JRCA: A Joint Routing and Channel Assignment Scheme for Wireless Mesh Networks

Amitangshu Pal and Asis Nasipuri
Electrical & Computer Engineering, The University of North Carolina at Charlotte, Charlotte, NC 28223-0001
E-mail:{apal,anasipur}@uncc.edu

Abstract- **In this paper we consider the joint channel assignment and routing problem in multi-radio multi-gateway wireless mesh networks for improving the quality of communications in the network. This channel assignment problem is proven to be an *NP-complete* problem. We present a novel backtracking and genetic algorithm based channel assignment and quality aware route selection scheme to maximize the overall performance of communications while reducing the computational complexity. We perform extensive simulation studies that show that our proposed channel assignment and route selection scheme performs significantly better than single channel and random channel selection based schemes.**

Keywords: **Wireless mesh networks, channel assignment, QoS, genetic algorithm.**

I. Introduction

Wireless mesh networking is emerging as a viable technology for extending and enhancing broadband Internet access for mobile clients located at the edge of wired networks. Co-channel interference is the main factor that reduces the network throughput in such networks. To cope with this, the IEEE 802.11 standards provide multiple overlapping frequency channels to support multiple simultaneous transmissions in the same interference region. For example, IEEE 802.11b/g offers 3 non-overlapping channels, while IEEE 802.11a offers 12 non-overlapping channels. By exploring the advantage of multiple channels and multiple radios, the system performance of the mesh networks can be improved significantly compared to single-channel wireless access networks. However, all these benefits can only be achieved by applying a carefully designed channel assignment scheme so as to utilize these multiple channels and radios effectively.

In addition to effective channel assignment and management of usage of radios or network interface cards (NICs) at the nodes, a key factor that determines the end-to-end communication quality in wireless mesh networks is the routing protocol. Due to the fact that co-channel interference at a node is determined by the assignments of channels to the neighboring nodes as well as their traffic patterns, an ideal approach for this problem is to consider both channel assignment and routing simultaneously. In this work, we consider an anycasting routing model, where multiple gateways are used for reducing congestion on one gateway and increase network performance. Our earlier works on this problem ([1], [2]) involved the development of quality based routing protocols in the presense of multiple gateways in single-channel wireless mesh networks. This paper extends our previous work to include multiple channels on top of our anycast quality based routing model.

We consider a centralized approach to address the joint routing and channel selection problem, where it is assumed that the gateway nodes are connected by an infrastructured network such as an optical fiber network, and collaborate with each other for determining the optimum gateway, route and channel selections for all active nodes in the network. The central challenges for addressing this problem include determining effective measures for estimating the route quality prior to channel and route selection and addressing the computational complexity of the tasks. In this paper, we propose a *joint routing and channel assignment protocol JRCA*, which addresses these challenges using the following principles. First, it employs a novel route quality metric that utilizes performance characteristics of data packet transmissions as opposed to control packets, and effectively captures the effects of intraflow and interflow interference. Second, it employs a combination of *backtracking* and *genetic algorithm* to reduce the convergence time. We show the improvement of JRCA against single chanel and random channel selection schemes using extensive simulation results.

II. Related Work

Prior work on channel assignment schemes can be broadly classified into three categories: static assignment, dynamic assignment and hybrid assignment.

Static assignment strategies assign a channel to each interface for permanent use. In [3], the authors formulate the channel assignment problem as a topology control problem. They develop a greedy algorithm that minimizes the maximum link conflict weight and simultaneously preserves the connectivity of the connectivity graph. Another tabu search based centralized scheme is proposed in [4]. In [5], the authors propose two algorithms that also use the link conflict graph to model interference. The first algorithm minimizes the average link conflict weight, while the second minimizes the maximum link conflict weight. Both algorithms are based on an approximation algorithm for the *MAX k-CUT problem*. Authors in [6] propose two integer linear-programming models. The objective is to maximize the number of simultaneous transmissions in the network, subject to connectivity restrictions. In [7], the authors present a multi-commodity network flow model used to find an upper bound of the achievable throughput for a given set of flows. In [8], the authors propose a *joint radio and channel assignment scheme (JRCA)* that first uses a maximum flow based centralized channel assignment to obtain an initial assignment of channel. Then the residual demand, i.e. actual link demand minus the allocated demand is calculated for each link. Finally, the links are visited in decresing order of residual demands and the least used channels are assigned one by one.

Dynamic protocols enforce nodes to switch their interfaces dynamically from one channel to another between successive data transmission. In [9] a centralized architecture for channel assignment and routing is presented. Given the node placement and the traffic load between each pair of nodes, the channel assignment algorithm binds each interface to a channel such that the available bandwidth on each link is proportional to its expected load. If the loads change over time, the algorithm can perform channel reassignments. In [10] a distributed architecture for routing and channel assignment is discussed. In this scheme, when a node finds a channel with a lower usage, it can perform a reassignment to that channel. A genetic algorithm based channel assignment scheme is proposed in [11] where a central unit invokes the genetic algorithm based channel selection procedure periodically and sent back that assignment to the mesh routers. In [12], the authors model the traffic flows among the mesh routers as linear programming problem, targeting to find the fair flow of each mesh router. Based on the fair flows, a weighted flow-based conflict graph is constructed and then channels are assigned to each vertex of the conflict graph based on vertex coloring scheme. The channels are reassigned after a certain time period because of the change in traffic demands.

