

# Channel Assignment with Partially Overlapping Channels in Wireless Mesh Networks

Yong Ding, Yi Huang, Guokai Zeng, Li Xiao  
Department of CSE, Michigan State University  
{dingyong, huangyi7, zengguok, lxiao}@cse.msu.edu

## ABSTRACT

Many efforts have been devoted to maximizing the network throughput with limited channel resources in multi-radio multi-channel wireless mesh networks. It has been believed that the limited spectrum resource can be fully exploited by utilizing partially overlapping channels in addition to non-overlapping channels in 802.11b/g networks. However, there are only few studies of channel assignment algorithms for partially overlapping channels. In this paper, an extension to the traditional conflict graph model, weighted conflict graph, is proposed to model the interference between wireless links more accurately. Based on this model, we first present a greedy algorithm for partially overlapping channel assignment, and then propose a novel genetic algorithm, which has the potential to obtain better solutions. Through evaluation, we demonstrate that the network performance can be dramatically improved by properly utilizing the partially overlapping channels. In addition, the genetic algorithm outperforms the greedy algorithm in mitigating the interference within the network and therefore leads to higher network throughput.

## Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design—*Wireless communication*

## General Terms

Algorithms, Performance

## Keywords

Wireless Mesh Networks, Partially Overlapping Channel, Channel Assignment, Genetic Algorithm.

## 1. INTRODUCTION

Wireless mesh networks (WMNs) consist of mesh routers and mesh clients, where mesh routers have minimal mobility and form the backbone of the network [1]. Recently,

there has been growing interest in using WMNs to extend or enhance Internet connectivity on the last mile. Some commercial deployments are already working to provide low cost connectivity to residents and local businesses.

One major problem facing WMNs is the capacity reduction caused by the interference among multiple simultaneous transmissions [2]. When two nearby wireless links communicate on the same frequency band, they cannot transmit data simultaneously. As a result, the throughput of each link may be decreased dramatically due to the interference from the other link. Also, a router cannot transmit and receive simultaneously with a single radio. To alleviate these problems, in many WMNs, mesh routers are equipped with multiple radios, which can be configured to operate on different channels. Thus, nodes are able to transmit and receive simultaneously in multi-radio multi-channel WMNs.

It is known that 802.11b/g and 802.11a provide 3 and 12 non-overlapping channels respectively. Although 802.11a provides more channel resources than 802.11b/g, it has several drawbacks. As 802.11a works on higher frequency spectrum (5GHz) than 802.11b/g (2.4GHz), it is more difficult to penetrate walls and other obstructions, and thus has a shorter range. In addition, 5GHz belongs to the regulated frequency spectrum, which makes 802.11a more expensive to operate. As a result, 802.11b/g is more commonly used.

To improve the throughput of WMNs, much research [3-10] has been done on configuring the network interfaces of mesh routers with different non-overlapping channels to avoid interference. However, due to the limited number of channels available, the interference cannot be completely eliminated. This is especially true in the case of 802.11b/g, which provides only three non-overlapping channels.

802.11b/g provides 14 channels, of which only the first 11 channels are permitted in US. According to 802.11b/g, if the channel separation is greater than 4, the two channels are non-overlapping channels (or orthogonal channels). Otherwise, they are partially overlapping channels. Thus, the number of non-overlapping usable channels is at most three (channel 1, 6 and 11). Previous algorithms [3-10] only consider the non-overlapping channels in the channel assignment. A simplified interference model is usually assumed, that is, if two links are within interference range of each other (twice the transmission range  $R$ ), they can transmit and receive simultaneously only if they use different non-overlapping channels. As a result, the frequency spectrum has not been fully exploited in these cases.

In this paper, we will study how to further mitigate the effects of interference in 802.11b/g mesh networks by fully

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WICON '08, November 17-19, 2008, Maui, Hawaii, USA.  
Copyright 2008 ACM 978-963-9799-36-3 ...\$5.00.

exploiting the spectrum resource, that is, utilizing both non-overlapping channels and partially overlapping channels, and efficient channel assignment algorithms. The contributions of this paper are three-fold.

- We make an extension to the traditional conflict graph model, weighted conflict graph, which can model the interference between wireless links more accurately with both non-overlapping and partially overlapping channels. The new model is based on our experimental results of interference measurement in a test-bed.
- We present a greedy channel assignment algorithm based on the weighted conflict graph in 802.11b/g mesh networks, which fully utilizes the spectrum resource. From simulations, we conclude that the network throughput can be dramatically improved by utilizing partially overlapping channels as well as non-overlapping channels.
- Besides the greedy algorithm, we propose using genetic algorithms to solve the channel assignment problem. To the best of our knowledge, no previous research has focused on using genetic algorithms for this kind of problem. In this paper, we discuss on the proper design of the genetic algorithm for channel assignment, and demonstrate that it can obtain better results than the greedy algorithm based on simulation results.

The rest of the paper is organized as follows. We summarize previous work in Section II. Section III presents the network model, including the derivation of the new weighted conflict graph model and a formulation of the channel assignment problem. In Section IV, we propose two channel assignment algorithms, the greedy algorithm and the genetic algorithm, and discuss on the design details. The performance evaluation is shown in Section V. Finally, we conclude our work in Section VI.

## 2. RELATED WORK

A major problem facing multi-hop wireless networks is the interference between adjacent links. The throughput of a single-radio single-channel wireless network has been studied in [2]. The authors formalized it as a multi-commodity flow problem with constraints from conflict graph, which is NP hard, and gave an upper bound and a lower bound of the problem.

There have been many studies on how to assign limited channels to network interfaces in a multi-radio multi-channel wireless mesh network so as to minimize interference and maximize throughput. They differ in several assumptions made in WMNs, and therefore in the models and related solutions.

