of relay nodes algorithm

5. Experimental results

To study the efficiency of LAMR we have conducted several simulations in OMNET++ [56] which is a component-based, modular and open-architecture discrete event network simulation framework. A 50-node MCMR WMN was randomly generated and used in all conducted experiments. Also, each data point reported in this paper has been averaged over the results of ten different runs.

In conducted experiments, the performance of different methods is measured in terms of the end-to-end delay, average throughput, average packet delivery ratio, and total tree cost with the following definitions;

- **Average packet delivery ratio**
  Packet delivery ratio is defined as the number of packets received at multicast receivers over the number of the packets sent by the multicast source averaged on all multicast receivers. This criterion specifies the number of packets delivered to the multicast receivers over the number of packets expected to be received by the multicast receivers.
- **Average end-to-end delay**
  End-to-end delay is defined as the average time elapsed between sending the packets by the multicast source and receiving at all the multicast receivers. This criterion is averaged on all multicast receivers.
- **Average throughput**
  Throughput is defined as the number of packets received by the receiver over the required time to deliver this number of packets averaged on all multicast receivers.

  $$\text{Average throughput} = \frac{1}{|MRS| \times RT} \sum_{i=1}^{|MRS|} NRP(MR_i)$$

  Where $NRP(MR_i)$ denotes the number of received packet at i-th multicast receiver. Also $|MRS|$ specifes cardinality of multicast receiver set and RT is the required time to deliver the number of packets.
- **Total cost**
  Total cost is defined as the number of links forming the multicast routing tree.

In continue, the results of conducted experiments are presented. Experiments 1 to 4 are essentially conduced to assess the efficiency of LAMR compared with LCA and two versions of MCM [31,32] as well as genetic algorithm-, simulated annealing-, and tabu search-based methods [36-38] in terms of aforementioned criteria. In this experiment, each node of MCMR WMN is equipped with just two radio interfaces. It is noted that the simulations configuration for the above comparable methods is tuned according to proffered parameter setting in their associated references.

In experiment 5, we study the achieved throughput while LAMR is utilized in 4-radio configuration network compared with 2-radio. The rest of the experiments are dedicated to evaluate the efficiency of LAMR compared with MCM [32] in terms of above criteria.

Experiment 1

To assess the average end-to-end delay for different multicast routing protocols, we have conducted three sets of simulations related to 4, 10, and 15 multicast receiver nodes in MCMR WMN. In each set, the number of channels is set to 4, 7, 10, and 13. From the results reported in Figures 7 to 9, utilizing LAMR leads to the minimum end-to-end delay of packets. The reason is that during in phase 1 of the proposed method the path with minimum round trip time has been considered. As shown in Figure 7, the end-to-end delays in both versions of MCM are the same. There is a same situation in the obtained results for simulated annealing- (SA) and tabu search- (TS) based methods. The reason is that in this experiment the number of multicast receivers is small and hence, the efficiency of different methods cannot be properly exploited. While we increase the number of receivers to 10 and 15, we found that SA-based method is more effective than TS-based method (see Figs. 8 and 9). This superiority thanks to inherent feature of SA search technique. That is, during the annealing if the new state is achieved it is accepted, otherwise it is rejected by a probability dependent on the elapsed time of annealing process not certainly.

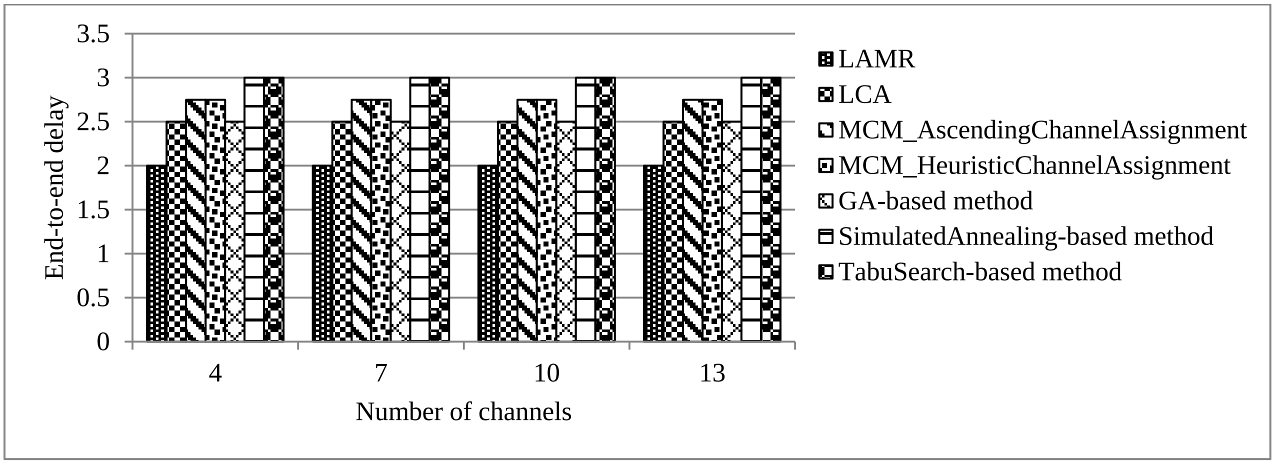

Figure 7: Average end-to-end delay as a function of channel number variations for different algorithms while the number of multicast receivers is 4

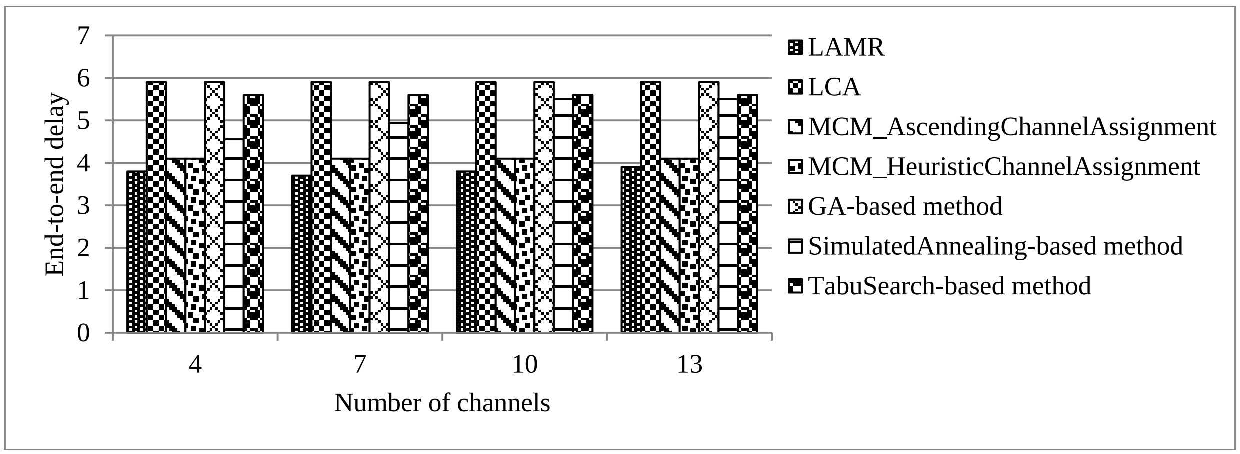

Figure 8: Average end-to-end delay as a function of channel number variations for different algorithms while the number of multicast receivers is 10