Another set of strategies [13], [14] known as *hybrid approaches* apply a static or semi-dynamic assignment to the fixed interfaces and a dynamic assignment to the switching interfaces. In [13], the authors present a scheme where at least one interface of the receiver is assigned to a channel statically or semi-dynamically, while interfaces of the senders are dynamically switched to one of the assigned channel of the receiver. In [14], the authors propose a scheme where one radio on each mesh router operates on a *default channel* to preserve network connectivity. The authors introduce the concept of *multi-radio conflict graph* and then use a *breadth-first search* from the gateway to assign channel such that the interference is minimized.

The model proposed in this paper falls in the hybrid category, where we assume a dedicated control channel on which all nodes assign a NIC. This is the *default channel* which is mainly used to send request and reply packets. Other interfaces are switched between different data channels for data transmission. Our main contributions in this paper are as follows: Firstly, we develop a novel *interference and delay aware quality metric* based on the end-to-end probability of success and route delay. Secondly, we use this quality metric to design a joint routing and channel assignment scheme based on *genetic algorithm* in presence of multiple gateways. To reduce the convergence time we use the genetic algorithm on the subpart of the whole graph. Finally, we use extensive simulation results to show the improvement of JRCA over single channel and random channel selection schemes.

III. Problem Formulation And JRCA Design

In our network model, there are multiple gateways where source can connect to any of the gateways to reach the Internet. For each source, a gateway out of a number of gateways as well as the route from the source to that gateway is selected and at the same time the channels are assigned to each link of that route. The problem of optimal gateway and channel selection can be formulated as follows. Consider a case of n sources $\{S_1, S_2, ..., S_n\}$ and a group of m gateways $\{G_1, G_2, ..., G_m\}$ where $1 \leq m \leq n$. Also assume that the S_C is the set of C channels and all the available channels are orthogonal. The problem is to assign the n sources to m gateways and all the interfaces to C channels, so that the total quality of the network is maximized. The problem can be formulated as a *0-1 integer programming problem* as follows:

$$Maximize \sum_{i=1}^{n}\sum_{j=1}^{m} Q_{S_iG_j} \times X_{S_iG_j} \quad (1)$$

subject to

$$\sum_{j=1}^{m} X_{S_iG_j} = 1 (1 \leq i \leq n) \quad (2)$$

$$X_{S_iG_j} = 0 \text{ or } 1 \ (1 \leq i \leq n) \ (1 \leq j \leq m) \quad (3)$$

where $Q_{S_iG_j}$ is the quality of the best route between S_i and G_j, $X_{S_iG_j}$ is a binary variable used for gateway selection: if the best gateway chosen for S_i is G_j, then $X_{S_iG_j}$ =1; otherwise $X_{S_iG_j}$ =0. Constraint (2) states that S_i can only transmit all its packets to one gateway only.

We assume that each router has a limited number of interfaces. We also assume that the gateways are connected to some infrastructured network such as a fiber network so that they can collaborate with each other to select the routes for all sources and also the channel for each link in that route. For this the gateways need to know the positions of all mesh routers. In our scheme, the gateways wait for the first N RREQ packets from each source, track the route traversed by each RREQ packet, and sends a RREP packet through the route that maximizes the quality after channel assignment. For doing this, the gateways have to assign channels to all the routes and measure the quality. For quality measure, we use the quality metric discussed in Section III.A. For channel assignment, we first form the conflict graph (discussed in Section III.B.1) and use a vertex coloring scheme (where colors represent channels) for channel assignment. From this point onwards we use the word *channel* and *color* interchangably. But as the vertex coloring problem is an *NP-complete* problem [15], we apply the *genetic algorithm* [16] to solve it. We propose a novel mechanism to reduce the number of vertices on which we apply the genetic algorithm, to reduce the convergence time. This is achieved by planarizing the conflict graph using *vertex*

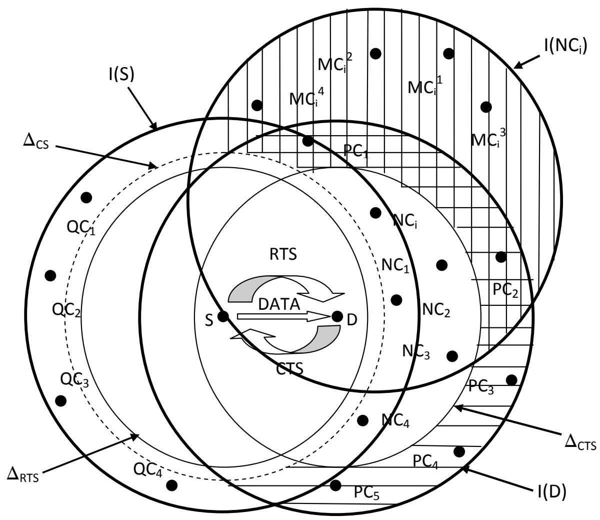


Fig. 1. Effect of interferers in presense of RTS/CTS for test link $S{\rightarrow}D$

deletion (in Section III.B.3), applying backtracking to color the planar subgraph (in Section III.B.4) and then using the genetic algorithm on the vertices that are not part of planar subgraph and those that violate the interface constraint (in Section III.B.5). The details of the scheme is described in the following subsections.

A. Route Quality Estimation

We now describe the development of a routing metric that tries to capture the quality of multihop routes in terms of the end-to-end probability of success (POS) and delay. Our approach is to determine the major factors that affect the POS and delay in multihop wireless networks using the *IEEE 802.11 MAC* and develop models that involve a compact set of parameters. These models are then applied to develop the routing metric that can be used to estimate the route quality using parameters obtained from the network. The detailed analysis of POS and delay are explained in our earlier works ([17], [18], [2], [19]). Here we propose a summary of the analysis.