One approach assumes a known traffic profile in the network, because the aggregate traffic load of each mesh router changes infrequently. The authors of [3] proposed an iterative approach to solve the joint routing and channel assignment problem, which can calculate a routing scheme as well as a channel assignment scheme, such that all traffic profiles can be satisfied. The problem has been formulated in [4] and [5] by using linear programming with constraints on interference and fairness, which is NP hard. The authors proposed approximation algorithms to get a joint routing and channel assignment scheme.

Other studies assume that the traffic profile of each mesh router is not known, and usually consider channel assignment and routing separately. The authors of [6] assumed that the traffic from the Internet gateway to clients is dominant, and thus they first constructed a load-balanced routing tree from the original network topology, and then proposed a distributed load-aware algorithm to assign channels to the links on the tree. In [7], the peer-to-peer traffic was assumed to be dominant in the network. The authors first constructed a  $k$ -connected backbone from the original network topology, and then assigned channels on the constructed topology. There have also been some heuristic channel assignment algorithms proposed in [8] [9] to minimize the interference in the wireless mesh network when the backbone topology is already determined.

Besides these static channel assignment algorithms, which assign channels to interfaces without change for a long time, there have been several dynamic channel allocation algorithms proposed, which allow interfaces to switch channels frequently. The authors of [10] proposed an on-demand channel allocation protocol in a wireless mesh network, where each node has two interfaces. In their framework, one interface of each node is devoted to controlling channel negotiation only while the other interface is used for data transmission. On the other hand, the frameworks proposed in [11] and [12] do not require a separate control interface, and the channel negotiation happens on the same interface for data transmission. As the overhead of dynamic channel switching cannot be neglected, static channel allocation strategies are more widely used in a static wireless mesh network. Thus, in this paper, we focus on static channel assignment algorithms.

Previous channel assignment algorithms are based on non-overlapping channels. The benefit of using partially overlapping channels in WLANs has been studied in [13], [14] and [15]. In [13], the authors measured the interference between different APs when partially overlapping channels are used. They proposed channel assignment algorithms with partially overlapping channels for APs in [14] and [15], which aim at minimizing the interference between different WLANs. A similar problem has also been studied in cellular networks [16]. The authors addressed the channel assignment problem of minimizing interference between same channels and adjacent channels. Different from these previous studies, we focus on utilizing partially overlapping channels to improve network throughput in WMNs, and study efficient channel assignment algorithms, which fully exploit the channel resources.

There have been several well-known test beds of WMNs. The MIT RoofNet [17] is a well-known test bed for wireless mesh networks built on the MIT campus. Microsoft has also constructed a test bed in their office building [18]. In these test beds, the wireless interference has been studied only on same channels. In this paper, we study the interference between partially overlapping channels.

The throughput of WMNs can be improved by using directional antennas. In [19], the authors proposed using directional antennas to establish point-to-point links. They observed that even in the presence of side-lobes, it is possible to transmit (receive) along all links of a node simultaneously under the same channel. They further considered multi-radio multi-channel WMNs using direction antennas, and proposed algorithms for channel assignment, which well

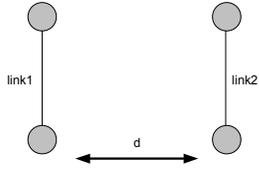


Figure 1: Experiment Setup

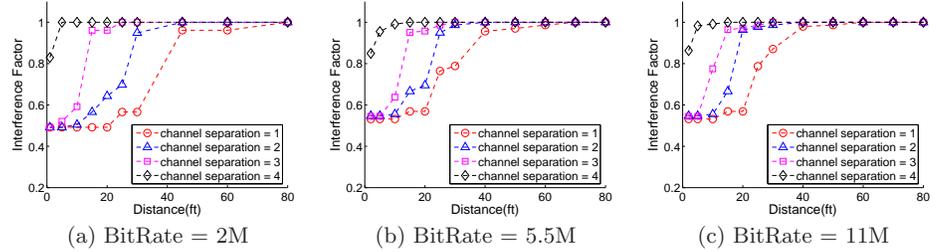


Figure 2: Interference with regard to Physical Distance and Channel Separation

utilize the special properties of directional antennas [20]. In this paper, we focus on omni-directional antenna WMN, which is more commonly used due to its low cost.

There have been many studies on routing metrics in WMNs. In [21], the authors provided a comparison of different routing metrics for static multi-hop wireless networks. WCETT (Weighted Cumulative ETT) has been proposed in [22], which has been shown to be more efficient than other traditional metrics in multi-radio multi-channel WMNs. Our work is focused on the channel assignment problem, which is below the routing layer in the network stack.

### 3. NETWORK MODEL

In this section, we first experimentally study the interference between partially overlapping channels. A similar experiment has been done in [13], which studied the interference of wireless links between mobile stations and APs. In this experiment, we study the interference between peer-to-peer wireless links. The focus of this section is to propose the weighted conflict graph model instead of the traditional conflict graph model, which can model the wireless interference between partially overlapping channels more accurately. We formulate the channel assignment problem based on the revised interference model in the end of this section.

#### 3.1 Interference Measurement

Fig.1 illustrates our experimental setup. We used four laptops, each equipped with a Netgear WAG511 802.11a/b/g PC Card. Linux kernel with Madwifi is used to drive the network cards. Paired laptops form a communication link, that is, the two end nodes of the link work on the same channel. We configured the two links with different channels and varied the distance between them. In this experiment, we aim to measure the level of interference between links configured with partially overlapping channels in 802.11b/g.