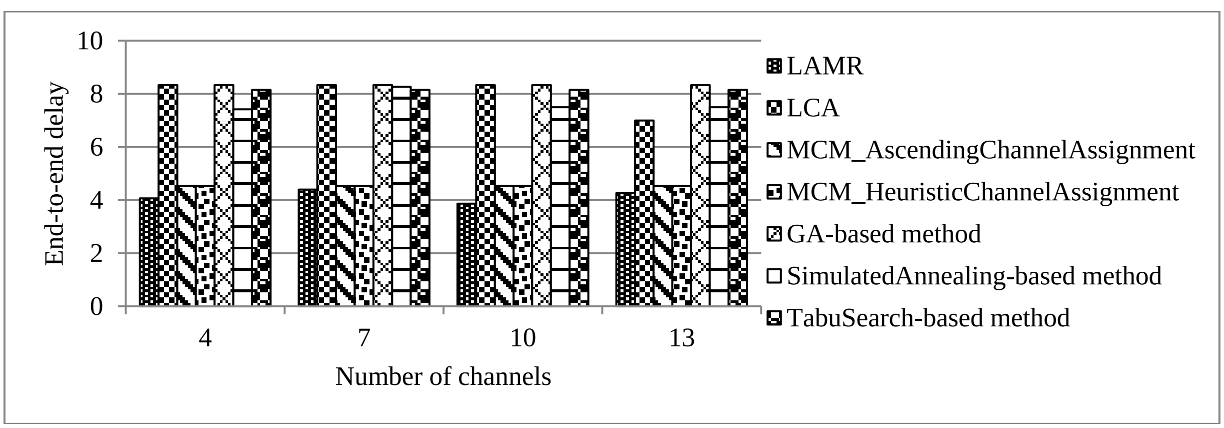

Figure 9: Average end-to-end delay as a function of channel number variations for different algorithms while the number of multicast receivers is 15

Experiment 2

This experiment was conducted to evaluate the average throughput in different methods. From the results reported in Figure 10, LAMR provides the most attained throughput compared with the other approaches. The reason is that LAMR tries to construct the multicast tree with minimal contention during the rounds. Throughput improvement in the LAMR is highlighted when the number of receivers reaches 10 (Fig. 11). As it is seen in Figure 12, when we increase the number of receivers to 15, the local contention of the links will increase and thus, the assessment of different methods will be more complicated. Nonetheless, LAMR still leads to highest throughput.

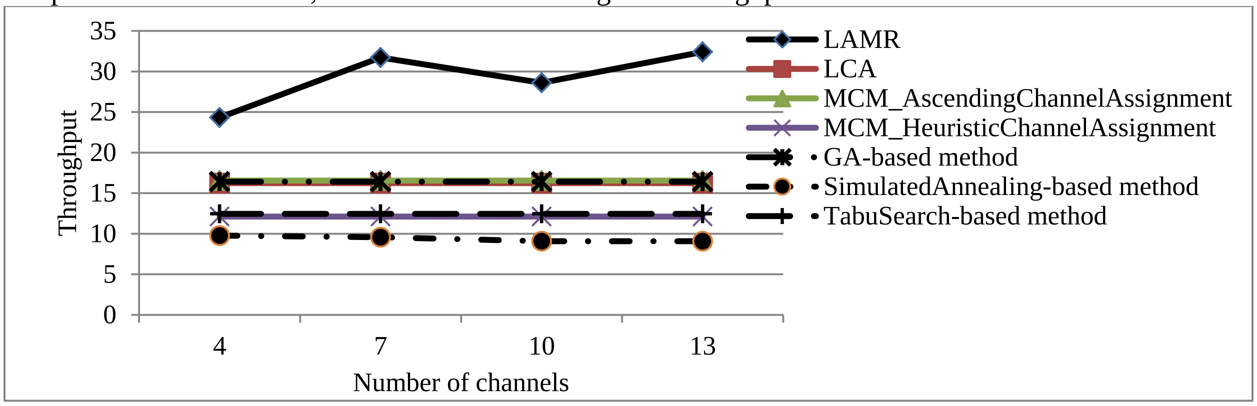

Figure 10: Average throughput as a function of channel number variations for different algorithms while the number of multicast receivers is 4

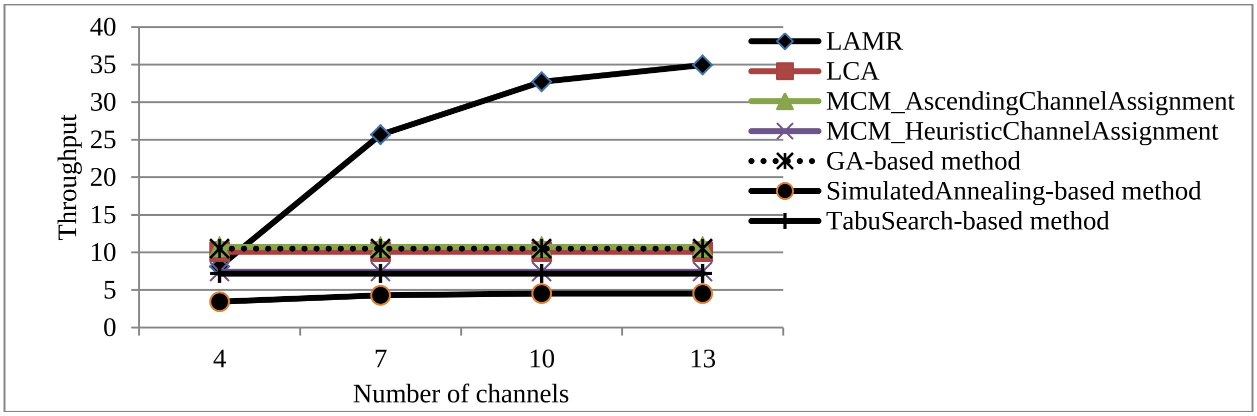

Figure 11: Average throughput as a function of channel number variations for different algorithms while the number of multicast receivers is 10

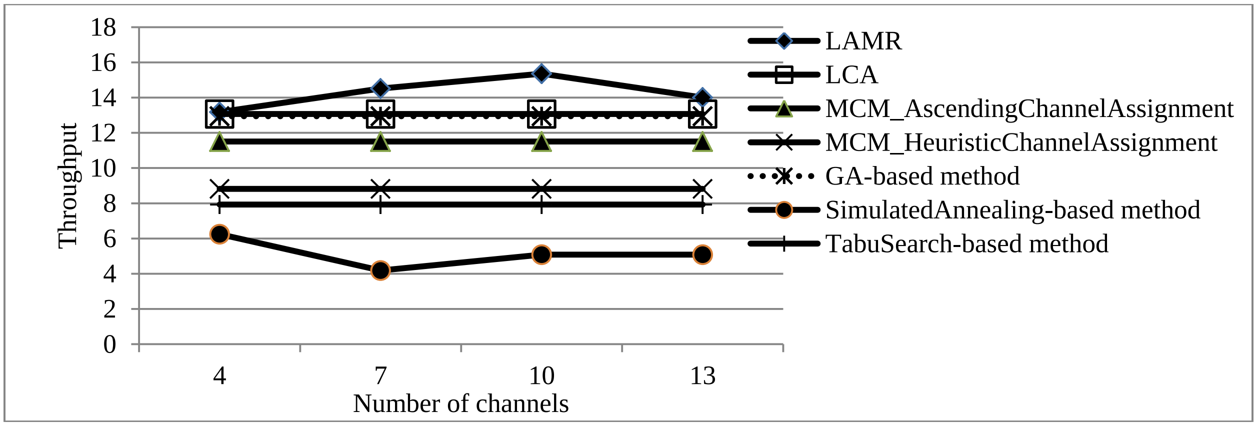

Figure 12: Average throughput as a function of channel number variations for different algorithms while the number of multicast receivers is 15

Experiment 3

This experiment is mainly conducted to evaluate the achieved average packet delivery ratio in LAMR compared with the other methods. The configurations in this experiment are the same as two prior experiments. As it is observed in Figures 13 to 15, LAMR provides the highest average packet delivery ratio among the compared schemes in all scenarios. The superiority of LAMR over the other methods is more highlighted when the number of receivers is 15 (Figure 15). The reason is that, unlike the other representative methods in which channels are assigned to the links in ascending order, LAMR effectively utilizes all available channel pool so that the minimal interference multicast tree is constructed and hence, the average packet delivery ratio rises with increasing the number of available channels.