1) Probability of success (POS): We model the POS of a test link $S{\rightarrow}D$ in Fig. 1 using a simple set of measurable parameters. Here, Δ_{RTS} and Δ_{CTS} denote the regions where the RTS and CTS packets for the test link can be received, respectively. Δ_{CS} denotes the area around *S* where nodes can sense the transmission from *S*. For any node i, $I(i)$ denotes the area from where a transmission from any node $j \in I(i)$ can interfere with a packet being received at i. The POS in the test link is obtained by analyzing the possible cases that can cause a data packet transmission in the test link to be unsuccessful, which are described below.

Case-1: The transmitted data packet is unsuccessful due to interference from nodes that are outside the reception range of the CTS packet but within the interference range of D. These nodes are marked as PC_i in Fig. 1, of which we assume q and r nodes are in sending and receiving states, respectively. In the absence of any other interferer that can affect the reception of the test data packet, the effect of a single interfering sending node among PC_i can be evaluated as $e^{\frac{-\lambda\times DLEN}{B}}$, where λ is the arrival rate of data packets, which is assumed to follow a Poisson distribution, B is the bandwidth of the channel and $DLEN$ is the data packet length. For q independent senders $(PC_1, PC_2, ..., PC_q)$ in this region, the probability of success of the *DATA* packet is given by $e^{\frac{-\lambda\times DLEN\times q}{B}}$.

The receiving nodes among PC_i can interfere by the transmission of CTS packets during the transmission of the test *DATA* packet; thus the probability of success in the presense of r such nodes is $e^{\frac{-\lambda\times DLEN\times r}{B}}$.

Case-2: The transmitted data packet is unsuccessful due to interference from nodes that are within the transmission range of D but fail to receive the CTS packet from D. Such interfering nodes, marked as NC_i in Fig. 1, do not receive the CTS from D due to an overlapping transmission from MC^i_j. The probability of this event is $1-e^{\frac{-\lambda\times DLEN}{B}}$. In general, if there are m such interferers among MC^j_i, then the probability that the CTS is not received by NC_i is $1-\prod_{i=1}^{m} e^{\frac{-\lambda\times DLEN}{B}} = 1-e^{\frac{-\lambda\times DLEN\times m}{B}}$.

The probability that NC_i, having failed to receive the CTS from D, interferes with the reception of the *DATA* packet at D is then given by $(1-e^{\frac{-\lambda\times DLEN}{B}})$, consequently the POS of the test *DATA* transmission from S to D in the presence of interference from any node NC_i is given by $1-(1-e^{\frac{-\lambda\times DLEN}{B}})(1-e^{\frac{-\lambda\times DLEN\times m}{B}})$. In the presence of n such nodes $(NC_1, NC_2, \cdots, NC_n)$, the POS is given as $\prod_{i=1}^{n} 1-(1-e^{\frac{-\lambda\times DLEN}{B}})(1-e^{\frac{-\lambda\times DLEN\times m}{B}})$.

Case-3: The data transmission is unsuccessful due to failure in receiving the ACK packet at S, caused by interference from neighbors of S. The ACK packet can be interfered only by RTS packets transmitted by neighbors of S. As the packet sizes of both ACK and RTS are very small, the corresponding probability of interference is also small, and we ignore this case.

By taking into account all the factors described above, the POS for the *DATA* packet on the test link is given as:

$$POS = \left\{\prod_{i=1}^{n} 1-\left(1-e^{\frac{-\lambda\times DLEN}{B}}\right)\left(1-e^{\frac{-\lambda\times DLEN\times m}{B}}\right)\right\} \times e^{\frac{-\lambda\times DLEN\times q}{B}} \times e^{\frac{-\lambda\times DLEN\times r}{B}} \quad (4)$$

Hence, the POS of a link can be estimated using the above equation as long as the values of m, n, q and r are known, which can be obtained in a static mesh network from knowledge of node locations. The above model for the POS has been verified with extensive simulation experiments, which could not be included in this paper due to space constraints. We refer the readers to [17], [18], [2] for further details.

2) Queuing And Access Delay: The transmission delay in a wireless link using the 802.11 MAC depends on the complex interaction of channel access, back-offs, transmissions and retransmission attempts of multiple active nodes in the neighborhood of the test link. However, it can be shown that for a given offered load, the total delay on a test link can be approximately estimated from the number of active neigbors of the sender (n_a) and the number of active neighbors of the receiver (n_b) as follows:

$$T_d(n_a, n_b) = A(n_a^2 + n_b^2) + B(n_a + n_b) + C \tag{5}$$

We use simulation experiments to develop this approximate model for $T_d(n_a, n_b)$ at various offered loads in [17], [18], [2]. In equation (5) A, B, C varies with load and are obtained by best fitting the simulated values.

3) Route Quality Metric: With these, we propose a route quality metric with the objective of maximizing the end-to-end POS and minimizing the end-to-end delay in the route. For any route, the net POS is taken as the product of the POS of every individual link on the route and the total delay is taken as the sum of the link delays. Consequently, we define the route quality metric for route R of length v as follows:

$$Q(R) = \frac{\prod_{f=1}^{v} P_S(I_f)}{\sum_{f=1}^{v} T_d(n_{af}, n_{bf}) + \sum_{f=1}^{v} s_d \times y_f} \tag{6}$$

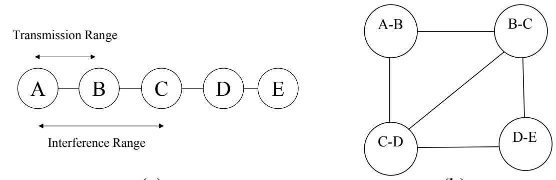
Here, f is a link on the route from source to gateway , $P_S(I_f)$ is the POS of link f, I_f is the set of interferers. $T_d(n_{af}, n_{bf})$ is the delay with n_{af} and n_{bf} active neighbors at the sender and the receiver end respectively. s_d is the switching delay for an interface to switch from one channel to another and y_f is a binary variable which is 1 when the interfaces of link f switch and 0 otherwise. For positive switching delay, this model prefers routes that avoid frequent switching of channels.