We used the following metric to evaluate the effect of interference. Let  $s_1$  and  $s_2$  be the throughput of *link1* and *link2* respectively when the other link is turned down. Let  $s'_1$  and  $s'_2$  be the throughput of *link1* and *link2* when both links are active. Then, the interference between these two links can be evaluated by *IF* (the Interference Factor).

$$IF = \frac{s'_1 + s'_2}{s_1 + s_2}$$

Therefore, if  $IF = 1$ , there is no interference between these two links, and they can transmit or receive simultaneously. When  $IF < 1$ , there exists interference between them. The lower *IF* value is, the higher the interference.

The experimental results are shown in Fig.2, where we can see the variance of interference factor with the physical distance and the channel separation of the two links. Fig.2(a), Fig.2(b) and Fig.2(c) illustrate the relationship when the bit rate of network cards are set to 2M, 5.5M, and 11M respectively. There are four possible values of channel separation for partially overlapping channels. We repeated our experiment for all possible pairs of partially overlapping channels. From the figure, we can observe that when the channel separation is fixed, interference decreases with increasing distance. When the distance is fixed, interference decreases with the increase of channel separation. Note that the results may vary slightly with different network cards. However, we observed the same trend.

#### 3.2 Weighted Conflict Graph

As shown in Fig.2, when the channel separation is fixed, the interference between two links can jump from severe interference (*IF* around 0.5) to almost no interference (*IF* around 1) with only a slight increase in distance. Thus, we can use the binary relationship to approximate the interference as follows. Let  $I_c$  be the interference range of two links with channel separation  $c$ . When the channel separation of the two links is  $c$ , they will interfere with each other if their distance is less than  $I_c$ , and otherwise not.

For example, when two links use the same channel, that is, their channel separation is 0, they will not interfere with each other as long as their distance is over  $2R$ , where  $R$  is the radio transmission range. When two links use non-overlapping channels, that is, their channel separation is 5, they will not interfere with each other no matter how close they are. Thus,  $I_0 = 2R$  and  $I_5 = 0$ , which is consistent with the traditional interference model.

The interference ranges for different channel separations under different bit rates are shown in Table.1. We use a

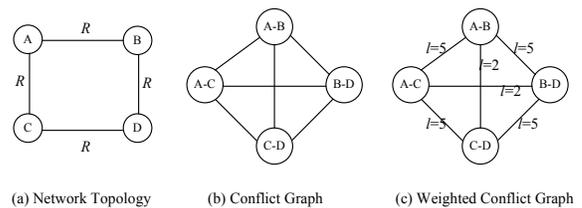


Figure 3: An Example of Weighted Conflict Graph

Table 1: Interference Range

	$I_0$	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$
2M	2R	1.125R	0.75R	0.375R	0.125R	0
5.5M	2R	R	0.625R	0.375R	0.125R	0
11M	2R	R	0.5R	0.375R	0.125R	0

threshold of 0.95, that is, for each channel separation, we find the minimum distance such that  $IF$  exceeds the threshold as the interference range. In our experiments, the radio transmission range  $R$  is 40ft. The reason we use binary approximation is that the optimal channel allocation problem with non-overlapping channels is already NP hard. By sacrificing a little accuracy, we are able to design more efficient algorithms to solve the channel allocation problem with partially overlapping channels.

Conflict graph has generally been used to model interference in previous work. Let  $G(V, E)$  represent a wireless mesh network with  $V$  denoting mesh routers and  $E$  denoting wireless links. A conflict graph  $F(S, T)$  of  $G(V, E)$  is defined as a graph that has each vertex  $s_i \in S$  corresponding to each link  $e_i \in E$ . When only non-overlapping channels are considered, the conflict graph  $F$  has an edge  $s_i s_j \in T$  if the distance between the two corresponding links  $e_i, e_j \in E$  is within interference range of each other (usually  $2R$ ). The distance between two links is defined as the minimum distance between any node of one link and any node of the other link. However, the traditional conflict graph does not model the interference between partially overlapping channels.

A Weighted Conflict Graph of  $G$  is denoted as  $(F(S, T), l)$ , where  $F$  is the conflict graph of  $G$ , and  $l$  is a label on  $T$ . Let  $s_i, s_j \in S$ , whose corresponding links  $e_i, e_j$  are within  $2R$  away. Then, we have  $s_i s_j \in T$ . Denote the distance between these two links by  $d(s_i s_j)$ . The label  $l$  is defined as:

$$l(s_i s_j) = \min\{c \mid d(s_i s_j) \geq I_c\}$$

$l(s_i s_j)$  actually indicates the minimum channel separation that links  $e_i$  and  $e_j$  must have so that they will not interfere with each other.

An example of the Weighted Conflict Graph is shown in Fig.3. Fig.3(a) shows a simple example of network topology, where  $R$  is the radio transmission range. Fig.3(b) illustrates the conflict graph. Assume the bit rate we are using is 2M. The weighted conflict graph is shown in Fig.3(c). As link  $A - B$  and  $A - C$  share a common node, they must use non-overlapping channels in order to avoid interference. In other words, their channel separation should be no less than 5,  $l(V_{A-B}V_{A-C}) = 5$ . On the other hand, the distance between links  $A - B$  and  $C - D$  is  $R$ . According to Table.1, as long as their channel separation is greater than or equal to 2, they will not interfere with each other. Thus, we have  $l(V_{A-B}V_{C-D}) = 2$ .

### 3.3 Problem Formulation

Since we are focusing here on channel assignment algorithms, we assume that the network topology has been determined through careful planning or by some topology control algorithms beforehand such as [6] and [7]. We abstract the mesh network topology as a graph  $G(V, E)$ , where  $V$  represents mesh routers, and  $E$  represents wireless links. Each pair of mesh routers of a link has a separate interface de-

voted to constructing the link. Similar to [7], we assume the wireless mesh network has dynamic traffic, that is, the connection demands have random sources, destinations and arrival times. This is because there will be substantial random and unpredictable traffic within the mesh network caused by peer-to-peer and newly emerging applications in addition to the traffic from and towards the Internet.