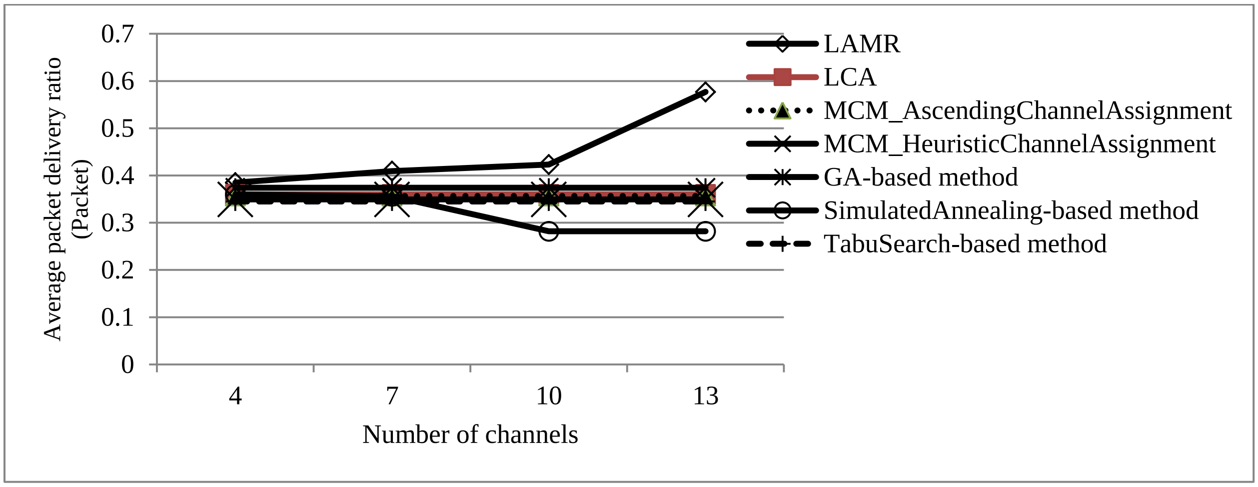

Figure 13: Average packet delivery ratio as a function of channel number variations for different algorithms while the number of multicast receivers is 4

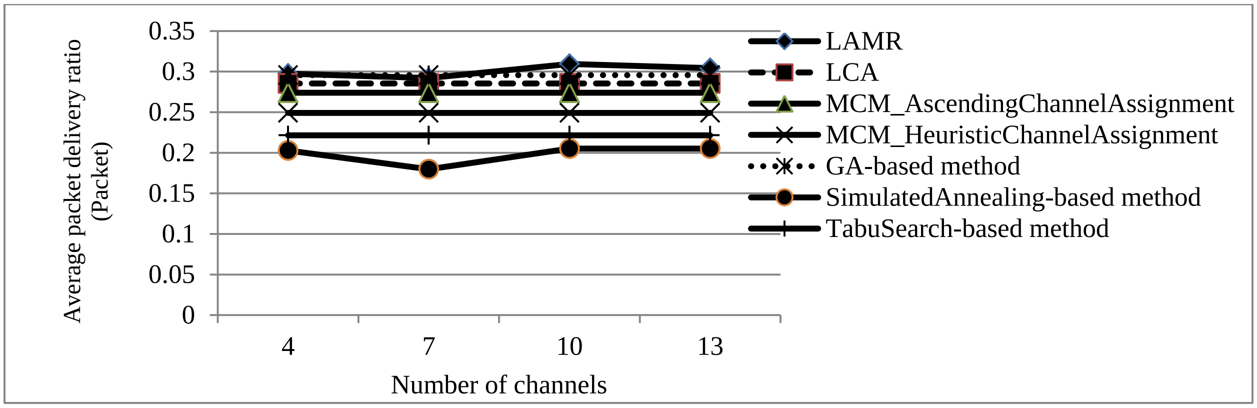

Figure 14: Average packet delivery ratio as a function of channel number variations for different algorithms while the number of multicast receivers is 10

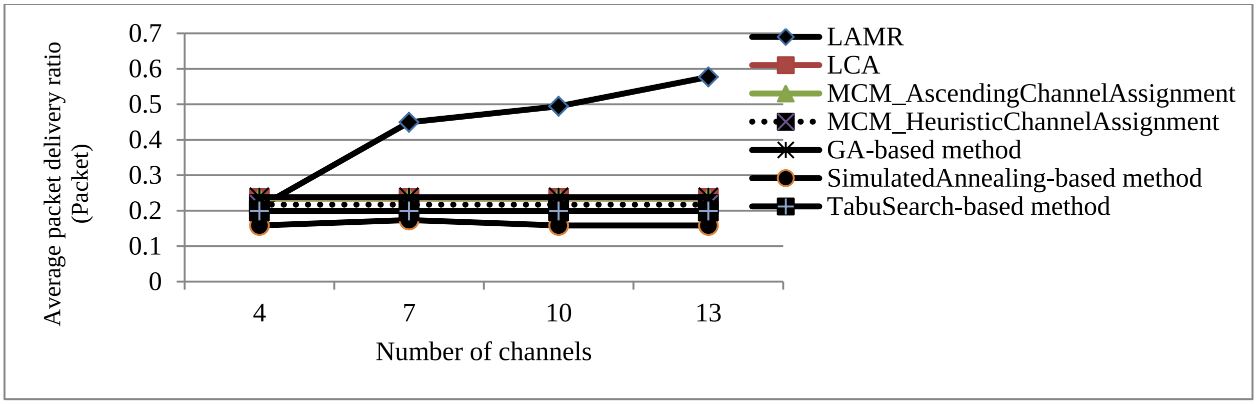

Figure 15: Average packet delivery ratio as a function of channel number variations for different algorithms while the number of multicast receivers is 15

Experiment 4

The objective of this experiment is to measure the cost of multicast tree in terms of total number of link forming the tree. From the results shown in Figure 16, the cost of tree in SA-based method is the highest among the other methods. Tree costs in the other methods are approximately the same (Figs 16 to 18). The reason is that SA-based technique might be fall in local optima.