B. Channel Assignment and Route Selection Based on Our Route Quality Estimation

Our approach for maximizing the quality metric through joint route and channel selection requires an effective representation of co-channel interference, which we model using the conflict graph.

1) Conflict Graph: Consider a wireless mesh network where all routers have identical transmission ranges (denoted by R) and the interference range is denoted by $R' \geq R$. For each link $i - j$ in the connectivity graph, the *conflict graph* [20] contains a vertex. There exists an edge between two nodes (say, $A - B$ and $C - D$) in the conflict graph if the corresponding links interfere in the connectivity graph. A transmission from A to B is successful if no other node located R' from B transmits at the same time. In the presence of RTS/CTS, it is additionally required that all nodes located within R' from A refrain from transmission. Thus, there is an edge between $A - B$ and $C - D$ in the conflict graph if either A or B are located within distance R' from C or D. Fig. 2 shows an example illustrating this model.

Hence, if there is a link between two vertices in the conflict graph, then those two vertices interfere each other, thus we have to assign different channels to these two vertices (vertices in the conflict graph are links in the connectivity graph). This is similar to the vertex coloring problem, i.e. no two vertices in the conflict graph having a link have the same color. More precisely, if $G = (V, E)$ be an undirected graph with n vertices then a coloring of G is a mapping $\pi : V \rightarrow C$ such that for



(a) (b)

Fig. 2. (a) Connectivity graph and (b) Conflict graph

any two vertices x and y if $(x, y) \in E$, then $\pi(x) \neq \pi(y)$. So, we can formulate the channel assignment problem as a vertex coloring problem.

2) Planar Graph and the Four Color Theorem: In graph theory, a *planar graph* is a graph that can be embedded in a plane, i.e. it can be drawn on a plane in such a way that its edges intersect only at their endpoints [21]. On the other hand, graphs that are not planar are called *non-planar graphs*. According to *four color theorem* all planar graphs are four colorable. Thus, if we can get the planar subgraph of the conflict graph, we can color that subgraph with four colors.

3) Vertex Deletion to get the Planar Subgraph: We use *vertex deletion* to get the planar subgraph of the conflict graph. The Boyer and Myrvold planarity test [21] is used to check whether a graph is planar or not in linear time. First the planarity of the conflict graph G is checked. If it is non-planar, the vertex with the highest degree is removed from G and placed in *genetic-colored-list (GCL)*, and then the planarity condition is checked again on the remaining graph (line 3-6 in Algorithm 1). This vertex deletion process is repeated until the remaining graph becomes planar. At the end of this process, GCL consists of the removed vertices, all the other vertices are stored in *fixed-colored-list (FCL)*. Thus, the subgraph consists of FCL and their edges is planar.

Algorithm 1 Function Vertex Deletion (Input graph G)

```
1: GCL = FCL = NULL
2: Sort v_i ∈ G in decreasing order of vertex degree
3: while G ≠ PLANAR do
4:    G = G \ v_i, v_i is of maximum degree in G
5:    GCL = GCL ∪ v_i
6: end while
7: FCL = G
8: return GCL and FCL
```

4) Our Proposed Algorithm for Coloring the Planar Subgraph: Let $\tilde{G}$ be the planar subgraph that consists of vertices in FCL and their corresponding edges. Now, according to the *Four Color Theorem*, $\tilde{G}$ can be colored with 4 colors. So, we propose an algorithm (line 4 - line 23 in Algorithm 2) based on *backtracking* to color $\tilde{G}$. This is explained with the help of Fig. 3, where the graph in Fig. 3(a) is the planar subgraph and Fig. 3(b) shows it's backtracking tree. Let us assume that at first all the vertices can use RED, BLUE, GREEN and BLACK that are indexed as 1, 2, 3, 4 respectively in the *Color* array in Algorithm 2. In Algorithm 2, the nodes in FCL are denoted as $\{v_1, v_2, ..., v_n\}$. We start with vertex *A* and color it RED. So, all the neighbors of *A* cannot use RED. Thus *B* can use only

BLUE, GREEN and BLACK. In this way if we proceed and if *C*, *D*, *E* and *F* are colored with RED, BLUE, GREEN and BLACK respectively, then there is no color left for *E*. Thus we need to backtrack and color *D*, *E* and *F* with GREEN, BLUE and GREEN respectively, and then G is colored with BLACK. This process of backtracking is guaranteed to give a 4-coloring to $\tilde{G}$. After coloring the subgraph $\tilde{G}$ in this fashion, if the number of channels used (C_u) in any node in the connectivity graph is more than the number of interfaces (I), then the interface constraint is violated. Thus, for each node in the connectivity graph, we check the interface constraint and if this constraint is violated, $C_u - I$ links (vertices in conflict graph) around that node are selected randomly and added to GCL (line 24). Then the GCL is passed to the *genetic algorithm* (line 25).