If there are enough channel resources, the problem becomes assigning the vertices of the weighted conflict graph with channels (or colors) while satisfying that the distance between the channels of adjacent vertices is no less than the weight on the edge between them, such that the span between the minimum and maximum channel used is minimized. This problem can be modeled as T-Coloring problem [23], which is NP-hard. In reality, the channel resource that we can use is usually limited, so our goal becomes minimizing the interference in the network with limited channel resource. The channel assignment problem can be formulated as follows.

Let  $(F(S, T), l)$  be the weighted conflict graph of  $G(V, E)$  and  $C$  be the set of channels. We define a label  $A$  on  $S$ ,  $A(s_i) \in C$  is the channel on which link  $s_i \in S$  is working. We also call  $A$  as a channel assignment scheme for the wireless mesh network.

Let  $I(s_i, s_j, A(s_i), A(s_j))$  be the interference indicator between links  $s_i, s_j \in S$ , that is, it indicates whether these two links will interfere with each other under channel assignment  $A$ . This can be calculated by Algorithm 1 based on the weighted conflict graph.

---

#### Algorithm 1 $I(s_i, s_j, A(s_i), A(s_j))$

---

```

1: if  $s_i s_j$  is not an edge in  $F$  then
2:    $I = 0$  // no interference
3: else
4:   if  $|A(s_i) - A(s_j)| \geq l(s_i s_j)$  then
5:      $I = 0$ 
6:   else
7:      $I = 1$  // interfere with each other
8:   end if
9: end if

```

---

For the simplicity of presentation, we define the following two objective functions:

$$H_1(\langle F, l \rangle, A) = \sum_i \sum_{j \neq i} I(s_i, s_j, A(s_i), A(s_j)) / 2 \quad (1)$$

$$H_2(\langle F, l \rangle, A) = \max_i \sum_{j \neq i} I(s_i, s_j, A(s_i), A(s_j)) \quad (2)$$

$H_1$  defines the total interference within the network, that is, the total number of link pairs that interfere with each other, while  $H_2$  denotes the maximum link interference. Therefore, we are trying to find the channel assignment  $A$  that minimizes  $H_1$  or minimizes  $H_2$ . These problems are NP hard, because the graph coloring problem is an NP complete problem. In the next section, we will solve for approximate solutions with objective  $H_1$ . The proposed algorithms can be easily modified for objective  $H_2$ .

## 4. CHANNEL ASSIGNMENT

In this section, we propose two algorithms, the greedy algorithm and the genetic algorithm, for channel assignment using partially overlapping channels. By presenting

the greedy algorithm, we want to show how the network interference can be further minimized by using partially overlapping channels, because previous studies, such as [6] [7] [8], also use greedy strategies for channel allocation with non-overlapping channels. Genetic algorithms have not been used to solve the channel assignment problems before, yet it has the potential to generate better results than the greedy algorithm.

## 4.1 Greedy Algorithm

Generally speaking, our greedy algorithm is a series of decisions, each of which assigns a channel to a link, until all links have been assigned channels. Each decision is usually composed of two steps – *select* and *assign*. In the *select* step, a link that has not been assigned a channel is chosen according to metric  $\alpha$ , and in the *assign* step, a proper channel is assigned to the selected link according to metric  $\beta$ . In each step, the link and its channel selection are determined by maximizing (or minimizing) their corresponding metrics  $\alpha$  and  $\beta$ .

We define the metric  $\alpha$  of a link as the expected level of interference between this link and all the other links in the network. As during the greedy channel assignment process, some links may not have been assigned channels yet, we use the expected value to evaluate the interference when selecting a link. Given the weighted conflict graph  $(F(S, T), l)$ , the expected interference of link  $s \in S$ , denoted by  $\alpha(s)$ , is computed as follows:

$$\alpha(s) = \sum_{s' \in S_1} \bar{I}_1(s, s') + \sum_{s' \in S_2} \bar{I}_2(s, s')$$

where  $S_1$  is the set of links that have already been assigned channels, and  $S_2$  is the set of links not assigned channels yet.  $\bar{I}_1(s, s')$  (or  $\bar{I}_2(s, s')$ ) denotes the expected interference between  $s$  and  $s'$ , which has been assigned (or not assigned) a channel. They are calculated in the following ways:

$$\bar{I}_1(s, s') = \frac{1}{|C|} \sum_{i \in C} I(s, s', i, A(s'))$$

$$\bar{I}_2(s, s') = \frac{1}{|C|^2} \sum_{i, j \in C} I(s, s', i, j)$$

Therefore, in each step, we select the link  $s$  that has the minimum expected interference  $\alpha(s)$ .

In the *assign* step, we define the metric  $\beta(c)$  for each candidate channel  $c$  that can be assigned to the selected link.  $\beta(c)$  indicates the interference between the selected link and those links already assigned channels.

$$\beta(c) = \sum_{s' \in S_1} I(s, s', c, A(s'))$$

We select the channel  $c$  that has the minimum  $\beta(c)$ , thus minimizing the interference added to the network when we assign channels to the selected link.

As described in Algorithm 2, given the weighted conflict graph  $(F(S, T), l)$  and the channel set  $C$ , the greedy algorithm obtains a channel assignment scheme  $A$ . This algorithm is able to find a solution very fast because it never changes a link's channel once it is assigned. Next, we will present a genetic algorithm, which has the potential to obtain near-optimal results.