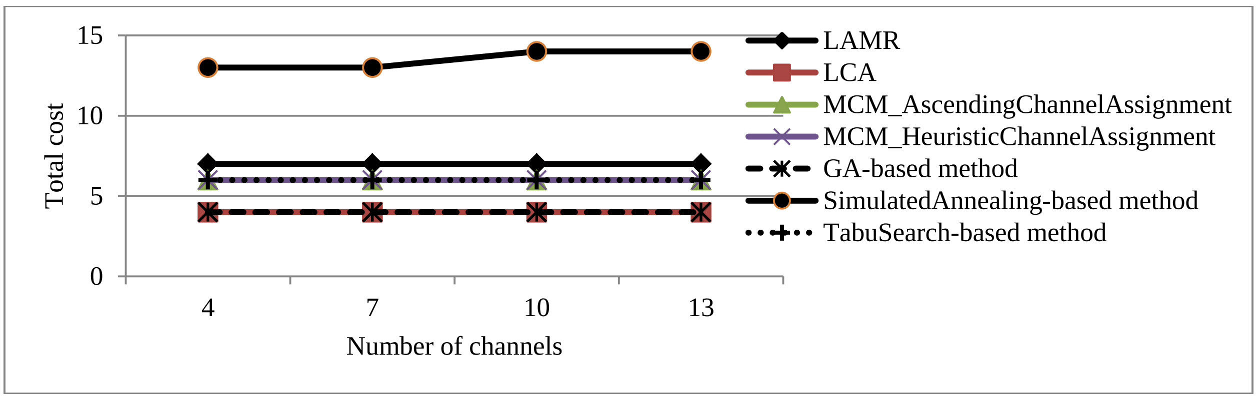

Figure 16: Total tree cost as a function of channel number variations for different algorithms while the number of multicast receivers is 4

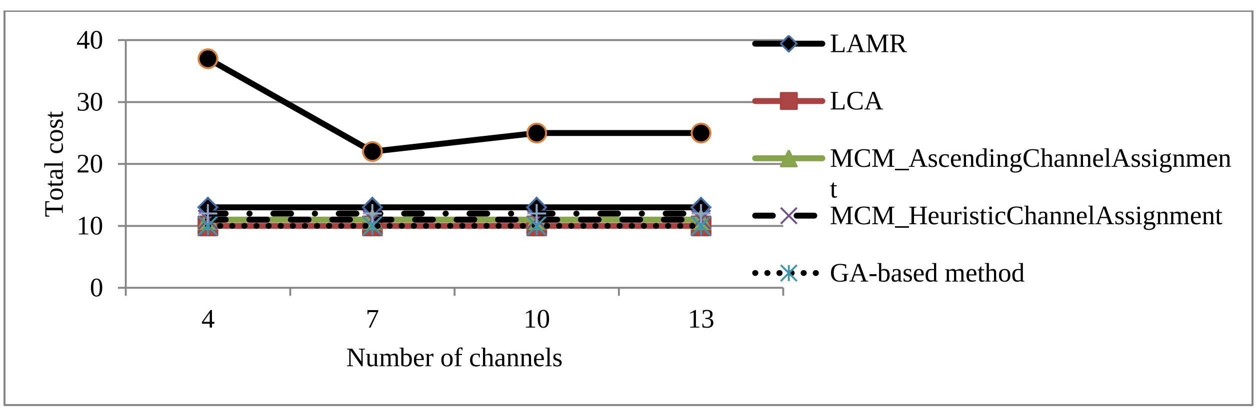

Figure 17: Total tree cost as a function of channel number variations for different algorithms while the number of multicast receivers is 10

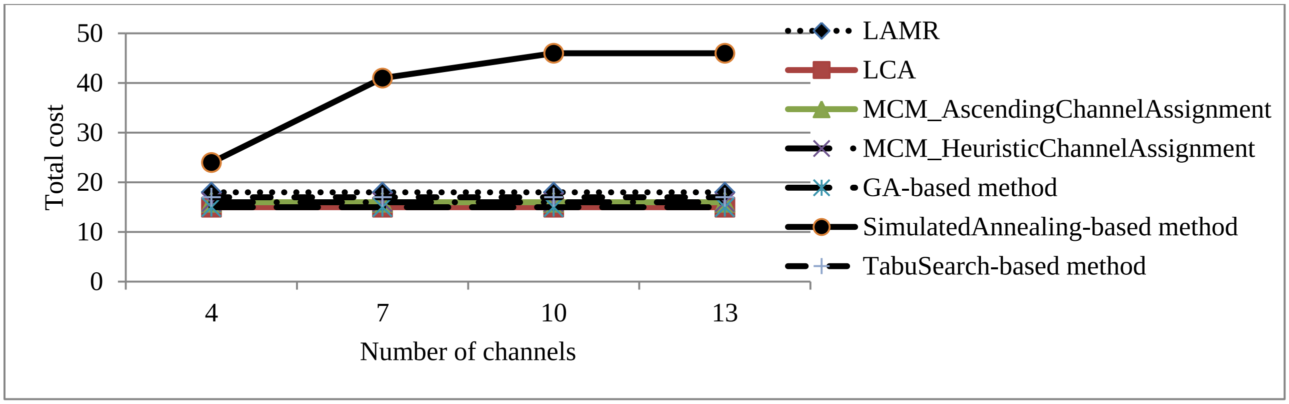

Figure 18: Total tree cost as a function of channel number variations for different algorithms while the number of multicast receivers is 15

Experiment 5

Equipping the mesh nodes with more interfaces could further improve the network throughput. As mentioned in Introduction, the number of available channels is limited and in the other hand, the number of interfaces is less than or equal to the number of channels. Currently, the number of interfaces usually is 2, 3, or 4 due to economical considerations [32]. Accordingly, in this experiment we equipped the nodes with four radios and the objective is to study the achieved throughput comparing two-radio nodes. While the members of multicast receiver set are close to each other, using more interfaces leads to more local flow interference. In this case, if the number of available channels is not enough, then the extra contention caused by extra interfaces degrades the network throughput. That is why, in Figure 19 with 4 and 10 channels, LAMR-2 works better than LAMR-4. But as it is discussed in [32] and also seen in Figure 19 when the number of channels is much enough, utilizing the more radios significantly improves the achieved throughput. This improvement is more highlighted in a network with the more multicast receivers as depicted in Figures 20 and 21 for the number of 10 and 15 receivers, respectively.

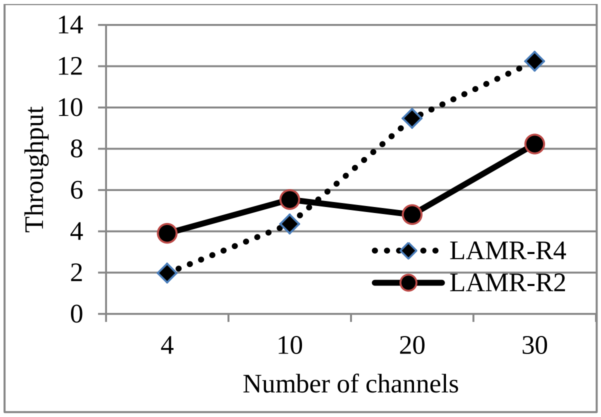

Figure 19: Average throughput in 2- and 4-radio nodes as a function of channel number variations while the number of multicast receivers is 4

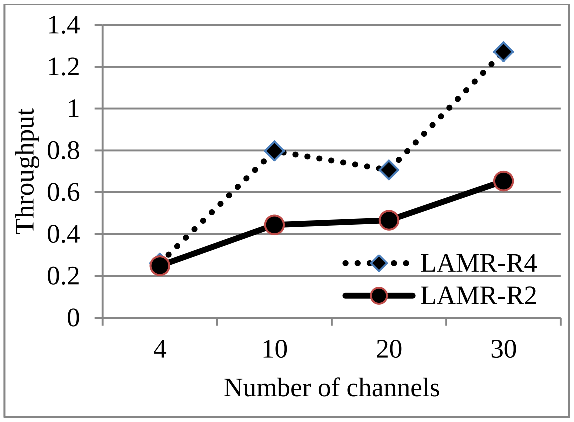

Figure 20: Average throughput in 2- and 4-radio nodes as a function of channel number variations while the number of multicast receivers is 10