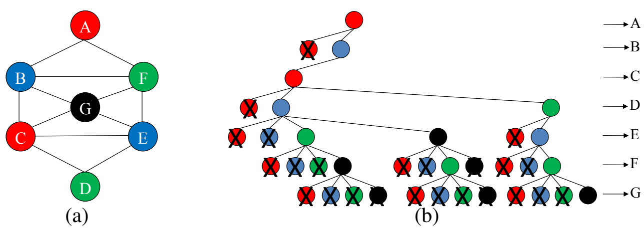


Fig. 3. (a) A planar graph and (b) Backtracking tree of graph

Even if the worst case complexity of backtracking is exponential with the number of nodes in the conflict graph, as all the nodes have only 4 colors, the complexity is not that high. At the same time as there are a number of solutions for a vertex coloring problem, we do not need to explore the whole tree. As an example, if we explore the whole search tree of n vertices, we need to explore $1 + 4 + 4^2 + ... + 4^{n-1} = \frac{4^n - 1}{3}$ nodes. At the same time we get a large number of solutions. For instance, a *tree* and a *cycle* of n vertices can be colored in $t(t-1)^{n-1}$ and $(t-1)^n + (-1)^n(t-1)$ ways respectively, with t colors [22]. Thus, the number of solutions with 4 colors for a tree and a cycle are 4.3^{n-1} and $3^n + 3.(-1)^n$, respectively. As a conflict graph is composed of trees and cycles, the search tree will have a very large number of solutions. Hence, for getting one solution, we need to search a small fraction of the whole search tree. Let us consider the effectiveness of this backtracking strategy to reduce the convergence time of the genetic algorithm, which is to be performed on the vertices in GCL. As mentioned in [23], the number of vertex deletions required to make the non-planar graph $K_{m,n}$ planar is given by $\min\{m, n\} - 2$. Hence, for $m = n = 100$, the number of vertex deletions is 98. Thus FCL consists of 102 vertices, whereas GCL consists of only 98 vertices instead of 200. Consequently, in this case we can reduce the length of GCL by around 50% which results in reduced convergence time of the genetic algorithm. It must be noted that in the worst case (when the conflict graph is a complete graph), almost all the vertices are in the GCL, but in reality, conflict graphs are hardly complete graphs. Thus on an average, our scheme is able to reduce the convergence time of the genetic algorithm.

5) Our Proposed Genetic Algorithm for Channel Selection: Genetic algorithms are probabilistic techniques that mimic the natural evolutionary process. A genetic algorithm maintains a population of candidate solutions. This has the potential to better explore the search space. Each candidate solution in the population is encoded into a structure called the *chromosome*. To each chromosome, a value called *fitness value* is assigned, which represents the quality of the candidate solution. The process of assigning fitness values to chromosomes is called *evaluation*. A selection process simulates the *survival of the fittest* paradigm from nature. Better-fitted chromosomes have higher chances of surviving to the next generation. The number of chromosome per generation is constant.

As in natural life, offspring chromosomes are obtained from parent chromosomes. One possibility is for two parents to exchange encoded information and thus creating two new offsprings; this process is called *crossover*. Another possibility is to alter the encoded information in a chromosome obtaining a slightly different new chromosome; this process is known as *mutation*. Some other chromosomes simply survive unaltered, while others die off. Mutation and crossover are referred to as *genetic operators*.

Genetic Representation: Let U is the number of vertices in GCL. Let us define a chromosome as a vector $(c_1, c_2, ..., c_U)$, where $c_i \in S_C$ is the channel/color assigned to vertex i. As an example, if $U = 6$ then chromosome 314252 means vertices $1, 2, 3, 4, 5, 6$ are assigned to channels $3, 1, 4, 2, 5, 2$ respectively. We assume that there are M chromosomes in a *mating pool*. The fitness value of each chromosome is the overall network quality based on the channel assignment from each chromosome. For all vertices in the GCL, initially we assign random channels in between 1 and C and make M chromosomes so that the interface constraint is satisfied.

Selection Process: Selection is the process of choosing individual chromosomes to participate in reproduction. After geting the initial M chromosomes in the mating pool, the fitness values of all the chromosomes are calculated based on the quality metric. We use the well known *elitism* selection process, where the $M_e < M$ best chromosomes (as determined from their fitness evaluations) are placed directly into the next generation. This guarantees the preservation of the M_e best chromosomes at each generation. Note that the elitist chromosomes in the original population are also eligible for selection and subsequent recombination. Next, $M - M_e$ parents are selected based on *roulette wheel* selection process. So, better chromosomes have higher chances to be selected. These $M - M_e$ parents take part in crossover and mutation.

Crossover and Mutation Process: Crossover is designed to propagate and exchange information between two parent chromosomes and the result is two child chromosomes. We use two point crossover and the two crossing points are selected randomly between 1 and U. Usually, high values are chosen for the crossover probability $(90\% - 100\%)$, we assume a value of 100% in our simulations. After a crossover, if the child chromosomes do not satisfy the interface constraint, some channels are merged randomly to meet the constraint. One example of crossover is shown in Fig. 4(a).

The mutation process is performed for each new generation after crossover. In this process two random numbers are generated between 1 and U and the colors of these two vertices are interchanged. Generally mutation probability is pretty low, we assume a value of 1% in our simulations. An example of mutation process is shown in Fig. 4(b) where the colors of vertex 1 and 6 are exchanged.

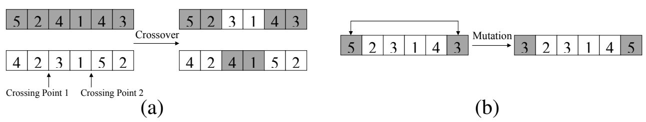


Fig. 4. (a) Two-point crossover and (b) Mutation

The algorithm stops when the best solution does not improve significantly for a fixed number of consecutive iterations or a large predefined number of iterations is reached. When the stopping criterion is reached, the algorithm chooses the chromosome/solution with the highest fitness value. This process is repeated for all the candidate routes, and the route with highest fitness value/quality is selected.