---

## Algorithm 2 Greedy-Algorithm( $(F(S, T), l), C$ )

---

```

1:  $A(s) = null$  for all  $s \in S$ 
2:  $S_1 = \{\}$ ,  $S_2 = S$ 
3: while  $S_2 \neq \{\}$  do
4:   Calculate  $\alpha(s)$  for each link  $s \in S_2$ 
5:   Select  $s^*$  such that  $\alpha(s^*) = \min_{s \in S_2} \alpha(s)$ 
6:   For link  $s^*$ , calculate  $\beta(c)$  for each channel  $c \in C$ 
7:   Select  $c^*$  such that  $\beta(c^*) = \min_{c \in C} \beta(c)$ 
8:    $A(s^*) = c^*$ 
9:    $S_1 = S_1 \cup \{s^*\}$ ,  $S_2 = S_2 - \{s^*\}$ 
10: end while

```

---

## 4.2 Genetic Algorithm

Genetic algorithms are adaptive heuristic search algorithms premised on the ideas of natural selection and genetic evolution. Its basic concept is to simulate the process of evolution in the natural system, during which fitter individuals are more likely to survive to the next generation.

The genetic algorithm is applicable to the channel assignment problem because of the following reasons: 1) The channel assignment problem has an inherent *local optimization* property. In other words, a good channel assignment scheme for a subnetwork that causes less interference locally is more likely to be contained in a good channel assignment scheme for the entire network that causes less interference globally. This property fits well into the genetic algorithm. In the genetic algorithm for channel assignment, we regard a channel assignment scheme for a subnetwork as local DNAs, and regard a channel assignment scheme for the entire network as an individual. Thus, we improve local DNAs to form better individuals in the genetic algorithm. 2) In the genetic algorithm, individuals from the current generation are selected to breed the next generation, which is expected to have fitter individuals with high probabilities. This *fitness preservation* property exists in the channel assignment problem. We observe that better channel assignment schemes are more likely to be created by combining the good “parts” (channel assignment for subnetworks) of the channel assignment schemes in the current generation.

### 4.2.1 Problem Mapping

In order to solve the channel assignment problem by genetic algorithms, the first step is to establish a mapping between them. In our channel assignment problem, we denote a channel assignment for a single link as a **DNA**, and a channel assignment scheme for all links of the entire network as an **individual**, which is as well a solution to the channel assignment problem. In the genetic algorithm, a generation is composed of a set of individuals. Thus, we define a **generation** as a set of channel assignment schemes (a set of solutions) in our problem. An example of the problem mapping is illustrated in Fig.4. The input of the problem is a network topology with 4 nodes and 4 links, where each node has two interfaces. The numbers beside the links represent channels. The channel assignment of each link is considered as a DNA. The channel assignment scheme shown in the top figure is considered as an individual, and the set of channel assignment schemes shown in the bottom figure constitutes a generation.

Each channel assignment scheme is encoded as a binary string in the genetic algorithm. The transformation is in the following way.

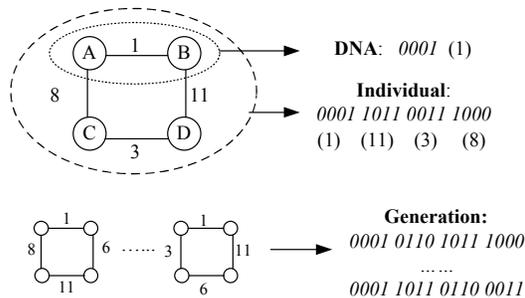


Figure 4: Problem Mapping

1. Sort the links in linear order;
2. Convert the channel assigned to each link into a binary string with a fixed length (a DNA);
3. Concatenate all the binary strings one by one into a single binary string (an individual) according to the order of the links.

For example, Fig.4 shows the binary representation of each individual. The links are sorted in the order of  $AB$ ,  $BD$ ,  $CD$ ,  $AC$ . The channel assigned to each link is encoded into a string with a fixed length of 4.

The order of the links has a great impact on the performance of the genetic algorithm. Intuitively, we prefer the links that are within interference range of each other to be close together in the sorted list. As a result, when we combine “parts” of individuals to create offsprings, the local optimization property of the parent individuals (good channel assignment for subnetworks) can be preserved in the next generation. There are several strategies that can be used for link ordering: i) random order, ii) breadth-first search ( $BFS$ ), iii) depth-first search ( $DFS$ ). In our genetic algorithm for channel assignment, we find that  $BFS$  and  $DFS$  give better performance (shown in Section V).

A genetic algorithm is equipped with a *fitness function*, which is used to evaluate the fitness of an individual. The fitter an individual is, the better the solution. Given a binary string representing a channel assignment scheme for a network, we define its fitness function  $f$  as the negative of the total interference within the network if the channel assignment scheme is applied, which is also the objective function of the optimization problem in Section III.

#### 4.2.2 Algorithm Design

Our genetic algorithm begins with an initial generation, which is composed of  $N$  randomly generated channel assignment schemes. Each next generation is generated by a series of *selection-reproduction* steps from the current generation. In the *selection* step, two individuals in the current generation are picked up. Two constraints must be satisfied: i) Fitter individuals are more likely to be selected. ii) The other individuals also have a proper possibility to be selected, which guarantees an extent of diversity. The maintenance of diversity plays an important role, because it can avoid early convergence on local optima. In the *reproduction* step, the two selected individuals breed two new individuals through *crossover* and *mutation*. *Crossover* is a

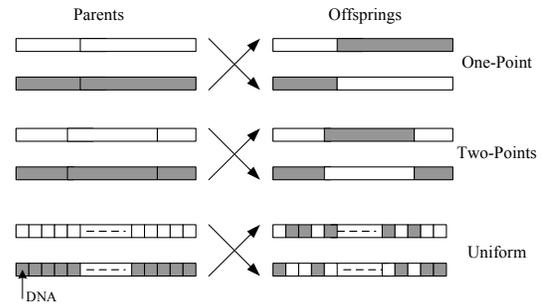


Figure 5: Crossover Strategies

strategy to combine parts of parent individuals to produce new individuals and *mutation* further changes part of a new individual with a small probability, which is called mutation rate, to enhance diversity.