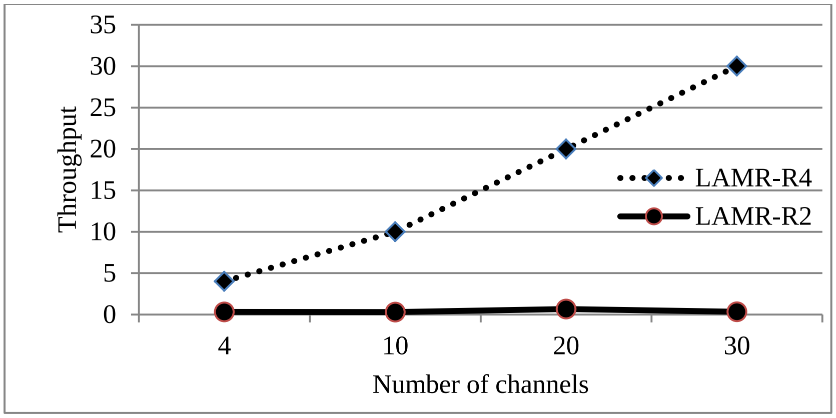

Figure 21: Average throughput in 2- and 4-radio nodes as a function of channel number variations while the number of multicast receivers is 15

Experiment 6

In this experiment, we evaluate the resultant average throughput of the LAMR compared with 4-radio version of MCM. From Figures 22 to 24, we found that LAMR significantly outperforms the MCM in terms of achieved throughput. As it is seen in these results, with increasing the number of channels the variation of achieved throughput in MCM is approximately constant. The reason is that in this scheme, given a network with $2k$-interface nodes, in each round of channel assignment the number of channels is restricted to $\frac{C}{k}$, where C and k are in turn the number of channels and radios. in other words, in MCM method channels are assigned to the radio interfaces in ascending order and it is obvious that the channel pool is not properly exploited.

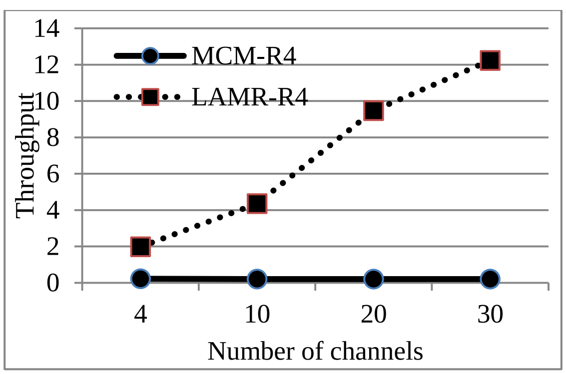

Figure 22: Average throughput in 4-radio versions of LAMR and MCM as a function of channel number variations while the number of multicast receivers is 4

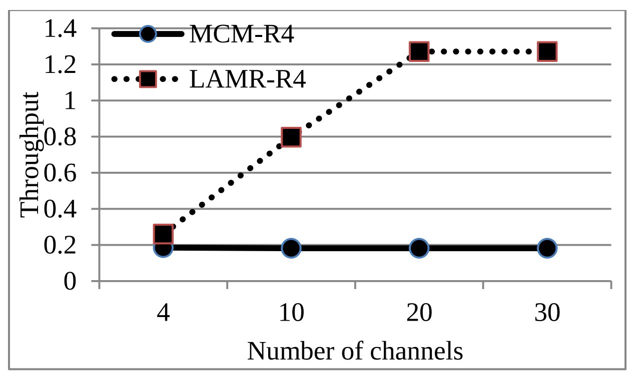

Figure 23: Average throughput in 4-radio versions of LAMR and MCM as a function of channel number variations while the number of multicast receivers is 10

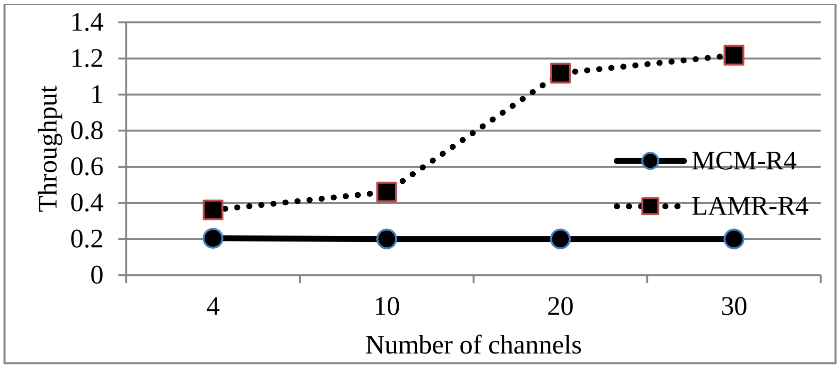

Figure 24: Average throughput in 4-radios versions of LAMR and MCM as a function of channel number variations while the number of multicast receivers is 15

Experiment 7

In this experiment we study the efficacy of the proposed solution with respect to end-to-end delay criterion while each node is equipped with 4 radios. As it is observed in Figures 25 to 27, the end-to-end delay in the proposed protocol is less than MCM. In our method route request messages are relayed in the network until they reach receivers and then route reply messages are sent back to multicast source. The first coming packets imply the paths with the least end-to-end delays. That is why the end-to-end delay in LAMR is less than that of MCM. As it is depicted in Figure 27, if the number of multicast receivers increases to 15, end-to-end delay in our method is dramatically decreased.

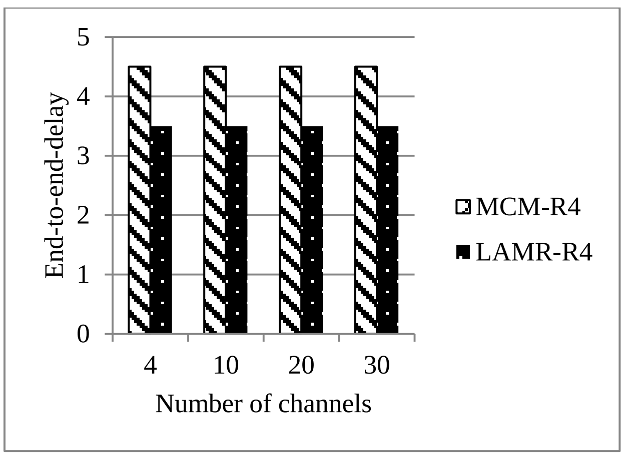

Figure 25: Average end-to-end delay in 4-radio versions of LAMR and MCM as a function of channel number variations while the number of multicast receivers is 4

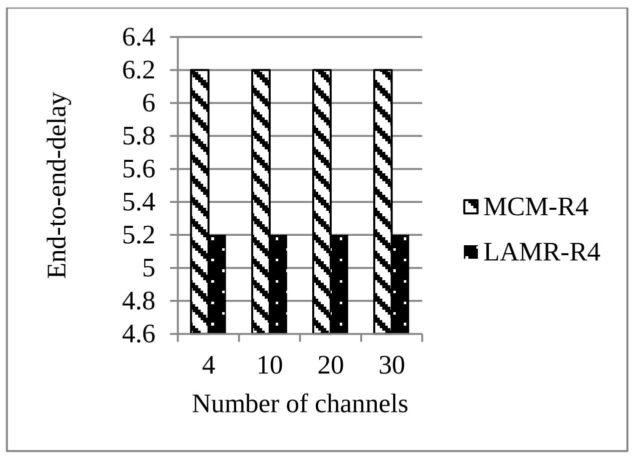

Figure 26: Average end-to-end delay in 4-radio versions of LAMR and MCM as a function of channel number variations while the number of multicast receivers is 10