We now calculate the reduction of the convergence time of the genetic algorithm achieved from our scheme. As mentioned in [24], the probability that the genetic algorithm converges at generation t of chromosome length l is given by

$$P(t,l) = \left[1 - \frac{6p_0(1-p_0)}{M}\left(1-\frac{2}{M}\right)^t\right]^l \qquad (7)$$

where p_0 is the initial frequency of the allele and M is the population size (*mating pool* size). Now for an example, if we assume $M = 1000$ and $p_0 = 0.5$, then to get 90% probabilty of convergence ($P(t,l) = 0.9$) for a 200 bit chromosome ($l = 200$), the algorithm takes 522 generations to converge, but if we can reduce the chromosome length by 50% ($l = 100$) it takes 176 generations to converge. This shows a significant amount of reduction in convergence time.

C. JRCA Routing Protocol

With these, the JRCA routing protocol can be described by the following set of actions.

Route Discovery: When the source does not have a route to the destination, it broadcasts a route request packet (RREQ) to its neighbors. In addition to many other fields, the RREQ contains a RREQ ID, the destination address, the source address, the number of active neighbors of the sender (A), the sequence of nodes that it has traversed and a timestamp. The RREQ ID combined with the source address uniquely identifies a route request. This is required to ensure that the intermediate nodes rebroadcast a route request only once in order to avoid broadcast storms. If any intermediate node receives a RREQ more than once, it discards it. All intermediate nodes do the same thing until the RREQ reaches the destination. The timestamp is used to reduce unnecessary flooding of RREQ packets throughout the network.

Algorithm 2 Algorithm for finding the color assignment

```
INPUT : Simple undirected graph G and the set of channels
OUTPUT : Color assignment of G
Vertex Deletion (G)
All_nodes_colored = false
v_i = v_1
Color(v_1) = 0
while All_nodes_colored == false do
  while Color(v_i) < 4 do
    if All_nodes_colored == true then
      break
    end if
    Color(v_i) = Color(v_i) + 1
    if ValidColor(Color(v_i), v_i) == true then
      if v_i == v_n then
        All_nodes_colored = true
      else
        v_i = v_{i+1}
        Color(v_i) = 0
      end if
    end if
  end while
  v_i = v_{i-1}
end while
GCL = GCL ∪ (vertices violating interface-constraint)
Perform Genetic-Algorithm(GCL)
return G with vertex coloring
```

Route Selection: The destinations (gateways) wait for the first ten RREQ packet, run *backtracking* and *genetic algorithm* on all the routes carried by the RREQ packets, collaborate with each other and choose routes and channels that maximizes route quality Q. For calculating Q, gateways use the node location and neighborhood information of the nodes. After choosing the route with highest Q, the gateway that receives the best route (carried of RREQ) forwards a route reply packet (RREP) back on the same route. All the intermediate nodes forward the RREP back to the source, perform channel switching if required, and update their routing table entry. The source then starts sending the data packets via this route. The *default channel* is used for the transmissions of RREQ and RREP packets.

Route Maintenance: If a routing table entry is not used for a long time, that entry is erased. This is required as the network scenario changes with time, thus after a long time if a source needs a route to the gateway, it has to start a route discovery to get a good quality route.

The overall scheme of joint route and channel selection (JRCA) is depicted in Fig. 5.

IV. Performance evaluation of JRCA

We next present the performance of the proposed JRCA routing protocol in comparison to single channel and random channel selection schemes. We use the *network simulator–2 (ns2)* [25] to measure the performance of different protocols, with substantial modifications in the physical and the MAC layers, to model the cumulative interference calculations and also include the physical carrier sensing based on cumulative received power at the transmitter. The *DataCapture* is also modelled in our modified *ns-2* version. Next we extend *ns-2* to support multiple channels and multiple radios as described in [26]. For our performance evaluations, we consider a grid network consisting of 30 nodes placed in a uniform grid. We

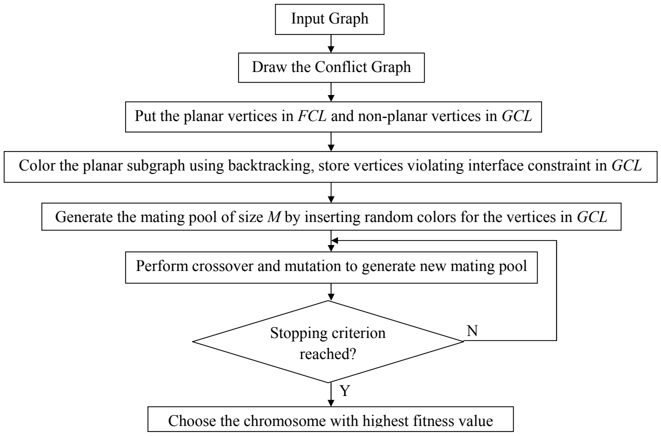


Fig. 5. Joint route and channel assignment (JRCA) scheme

choose two gateways and keep them fixed. The sources are selected randomly. Each flow runs UDP and is alive for 200 seconds. We have averaged the results over 5 such simulations. The parameters used in the simulations are listed in Table I. For simplicity, we assume the switching delay to be 0.

TABLE I
SIMULATION ENVIRONMENT

Parameter	Values	Parameter	Values
Max queue length	200	Data packets size	1000 bytes
Propagation Model	Two Ray Ground	Traffic Generation	Exponential
Antenna gain	0 dB	Transmit power	20 dBm
Noise floor	-101 dBm	SINRDatacapture	10 dB
Bandwidth	6 Mbps	PowerMonitor Thresh	-86.77 dBm
Basicrate	1 Mbps	Datarate	6 Mbps

The performance is measured in terms of the average throughput, delivery ratio, end-to-end delay, and jitter of the data flows using the single channel scheme, a multi-channel scheme with random channel selection, and JRCA. The results are shown in Fig. 6-Fig. 8.