---

#### Algorithm 3 Genetic-Algorithm( $(F, l), C$ )

---

- 1: Randomly generate  $N$  channel assignment schemes,  $P$
  - 2: Calculate  $f$  value for each scheme in  $P$
  - 3: **while**  $M$  loops are not completed **do**
  - 4:   **while**  $N/2$  loops are not completed **do**
  - 5:     Select  $p_1, p_2$  from  $P$  using *selection* strategy
  - 6:     Create  $o_1, o_2$  from  $p_1, p_2$  using *reproduction* strategy
  - 7:     Calculate  $f(o_1), f(o_2)$
  - 8:     Find two channel assignment schemes  $b_1, b_2$  that have the lowest  $f$  values in  $P$
  - 9:     Replace  $b_1, b_2$  with  $o_1, o_2$
  - 10:   **end while**
  - 11: **end while**
  - 12: Return the fittest channel assignment scheme in  $P$
- 

Our genetic algorithm for solving the channel assignment problem is depicted in Algorithm 3, where  $N$  denotes the initial population size (the number of individuals in the first generation) and  $M$  is the number of generations to evolve. The algorithm has polynomial running time, because there are  $MN/2$  steps, each of which requires polynomial time.

- *Selection* strategy: As the channel assignment problem has many local optima, we adopt the stochastic selection, which is good at maintaining population diversity. This strategy exploits two well-studied selection methods, the roulette wheel selection and the tournament selection. In roulette wheel selection, each individual is selected with a probability proportional to its fitness function. Individuals with higher fitness are more likely to be selected, while individuals with low fitness still has a chance to be selected. In tournament selection,  $n$  individuals are selected at random and the fittest one of them is chosen. In stochastic selection, we first selects 2 individuals independently by using roulette wheel selection, and then choose the better one of them according to tournament selection.
- *Reproduction* strategy: It includes *crossover* and *mutation*. When two individuals are selected to breed offsprings, both are split into several parts by cutting at

some random points. Offsprings are produced by combining different parts from the two parent individuals. With respect to the number of cutting points, well-known crossover methods include one-point crossover, two-points crossover and uniform crossover. In uniform crossover, each DNA is swapped between parent individuals with a fixed probability, typically 0.5. Fig.5 illustrates different crossover methods. In the figure, two parents are drawn in different colors, and offsprings contain different parts from parents. It has been discovered that the uniform crossover achieves better performance in most optimization problems. However, we find that, with respect to our channel assignment problem, the one-point crossover and two-point crossover work better (shown in Section V). This is due to the local optimization property of the channel assignment problem. Uniform crossover disperses the DNAs from parent individuals into the new individual, and thus does not preserve the good channel assignment schemes for local subnetworks in the new individual. To determine the proper mutation rate, it has been recommended that mutation rate be differed by an order of magnitude of 0.001 on binary-encoded continuous-valued optimization problems, and the mutation rate be in the range [0.005, 0.01].

- In addition, we have also used some other important techniques to further improve the performance, including *elitism*, *sharing* and *niching*. For a complete discussion on genetic algorithms, please see [24].

## 5. PERFORMANCE EVALUATION

In this section, we will evaluate both the greedy algorithm and the genetic algorithm using all channels (orthogonal channels and partially overlapping channels), and compare them with the greedy algorithm using orthogonal channels only. We compare these algorithms based on the following metrics. The value of the “objective function” is a simple metric to evaluate an algorithm’s capability of solving for optimal solutions. However, as this is not a direct metric for network performance, we further compare these algorithms by “average link bandwidth” and “network throughput”.

Given the channel assignment schemes computed by different algorithms, we evaluated the network performance in NS-2.31. The Hyacinth extension [25] was used to support multiple channels and multiple interfaces per node in the simulator. We made some further extensions to support partially overlapping channels in addition to non-overlapping channels. Assume each node has equal possibility to send out data, and thus we can measure the bandwidth of each wireless link in the network. If the network is without any interference, the bandwidth of each link should be its bit rate. However, with limited channel resources, the network interference cannot be completely eliminated, which causes the link bandwidth to decrease dramatically. To evaluate the network throughput, we assume dynamic traffic in the wireless mesh network, that is, the connection demands have random sources and destinations. We impose a certain number of *UDP* flows concurrently on the network, and measure the network throughput by the total throughput of all the flows. The packet size is set to 512.

Through simulations, we will analyze the performance improvement we can achieve by using all channel resources as

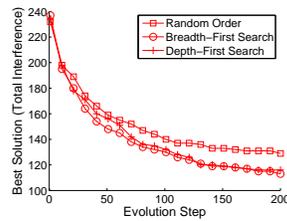


Figure 6: Impact of Edge Ordering

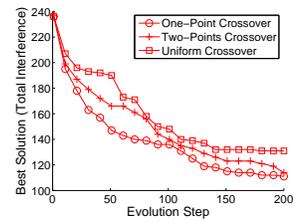


Figure 7: Impact of Crossover Strategy

compared with only using orthogonal channels in the channel assignment, and also demonstrate how the genetic algorithm can further improve network performance.

### 5.1 Impact of Parameters on Genetic Algorithm

Before presenting the simulation results, we first analyze several parameters that may influence the performance of the genetic channel assignment algorithm.

In Section 4.2.1, we introduced three edge ordering strategies; random order, *BFS* and *DFS*. In order to analyze which strategy is better, we run the genetic algorithm with different edge orders on random graphs with different sizes. Fig.6 shows the result for a random graph with 50 nodes. The x-axis denotes the number of generation in the evolution process, and the y-axis shows the best solution of each generation (measured by the total network interference). Each point corresponds to the average of 20 runs. As shown in the figure, at the first several generations, the three strategies have the similar performance. However, with the evolution of generations, the genetic algorithm with *BFS* or *DFS* edge order converges faster (getting better results in each evolution) than random order. We observe the similar phenomenon in networks of other sizes. This demonstrates that *BFS* and *DFS* edge ordering have better performance than random edge ordering. For the evaluation in the rest of this paper, we use *BFS* for edge ordering in our genetic algorithm.