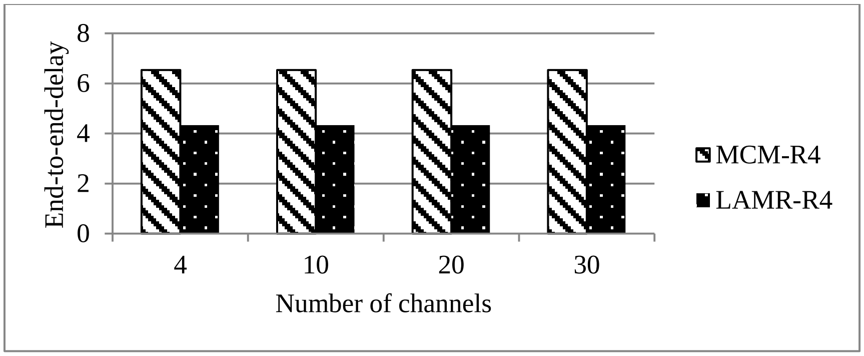

Figure 27: Average end-to-end delay in 4-radio versions of LAMR and MCM as a function of channel number variations while the number of multicast receivers is 15

Experiment 8

In this experiment, our aim is to observe the attained average packet delivery ratio in LAMR compared with MCM. From the results shown in Figure 28, it is realized that applying both methods to a 4-receiver network will result in the same average packet delivery ratio. In case of more receiver nodes (Figs. 29 and 30), the proposed method significantly outperforms the MCM in terms of average packet delivery ratio. This superiority thanks to minimal interference multicast tree construction idea of LAMR. Clearly, during the rounds LAMR tries to optimize the tree with respect to the overall interference. Less interference, more average packet delivery ratio.

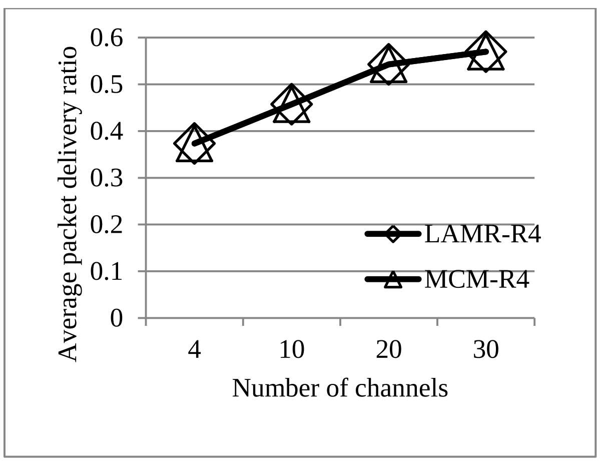

Figure 28: Average packet delivery ratio in 4-radio versions of LAMR and MCM as a function of channel number variations while the number of multicast receivers is 4

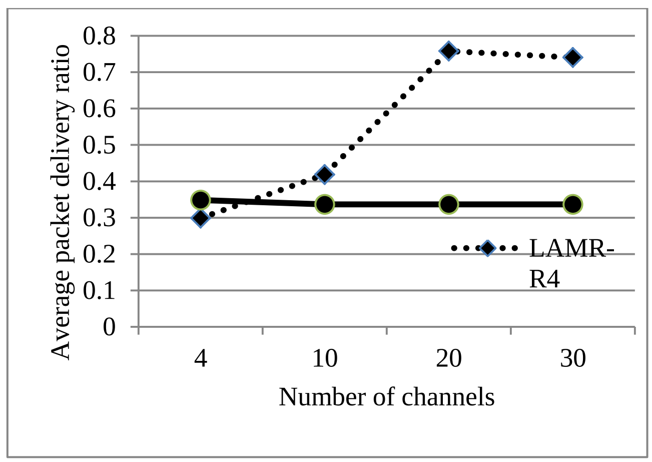

Figure 29: Average packet delivery ratio in 4-radio versions of LAMR and MCM as a function of channel number variations while the number of multicast receivers is 10

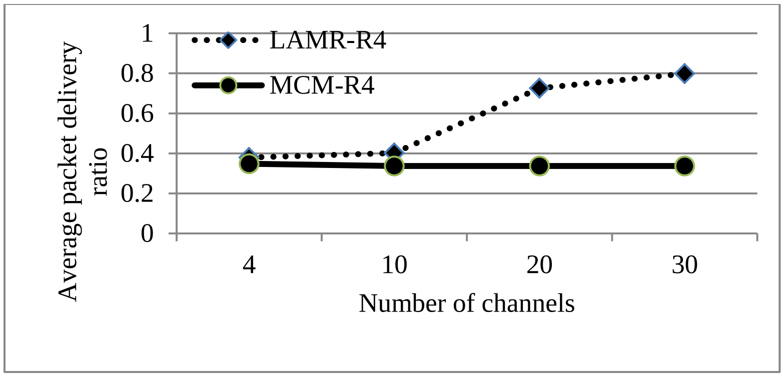

Figure 30: Average packet delivery ratio in 4-radio versions of LAMR and MCM as a function of channel number variations while the number of multicast receivers is 15

Experiment 9

MCM just takes into account the interference reduction in resultant tree. In addition to this criterion, LAMR tries to construct minimal end-to-end delay multicast tree. But both of them don't directly consider number of the links forming the tree. Nonetheless, this experiment is conducted to examine the number of links forming the multicast tree in both 4-radio version of LAMR and MCM. Obtained results in Figures 31 to 33 show that tree total costs in both methods are approximately identical.

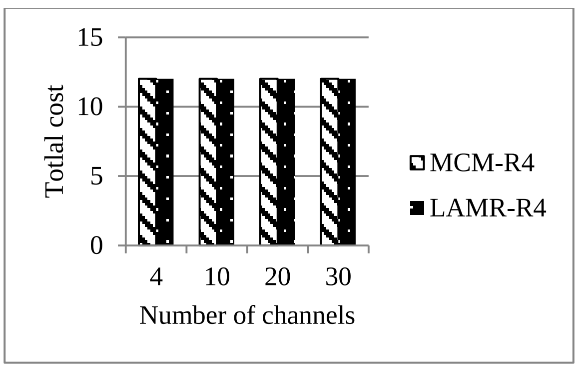

Figure 31: Total tree cost in 4-radio versions of LAMR and MCM as a function of channel number variations while the number of multicast receivers is 4

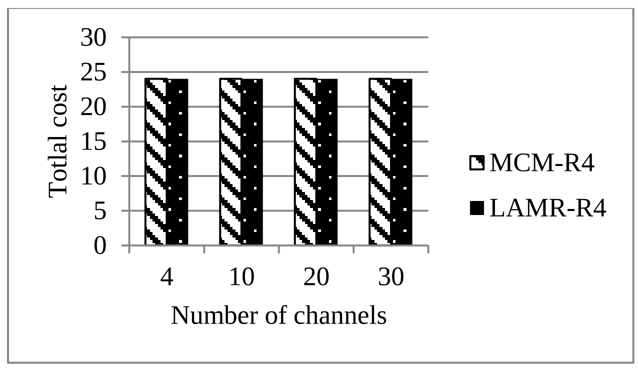

Figure 32: Total tree cost in 4-radio versions of LAMR and MCM as a function of channel number variations while the number of multicast receivers is 10