Comparison with different number of flows: First we compare the performance of different schemes (shown in Fig. 6) with different number of flows. We select the transmission rate of 185 KBps for these set of graphs. From Fig. 6, we observe that JRCA performs significantly better than single channel scheme in terms of throughput, delivery ratio, delay and jitter. The improvement in delivery ratio is because of reduced interference due to efficient channel assignments, whereas the reduction in delay and jitter is mainly due to reduction in channel access delay from using multiple channels in neighbouring transmitting nodes. These factors result in significant improvement in the throughput. As expected, increasing the number of NICs results in higher throughput and delivery ratio and lower delay and jitter. Also, the proposed JRCA gives better performance in comparison to random channel selection scheme because of efficient channel selection.

Comparison with different loads: Fig. 7 shows the variations of throughput, delay and jitter with varying traffic load for both single and multiple channel schemes. The number of channels in these set of figures is 12 and 10 sources are chosen randomly. Here also we observe the significant improvement in throughput, delivery ratio, delay and jitter in case of JRCA than the single channel scheme and random channel selection scheme.

Comparison with different number of channels: The variations of throughput, delay and jitter with different number of channels are shown in Fig. 8. For these set of figures, we set the transmission rate to 185 KBps and with 10 active sources. As expected, a higher number of NICs improves the throughput, delivery ratio, delay and jitter because of it's ability to reduce channel conflict in neighbouring and interfering links.

V. CONCLUSION

In this paper, we address the joint routing and channel assignment scheme in multi-channel wireless mesh networks, where each router is equipped with multiple radios. We develop a backtracking and genetic algorithm based channel selection scheme for solving this problem. For route selection, we propose a novel quality based routing metric based on probability of success and delay. Using simulations in ns-2, we demonstrate the effectiveness of our routing and channel selection scheme in improving the network throughput, delivery ratio, delay and jitter.

ACKNOWLEDGEMENT

This work was supported in part by NSF grant CNS-1117790.

REFERENCES

[1] A. Pal and A. Nasipuri, "A quality aware anycast routing protocol for wireless mesh networks," in *IEEE SoutheastCon*, 2010, pp. 451–454.
[2] A. Pal and A. Nasipuri, "GSQAR: A quality aware anycast routing protocol for wireless mesh networks," in *IEEE GLOBECOM*, 2010.
[3] M. K. Marina and S. R. Das, "A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks," in *BROADNETS*, 2005, pp. 412–421.
[4] A. P. Subramanian, H. Gupta, S. R. Das, and J. Cao, "Minimum interference channel assignment in multiradio wireless mesh networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 12, pp. 1459–1473, 2008.
[5] A. K. Das, R. Vijayakumar, and S. Roy, "Static channel assignment in multi-radio multi-channel 802.11 wireless mesh networks: Issues, metrics and algorithms," in *GLOBECOM*, 2006.
[6] A. K. Das, H. M. K. Alazemi, R. Vijayakumar, and S. Roy, "Optimization models for fixed channel assignment in wireless mesh networks with multiple radios," in *In SECON*, 2005, pp. 463–474.
[7] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," in *ACM MobiCom*. ACM Press, 2005, pp. 73–87.
[8] R. Koshy and L. Ruan, "A joint radio and channel assignment (JRCA) scheme for 802.11-based wireless mesh networks," in *GLOBECOM Workshops, 2009 IEEE*, December 2009.
[9] A. Raniwala, K. Gopalan, and T. cker Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *ACM Mobile Computing and Communications Review*, vol. 8, pp. 50–65, 2004.
[10] A. Raniwala and T. cker Chiueh, "Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network," in *INFOCOM*, 2005, pp. 2223–2234.
[11] J. Chen, J. Jia, Y. Wen, D. Zhao, and J. Liu, "A genetic approach to channel assignment for multi-radio multi-channel wireless mesh networks," in *GEC Summit*, 2009, pp. 39–46.

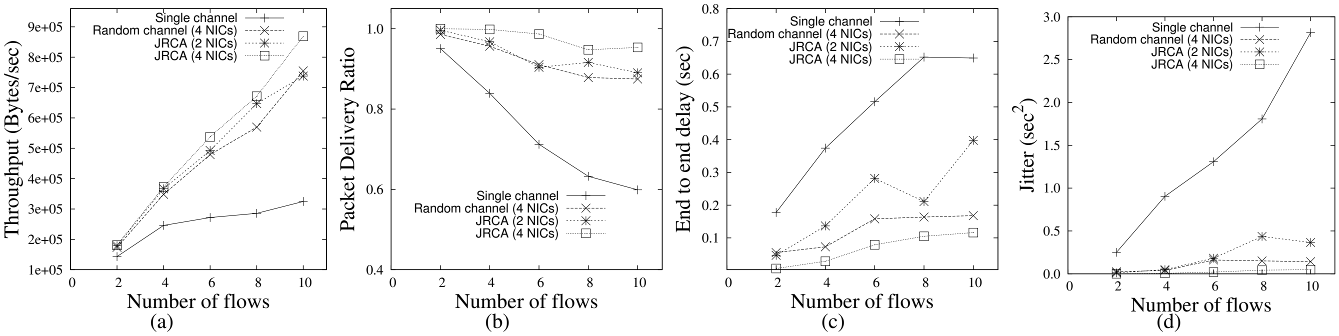


Fig. 6. Comparison of (a) Throughput (b) Delay (c) Jitter with different number of flows