Fig.5 illustrates the impact of different crossover strategies on the genetic channel assignment algorithm. We run the genetic algorithm with one-point, two-point and uniform crossover strategies on random graphs with 50 nodes respectively. From the figure, we can observe that the algorithm with one-point crossover has the fastest converging speed, while the uniform crossover has the lowest converging speed. For the evaluation in the rest of this paper, we use one-point crossover strategy in our genetic algorithm.

### 5.2 Minimizing Interference Under Regular Topologies

In this section, we study the scenario with regular topologies, because wireless mesh networks are usually deployed after careful planning, and the regular topology can average the performance over the whole network. We use the topology of  $N \times N$  squared grids with side length of  $R$  (the radio transmission range), that is, each vertex is deployed with a mesh router, and each edge denotes a wireless link. Each inside mesh router is equipped with 4 network interfaces that communicate with its 4 neighbors independently, while each boundary router is equipped with 3 or 2 interfaces depending on its number of neighbors.

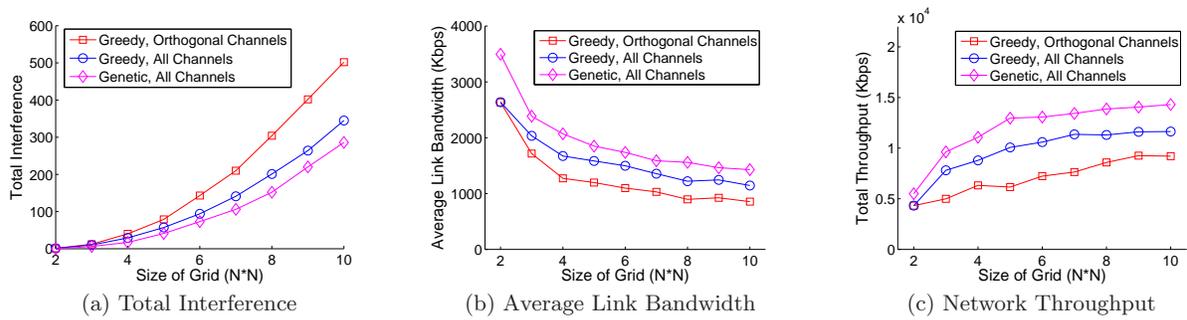


Figure 8: Evaluation of Channel Assignment Algorithms under Regular Topology

### 5.2.1 Using Partially Overlapping Channels

We use 802.11b with bit rate of 11Mbps. In the channel assignment with only orthogonal channels, channels 1, 6, 11 are used. In our approach, we use all the channels from 1 through 11. As our purpose is to analyze the effect of using partially overlapping channels on mitigating interference, we select the greedy algorithm for evaluation. That is, we run the greedy algorithm based on the conflict graph model (using only orthogonal channels) and weighted conflict graph model (using all the channels).

Fig.8(a) (the top two curves) shows the total interference within networks of different sizes for these two approaches. We can observe that the network interference can be further mitigated by using partially overlapping channels. With the increase of network size, the total interference for the channel assignment with all channels is roughly  $\frac{2}{3}$  of the channel assignment with only orthogonal channels. Fig.8(b) (the bottom two curves) compares the average link bandwidth of the network. By using partially overlapping channels, the average link bandwidth can be improved by around 32 percent when the network becomes large, which implies that the network can carry more traffic. Fig.8(c) (the bottom two curves) illustrates the network throughput under both approaches. If we fully exploit the channel resources, we can achieve an improvement of network throughput by approximately 42 percent.

### 5.2.2 Using Genetic Algorithm

We also proposed a potentially more efficient algorithm for channel assignment, the genetic algorithm. We will evaluate this algorithm by comparing it with the greedy algorithm. We assume that all the channels are used in the algorithms. The same metrics from the previous section are used in the evaluation. For the genetic algorithm, we set the population size to 5000, the mutation rate to 0.005, and the number of generations to 500.

Fig.8(a) (the bottom two curves) shows the total interference within the network when both channel assignment algorithms are applied. The genetic algorithm obtains better results than the greedy algorithm under different network scales. As in the previous experiment, Fig.8(b) and Fig.8(c) (the top two curves) compare the average link bandwidth and the network throughput. These two figures illustrate a similar trend. By using the channel assignment scheme computed from the genetic algorithm, the average link bandwidth and the network throughput can be improved by ap-

proximately 20 percent and 24 percent respectively, compared to the greedy algorithm.

## 5.3 Minimizing Interference Under Random Topologies

We use the topology generation tool *GT-ITM* [26] to generate random topologies. We choose the Locality Model to generate flat random graphs with uniformly distributed nodes. By properly setting the parameters such as the number of nodes, the separating length and the connectivity probability, we can obtain graphs with desired scales and average degrees.

We generate random graphs of different scales but with the same average degree ( $=3$ ). The minimum degree is no less than 2 and the maximum degree is no greater than 5. Three channel assignment algorithms are executed on these random topologies, that is, the greedy algorithm using only orthogonal channels, the greedy algorithm using all the channels, and the genetic algorithm using all the channels. The simulation results are shown in Fig.9.

In the figures, each point corresponds to the average of running each algorithm on 20 random graphs. We get similar results with the regular topology cases, although the performance improvement is not that dramatic. The network throughput can be improved by approximately 25 percent if we fully exploit the spectrum resource. Moreover, the throughput can be further improved by 15 percent using the genetic algorithm instead of the greedy algorithm.