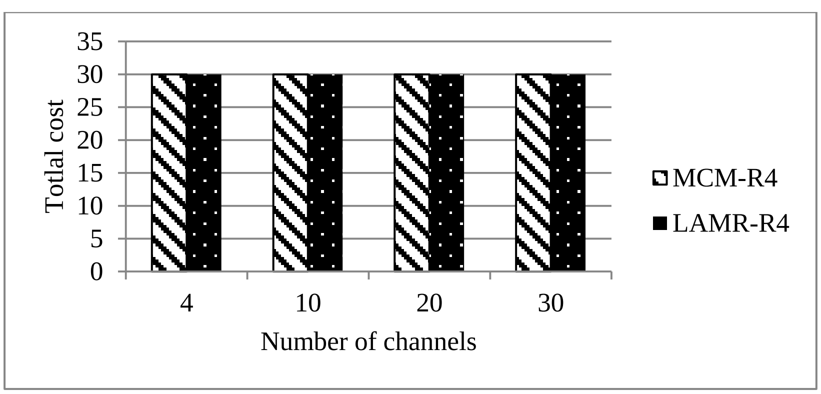

Figure 33: Total tree cost in 4-radio versions of LAMR and MCM as a function of channel number variations while the number of multicast receivers is 15

6. Discussion

Learning automata learn how to choose the optimal action by iteratively acting and receiving stochastic reinforcement signals from the environment. Using the received signal which exhibits desirability of the chosen actions, learning automata modify the probability of the taken action in favor of the most promising action. Particularly, in case of our scheme learning automata iteratively tune channel-radio associations by searching the solution space of the problem to learn the most admissible actions. Clearly, during the second phase of the LAMR protocol, the learning automaton residing on each radio gradually learns which association is the optimal choice. That is, in spite of their effectiveness in various domains, learning automata have been criticized for having a slow convergence rate [52].

7. Conclusion

In this paper, we investigated the problem of multicast routing in multi-channel multi-radio wireless mesh network which is known as NP-hard. Therefore, we proposed a new scheme based on learning automata to overcome this problem. Experimental results demonstrate that our method called LAMR significantly outperforms the existing methods in terms of achieved throughput, average end-to-end delay, average packet delivery ratio, and total cost. Furthermore, we analyzed the efficacy of LAMR in configurations with four radio interfaces per each node. Analyzing the results indicates using more interfaces along with enough channels can significantly improve the achieved throughput. Also, extensive performance studies conducted for this configuration show the superiority of our design over the protocol proposed by Zeng et. al (2010).

References

[1] I. F. Akyildiz and X. Wang, "*Wireless Mesh Networks: A Survey*", Elsevier Journal on Computer Networks, Vol. 47, No. 4, pp. 445-487, 2005

[2] J. Martínez and J. M. Bafalluy, "A Survey on Routing Protocols That Really Exploit Wireless Mesh Network Features", Journal of Communications, Vol. 5, No. 3, pp. 211-231, 2010

[3] L. Ma, G. Zhu, and B. Yin, "Optimization Models of Multi-Channel Assignment in Multi-Radio Wireless Mesh Networks, 4th International Conference on Wireless Communications, Networking and Mobile Computing, pp: 1-5, 2008.

[4] P. Baul, A. Adya, J. Padhye, and A. Walman, "Reconsidering Wireless Systems with Multiple Radios", ACM SIGCOMM Computer Communications Review, Vol. 34, No. 5, pp. 39-46, 2004

[5] A. Das, R. Vijayakumar, and S. Roy, "Static Channel Assignment in Multi-Radio Multi-Channel 802.11 Wireless Mesh Networks: Issues, Metrics and Algorithms", 49th IEEE Global Telecommunications Conference (GLOBECOM), San Francisco, California – USA, 2006

[6] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks", Information Theory, IEEE Transactions on Vol. 46, No. 2, pp. 388-404, 2000

[7] H. Skalli, S. Ghosh, S. K. Das, and M. Conti, "Channel Assignment Strategies for Multiradio Wireless Mesh Networks: Issues and Solutions", IEEE Communications Magazine, Vol. 45, No 11, pp. 86-95, 2007

[8] A. P. Subramanian, H. Gupta, and S. R. Das, "Minimum Interference Channel Assignment in Multi-Radio Wireless Mesh Networks", Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON"07), pp. 481– 490, 2007

[9] A. H. Mohsenian, V. Wong, "Joint Optimal Channel Assignment and Congestion Control for Multi-Channel Wireless Mesh Networks", IEEE International Conference on Communications (ICC 2006), Istanbul, Turkey, 2006

[10] K. Ramachandran, E. Belding, K. Almeroth, and M. Buddhikot, "Interference Aware Channel Assignment In Multi-Radio Wireless Mesh Networks", IEEE INFOCOM, pp. 1-12, 2006

[11] M. Marina and S. Das, "A Topology Control Approach for Utilizing Multiple Channels in Multi-Radio Wireless Mesh Networks", 2nd International Conference on Broadband Networks (BROADNETS 2005), Boston, Massachusetts – USA, 2005

[12] A. Das, H. Alazemi, R. Vijayakumar, and S. Roy, "Optimization Models for Fixed Channel Assignment In Wireless Mesh Networks With Multiple Radios", 2nd IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON), Santa Clara, California – USA, pp. 463–474, 2005

