Abstract—Wireless mesh networks are deployed as broadband backbones to provide ubiquitous wireless access for residents and local businesses. Utilizing multiple channels has the potential to scale up the system capacity of wireless access networks with delicately designed channel assignment algorithms. In this paper, we consider a static channel assignment in multi-radio multi-channel wireless mesh networks with the objective of maximizing overall end-to-end throughput. We first present an integer linear programming (ILP) optimization model for this static channel assignment problem. Then, by taking into account the “bottleneck links” of multi-hop flows, we propose a flow-aware heuristic scheme, which decomposes this ILP problem into a graph coloring subproblem and a linear programming subproblem. Simulation results on ring and grid topologies show that our scheme has significant gain in terms of network throughput.

I. INTRODUCTION

There is an increasing interest in deploying wireless mesh networks (WMNs) as broadband backbones to provide ubiquitous wireless access for residents and local businesses. In a WMN, a collection of stationary or minimally mobile wireless access routers are interconnected as a wireless multi-hop network. The emerging multi-radio/multi-channel enabled devices offer the possibility for utilizing multiple radios and channels to scale up the wireless network capacity [1], through delicate channel assignment.

Recently, numerous channel assignment schemes are proposed for multi-radio WMNs. According to the granularity of assignment, the existing schemes can be categorized into three groups: (i) Per-Packet Channel Assignment. In [2] and [3], channels are assigned for per outgoing packet, and the network interface cards switch channels in a small time scale. (ii) Link Based Channel Assignment. Channel is assigned to a link between two given neighboring nodes, and all packets between these nodes will be transmitted on this specific channel for a duration that the decision is valid for. In [4], [5], A. Raniwala et al. assign channels to links, aiming at providing each link the amount of bandwidth according to the expected load. In [6], the authors present an empirical algorithm to assign channels to links aiming at minimizing accumulated link interference in the mesh network. In [7], the authors address the spectrum management problem in open spectrum system. By extending greedy algorithms in graph coloring to color-sensitive graph, they proposed a heuristic approach to maximize the link based utility function. In [8], [9], several optimization models are also proposed for centralized channel assignment in static WMNs, which aim at either minimizing the total interferences among links [8], or maximizing the weighted sum of simultaneously active links. (iii) Flow/Component Based Channel Assignment. A single channel is assigned to several successive links which compose a flow/component. In [10], J. So et al. assign channels to flows along with the route establishment in single radio multi-channel network. This flow based scheme is extended in [11] to component based channel assignment. A component is formed by intersecting flows. An entire component is assigned a single channel, and different components may operate on different channels.

All these existing schemes leverage from the multiple channels by increasing spatial reuse. However, the per-packet channel assignments suffer from the switching delay and synchronization problems. It is also totally unaware of end-to-end flows. The flow/component based schemes are trade offs between spatial reuse and complexity. It is true that assigning channels to flows/components introduces fewer channel switching overloads and implementation cost, however, they are at the cost of decreasing the spatial reuse of multi-hop flows or components, especially in a dense network. In the link based channel assignments, maximizing the link based metric may not guarantee maximum end-to-end metric. A intuitive explanation is the “Barrel Theory”: a multi-hop flow’s bandwidth is only as much as its bottleneck link, which have the minimum bandwidth along this flow. To the best of our knowledge, none of the existing channel assignment schemes take into account the end-to-end throughput.

We first formulate the channel assignment problem in multi-radio multi-channel WMNs, which aims at maximizing the overall end-to-end throughput, as an Integer Linear Programming (ILP) optimization problem. Taking the bottleneck links into account, we propose a centralized flow-aware heuristic approximation algorithm for such a NP-hard problem. In our iterative algorithm, channels are assigned to the instantaneous bottleneck links of flows. Then for each link, by solving a Linear Programming (LP) problem, the link bandwidth is optimally allocated to a set of flows which are sharing this link. The key contributions of this paper are: (i) A generalized optimization model is presented for our end-to-end throughput-aware channel assignment in multi-radio multi-channel WMNs. (ii) This ILP problem is decomposed into

1This work has been supported by the National Basic Research Program of China (2007CB310607).
a graph coloring problem and a LP problem, which can be solved at an acceptable complexity.

The remainder of this paper is organized as follows. We discuss our system model and problem formulation in section II. Our solution is introduced in section III in detail. Section IV evaluates our approach in ring and grid topologies. Section V concludes this paper.