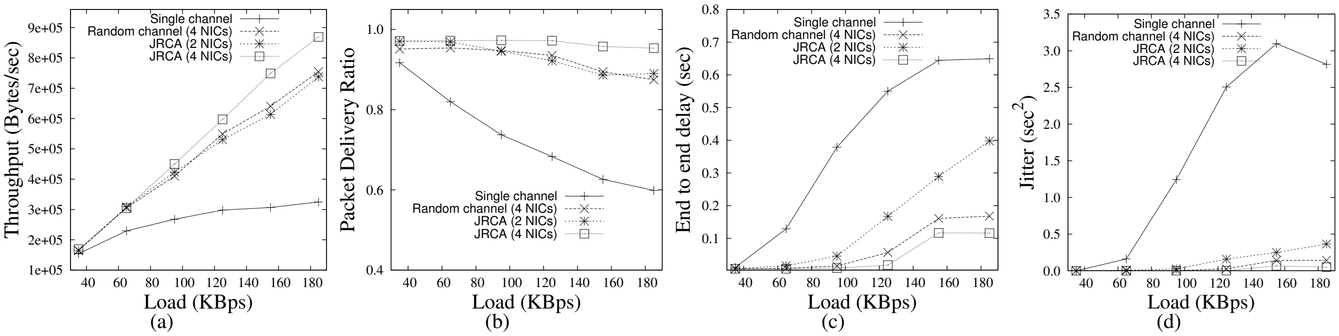


Fig. 7. Comparison of (a) Throughput b) Delivery ratio (c) Delay (d) Jitter with different loads

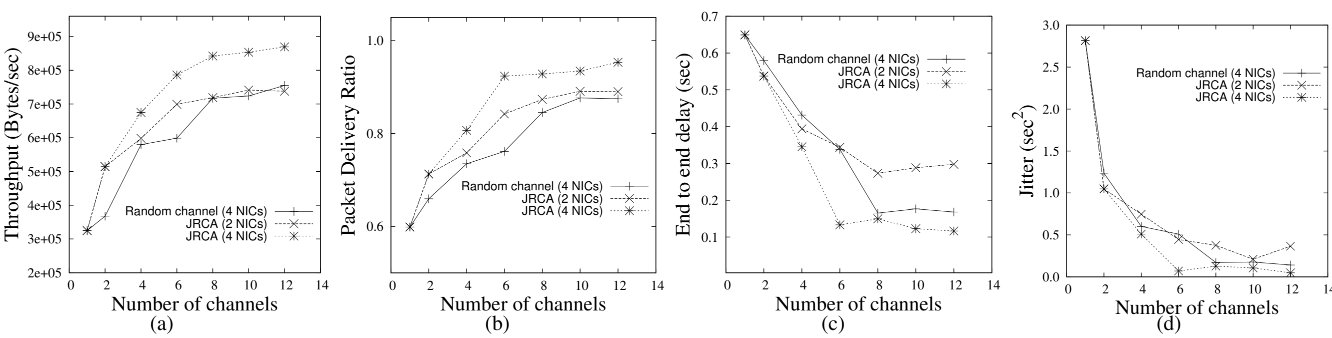


Fig. 8. Comparison of (a) Throughput (b) Delivery ratio (c) Delay (d) Jitter with different number of channels

[12] W. Fu, B. Xie, X. Wang, and D. P. Agrawal, "Flow-based channel assignment in channel constrained wireless mesh networks," in *ICCCN*, 2008, pp. 424–429.

[13] P. Kyasanur and N. H. Vaidya, "Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 10, no. 1, pp. 31–43, 2006.

[14] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks," in *INFOCOM 2006*, 2006, pp. 1–12.

[15] M. Garey, *Computers and intractability - A guide to the theory of NP-completeness*, 1995.

[16] "A tutorial for genetic algorithms," `http://www.obitko.com/tutorials/genetic-algorithms/`.

[17] A. Pal, S. Adimadhyam, and A. Nasipuri, "QoSBR: A quality based routing protocol for wireless mesh networks," in *ICDCN*, 2010, pp. 497–508.

[18] A. Pal and A. Nasipuri, "A quality based routing protocol for wireless mesh networks," *Pervasive and Mobile Computing*, vol. 7, no. 5, pp. 611–626, 2011.

[19] A. Pal and A. Nasipuri, "Performance analysis of IEEE 802.11 distributed coordination function in presence of hidden stations under non-saturated conditions with infinite buffer in radio-over-fiber wireless lans," in *IEEE LANMAN (accepted)*, 2011.

[20] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *MOBICOM*, 2003, pp. 66–80.

[21] J. M. Boyer and W. J. Myrvold, "On the cutting edge: Simplified o(n) planarity by edge addition," *Journal of Graph Algorithms and Applications*, vol. 8, no. 3, pp. 241–273, 2004.

[22] T. Hubai, "The chromatic polynomial," Master's thesis, Eötvös Loránd University, 2009.

[23] L. Faria, C. M. H. de Figueiredo, S. Gravier, C. F. Mendona, and J. Stolfi, "Nonplanar vertex deletion: maximum degree thresholds for np/max snp-hardness and a -approximation for finding maximum planar induced subgraphs," *Electronic Notes in Discrete Mathematics*, vol. 18, pp. 121 – 126, 2004.

[24] S. J. Louis and G. J. E. Rawlins, "Predicting convergence time for genetic algorithms," in *Foundations of Genetic Algorithms 2*, 1992, pp. 141–161.

[25] "The netwrok simulator ns-2," `http://www.isi.edu/nsnam/ns/`.

[26] "Multi-channel multi-interface simulation in ns2," `http://www.cse.msu.edu/ wangbo1/ns2/nshowto8.html`.