## 6. CONCLUSION

In this paper, we experimentally study the interference between overlapping channels, and propose the weighted conflict graph, which can model the interference more accurately compared to conflict graph. Based on this model, we propose two channel assignment algorithms, which utilize both orthogonal and partially overlapping channels in the channel allocation. The greedy algorithm is fast but may not get near-optimal results. The genetic algorithm has the potential to get better results with polynomial running time. Our simulations demonstrate that the network throughput can be dramatically improved by using the partially overlapping channels in addition to the orthogonal channels in the channel assignment. Moreover, the genetic algorithm can generate better channel assignment schemes than the greedy algorithm. Therefore, it can further reduce the interference and improve the network capacity.

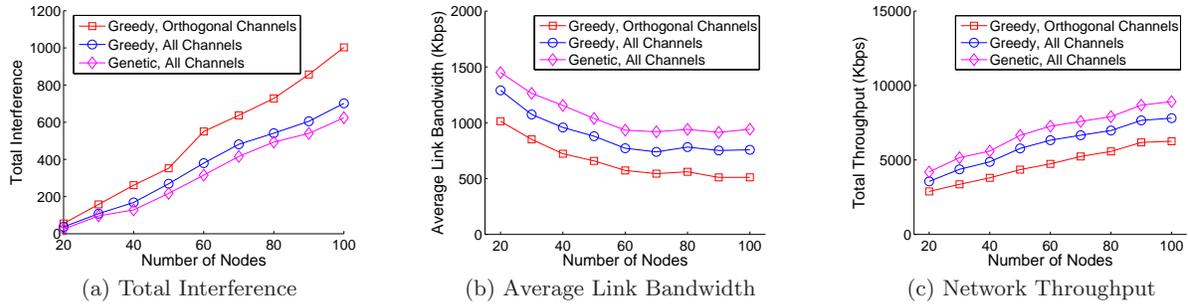


Figure 9: Evaluation of Channel Assignment Algorithms under Random Topology

## 7. ACKNOWLEDGMENTS

This work was supported in part by the US National Science Foundation under grants CCF-0514078, CNS-0551464, and CNS-0721441.

## 8. REFERENCES

- [1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," in *Computer Networks*, 2005.
- [2] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *MobiCom*, 2003.
- [3] A. Raniwala, K. Gopalan, and T. cker Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," in *MC2R*, 2004.
- [4] M. Alicherry, R. Bhatia, and L. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," in *MobiCom*, 2005.
- [5] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," in *MobiCom*, 2005.
- [6] A. Raniwala and T. cker Chiueh, "Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network," in *INFOCOM*, 2005.
- [7] J. Tang, G. Xue, and W. Zhang, "Interference-aware topology control and qos routing in multi-channel wireless mesh networks," in *MobiHoc*, 2005.
- [8] A. P. Subramaniam, H. Gupta, and S. R. Das, "Minimum-interference channel assignment in multi-radio wireless mesh networks," in *Technical Report*, 2006.
- [9] 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.
- [10] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu, "A new multi-channel mac protocol with on-demand channel assignment for multi-hop mobile ad hoc networks," in *ISPAN*, 2000.
- [11] J. So and N. Vaidya, "Multi-channel mac for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver," in *MobiHoc*, 2004.
- [12] P. Bahl, R. Chandra, and J. Dunagan, "Ssch: Slotted seeded channel hopping for capacity improvement in ieee 802.11 ad-hoc wireless networks," in *MobiCom*, 2004.
- [13] A. Mishra, E. Rozner, S. Banerjee, and W. Arbaugh, "Exploiting partially overlapping channels in wireless networks: Turning a peril into an advantage," in *Internet Measurement Conference*, 2005.
- [14] A. Mishra, S. Banerjee, and W. Arbaugh, "Weighted coloring based channel assignment for wlans," in *MC2R*, 2005.
- [15] A. Mishra, V. Shrivastava, S. Banerjee, and W. A. Arbaugh, "Partially overlapped channels not considered harmful," in *SIGMETRICS*, 2006.
- [16] W. Yue, K. Miyazaki, and X. Deng, "Optimal channel assignment in wireless communication networks with distance and frequency interferences," in *Computer Communications*, 2004.
- [17] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," in *SIGCOMM*, 2004.
- [18] J. Padhye, S. Agarwal, V. N. Padmanabhan, L. Qiu, A. Rao, and B. Zill, "Estimation of link interference in static multi-hop wireless networks," in *Internet Measurement Conference*, 2005.
- [19] B. Raman and K. Chebrolu, "Design and evaluation of a new mac protocol for long-distance 802.11 mesh networks," in *MobiCom*, 2005.
- [20] B. Raman, "Channel allocation in 802.11-based mesh networks," in *INFOCOM*, 2006.
- [21] R. Draves, J. Padhye, and B. Zill, "Comparison of routing metrics for static multi-hop wireless networks," in *SIGCOMM*, 2004.
- [22] R. Draves, J. Padhye and B. Zill, "Routing in multi-radio multi-hop wireless mesh networks," in *MobiCom*, 2004.
- [23] A. Graf, "Distance graphs and the t-coloring problem," in *Discrete Math*, 1999.
- [24] T. M. Michell, *Machine Learning*, N. Y. Mc Graw Hill, Ed., 1997.
- [25] "Hyacinth: An ieee 802.11-based multi-channel wireless mesh network, [http://www.ecsl.cs.sunysb.edu/multichannel/.](http://www.ecsl.cs.sunysb.edu/multichannel/)"
- [26] "[http://www.cc.gatech.edu/fac/ellen.zegura/graphs.html.](http://www.cc.gatech.edu/fac/ellen.zegura/graphs.html)"