[13] M. Alicherry, R. Bhatia, and L. Li, “Joint Channel Assignment and Routing for Throughput Optimization in Multi-Radio Wireless Mesh Networks”, 11th ACM International Conference on Mobile Computing and Networking (MOBICOM 2005), Cologne, Germany, 2005.
[14] J. Tang, G. Xue, and W. Zhang, “Interference-Aware Topology Control and QoS Routing in Multichannel Wireless Mesh Networks”, ACM MOBIHOC, pp. 68-77, 2005
[15] A. Raniwala, K. Gopalan, and T. Chiueh, “Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks”, Mobile Computing and Communications Review Vol. 8 No. 2, pp. 50–65, 2004
[16] F. Tasaki, H. Tamura, M. Sengoku, and S. Shinoda, “A New Channel Assignment Strategy towards the Wireless Mesh Networks”, APCC‟04, pp. 71–75, 2004
[17] M. Kodialam and T. Nandagopal, “Characterizing The Capacity Region in Multi-Radio Multi-Channel Wireless Mesh Networks”, 11th ACM International Conference on Mobile Computing and Networking (MOBICOM 2005), Cologne, Germany, pp. 73-87, 2005.
[18] B. Keegan, K. Kowalik, and M. Davis, “Optimisation of Multicast Routing in Wireless Mesh Networks”, Information Technology and Telecommunications Conference, Galway Mayo Institute of Technology, 2008
[19] P. M. Ruiz, and A. F. Gomez-Skarmeta, “Approximating Optimal Multicast Trees in Wireless Multihop Networks”, IEEE ISCC05, pp. 686–691, 2005
[20] U. Nguyen and J. Xu, “Multicast Routing In Wireless Mesh Networks: Minimum Cost Trees Or Shortest Path Trees?”, IEEE Communication Magazine., Vol. 45, No. 11, pp. 72–77, 2007
[21] U. T. Nguyen, “On Multicast Routing in Wireless Mesh Networks”, Elsevier Journal of Computer Communications, Vol. 31, No. 7, pp. 1385–1399, 2008
[22] S. Roy, D. Koutsonokolas, S. Das, and Y. C. Hu, “High-Throughput Multicast Routing Metrics in Wireless Mesh Networks”, Ad hoc Networks, Vol. 6, No. 7, pp. 878–899, 2008
[23] X. Zhao, C. T. Chou, J. Guo, and S. Jha, “Protecting Multicast Sessions in Wireless Mesh Networks”, IEEE Conference on Local Computer Networks (LCN‟06), pp. 467–474, 2006
[24] J. Yuan, Z. Li, W. Yu, and B. Li, “A Cross-Layer Optimization Framework for Multihop Multicast in Wireless Mesh Networks”, IEEE Journal of Seleted Areas In Communications, Vol. 24, No. 11, pp. 2092–2103, 2006
[25] W. A. Shittuet. al., “A Proposed QoS Multicast Routing Framework for Next-Generation Wireless Mesh Network”, International Journal of Computer Science and Network Security, Vol.8, No.9, pp. 280-284, 2008
[26] P. M. Ruiz, F. J. Galera, C. Jelger, and T. Noel, “Efficient Multicast Routing In Wireless Mesh Networks Connected to Internet”, First International Conference on Integrated Internet Adhoc& Sensor Networks, 2006
[27] I. F. Akyildiz and X. Wang, “Cross-Layer Design in Wireless Mesh Networks”, IEEE Transaction on Vehicular Technology, Vol. 57, No. 2, pp. 1061 – 1076, 2008
[28] O. B. Karimi, J. Liu, and Z. Li,“Multicast in Multi-Channel Wireless Mesh Networks”, Networking, LNCS 6091, pp. 148–159, 2010.
[29] J. Xu, “Multicast in Wireless Mesh Networks”, M.S. Thesis, York University, Toronto, 2006
[30] A. Gopinathan, Z. Li, and C. Williamson, “Optimal Multicast in Multi-Channel Multi-Radio Wireless Networks”, In The Proceedings of 16th IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Imperial College London, Uk, pp. 21-23, 2009
[31] G. W. Zeng, B. Ding, Y. L. Xiao, and M. Mutka, “Multicast Algorithms for Multi-Channel Wireless Mesh Networks”, IEEE International Conference on Network Protocols (ICNP‟07), pp. 1–10, 2007
[32] G. W. Zeng, et. al. "Efficient Multicast Algorithms for Multichannel Wireless Mesh Networks", IEEE Transactions on Parallel and Distributed Systems, Vol. 21 No. 1, pp. 86-99, 2010
[33] H. L. Nguyen and U. T. Nguyen, “Minimum Interference Channel Assignment for Multicast in Multi-Radio Wireless Mesh Networks”, IEEE International Wireless Communications and Mobile Computing Conference (IWCMC‟08), pp. 626–631, 2008
[34] H. L. Nguyen, and U. T. Nguyen, “Channel Assignment for Multicast in Multi-Channel Multi-Radio Wireless Mesh Networks”, Wiley Journal of Wireless Communications and Mobile Computing: Special Issue on Next Generation Wireless Communications and Mobile Computing Networking Technologies, Vol. 9, pp. 557–571, 2009
[35] Z. Yin, Z. Li, and M. Chen, “A Novel Channel Assignment Algorithm for Multicast in Multi-Radio Wireless Mesh Networks”, IEEE ISCC‟07, pp. 283–288, 2007
[36] H. Cheng, and S. Yang, “A Genetic-inspired Joint Multicast Routing and Channel Assignment Algorithm in Wireless Mesh Networks”, The 8th Annual Workshop on Computational Intelligence, pp. 159 – 164, UK, 2008
[37] H. Cheng and S. Yang, “Joint Channel Assignment and Multicast Routing in Multiradio Multichannel Wireless Mesh Networks Using Simulated Annealing”, SEAL, LNCS 5361, pp. 370-380, 2008

[38] H. Cheng and S. Yang, “Joint QoS Multicast Routing and Channel Assignment in Multiradio Multichannel Wireless Mesh Networks Using Intelligent Computational Methods”, International Journal of Applied Soft Computing, Vol 11, No. 2, pp. 1953-1964, 2011
[39] H. L. Nguyen, and U. T. Nguyen, “Bandwidth Efficient Multicast Routing in Multi-channel Multi-radio Wireless Mesh Networks”, International Conference on Ultra Modern Telecommunications & Workshops, pp: 1 – 8, 2009
[40] S. Lim, C. Kim, Y. Ko, and N. H. Vaidya, “An Efficient Multicasting for Multi-Channel Multi-Interface Wireless Mesh Networks”, IEEE Military Communications Conference (MILCOM), pp. 1-7, 2009
[41] Y. Xu and R. Qu, "A Hybrid Scatter Search Meta-heuristic for Delay-constrained Multicast Routing Problems", Journal of Applied intelligence, Vol. 36, No. 1, pp. 229-241, 2012
[42] A. F. Araújo and C. Garrozi, “MulRoGA: A Multicast Routing Genetic Algorithm approach considering multiple objectives”, Journal of Applied Intelligence, Vol. 32 No. 3, pp. 330-345, 2010
[43] S. J. Kim and M. K. Choi, “Evolutionary Algorithms for Route Selection and Rate Allocation in Multirate Multicast Networks”, Journal of Applied Intelligence, Vol. 26, No. 3, pp. 197-215, 2007
[44] J. Crichigno, M. Y. Wu and W. Shu, “Protocols and Architectures for Channel Assignment in Wireless Mesh Networks”, Ad Hoc Networks, Vol. 6, No. 7, pp. 1051-1077, 2008
[45] D. P. Palomar, M. Chiang, “A Tutorial on Decomposition Methods for Network Utility Maximization,” IEEE Journal of Seleted Areas in Communications, Vol. 24, No. 8, pp. 1439- 1451, 2006
[46] E. Hossain and K. K. Leung, “*Wireless Mesh Networks, Architectures and Protocols*”, Springer, 2008
[47] Y. Ding, Y. Huang, G. Zeng, and L. Xiao, “Channel Assignment with Partially Overlapping Channels in Wireless Mesh Networks”,4th Annual International Conference on Wireless Internet, pp. 963-978, 2008
[48] M. A. L. Thathachar and P. S. Sastry, “*Varieties of Learning Automata: An Overview*”, IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 32, No. 6, pp. 711–722, 2002.
[49] K. S. Narendra and M.A. L. Thathachar, “*Learning Automata: An Introduction”,* Prentice-Hall Inc, 1989.
[50] O. C. Granmo, B. J. Oommen, “Optimal sampling for estimation with constrained resources using a learning automaton-based solution for the nonlinear fractional knapsack problem”, Applied Intelligence, Vol. 33, No. 1, pp. 3-20, 2010
[51] J. A. Torkestani and M. R. Meybodi, “Mobility-based Multicast Routing Algorithm in Wireless Mobile Ad Hoc Networks: A Learning Automata Approach”, Journal of Computer Communications, Vol. 33, No. 6, pp. 721–735, 2010
[52] A. Yazidi, O. C. Granmo, and B. J. Oommen, “Service selection in stochastic environments: a learning-automaton based solution” , Journal of Applied Intelligence Vol. 36, No. 3, pp. 617-637, 2012
[53] R. Vafashoar, M. R. Meybodi, and A. H. Momeni, "CLA-DE: A Hybrid Model Based on Cellular Learning Automata for Numerical Optimization", Journal of Applied Intelligence, Vol. 36, No. 3, pp. 735-848, 2011
[54] A. Raniwala, and T. C. Chiueh, “*Architecture and algorithms for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network*”. IEEE INFOCOM, pp. 2223 – 2234, 2005
[55] O. T. Johnson, *Information theory and the central limit theorem,* Imperial College Press, p. 88, ISBN 1860944736, 2004.
[56] http://www.omnetpp.org