II. SYSTEM MODELS AND PROBLEM FORMULATION

We consider a static wireless mesh network with \( N \) links which are competing for \( M \) wireless channels. There are \( K \) multi-hop flows in the network. The sets of link, channel and flow indexes are \( \mathcal{N} = \{1, 2, \ldots, N\} \), \( \mathcal{M} = \{1, 2, \ldots, M\} \) and \( \mathcal{K} = \{1, 2, \ldots, K\} \) respectively. \( \mathcal{L}_i \) is used to denote a link and \( f_k \) for a flow. Each flow \( f_k \) is composed by a set of links that it passes through:

\[
f_k = \{ j \mid j \in \mathcal{N}, l_j \text{ is on the path of flow } k \}, \quad \forall k \in \mathcal{K}.
\]

In other word, flow \( f_k \) passes link \( l_i \) if and only if \( i \in f_k \). \( F_i \) denotes a set of flows which are sharing a common link \( l_i \):

\[
F_i = \{ k \mid i \in f_k \}, \quad \forall i \in \mathcal{N}.
\]

To clarify our presentation, the notations are summarized as follows:

- **Interference Constraint:** \( C = \{c_{i,j}\}_{N \times N} \), \( \forall c_{i,j} \in \{0, 1\} \), an \( N \) by \( N \) matrix representing the interference among links. If \( c_{i,j} = 1 \), link \( l_i \) and \( l_j \) interfere each other, thus not allowed to be assigned the same channel simultaneously.

- **Channel Reward:** \( B = \{b_{i,m}\}_{N \times M} \), an \( N \) by \( M \) matrix. An element \( b_{i,m} \) represents the bandwidth (or data rate) that can be achieved in conflict free condition by link \( l_i \) on channel \( m \). Different links may have different transmission rates on the same channel, and one link may have different transmission rates on different channels.

- **Feasible Channel Assignment:** \( A = \{a_{i,m}\}_{N \times M} \), \( \forall a_{i,m} \in \{0, 1\} \), an \( N \) by \( M \) matrix. If \( a_{i,m} = 1 \), channel \( m \) is assigned to link \( l_i \). Any feasible channel assignment must satisfy the interference constraint:

\[
a_{i,m} + a_{j,m} \leq 1 \quad \text{if} \quad c_{i,j} = 1, \quad \forall i, j \in \mathcal{N}, \forall m \in \mathcal{M}.
\]

That is

\[
a_{i,m} + a_{j,m} \leq 2 - c_{i,j}, \quad \forall i, j \in \mathcal{N}, \forall m \in \mathcal{M}.
\]

- **Link Rate:** \( \beta_i = \sum_{m=1}^{M} a_{i,m} b_{i,m} \) represents the total rate that link \( l_i \) gets from a given feasible channel assignment \( A \). If link \( l_i \) is shared by a set of flows \( (F_i) \), each flow passing this link gets a fraction of its rate.

- **Flow Rate:** \( \gamma_k \) represents the end-to-end throughput of a multi-hop flow \( f_k \). The total rate of the flows passing through a common link can not exceed the link’s rate.

\[
\sum_{k \in F_i} \gamma_k \leq \sum_{m=1}^{M} a_{i,m} b_{i,m}, \quad \forall i \in \mathcal{N}.
\]

In addition, We assume there are sufficient radios/network interface per node. Hence, radio limitation is not a constraint in this paper. Throughout this paper, we use variables \( i, j \) for link indexes, \( k \) for flow index, \( m \) for channel. Given notations above, the end-to-end throughput-aware channel assignment in WMNs can be formulated as the following ILP:

\[
\begin{align*}
\max \sum_{k=1}^{K} & \gamma_k \\
\text{s.t.} \quad & \sum_{k \in F_i} \gamma_k \leq \sum_{m=1}^{M} a_{i,m} b_{i,m}, \quad \forall i \in \mathcal{N}, \\
& a_{i,m} + a_{j,m} \leq 2 - c_{i,j}, \quad \forall i, j \in \mathcal{N}, j \neq i, \forall m \in \mathcal{M}, \\
& c_{i,j}, a_{i,m} \in \{0, 1\}, \quad \forall i, j \in \mathcal{N}, \forall m \in \mathcal{M}.
\end{align*}
\]

Similar ILP problem is proved to be NP-hard in past work [9].

III. FLOW-AWARE HEURISTIC APPROACH

We decompose the above ILP optimization problem into a graph coloring subproblem and a link rate allocation subproblem.

The WMN is represented by a bidirectional conflict graph \( G = (V, L, E) \), where \( V \) denotes a set of vertexes representing links in the network, \( L \) denotes the available color list at each vertex, and \( E \) is a set of edges representing interference between vertexes. Throughout the paper, we use node, link to refer to topology graph, vertex, edge to refer to conflict graph. Channel and color are used interchangeably. Several past work have demonstrated the effectiveness of using greedy algorithms in graph coloring to solve channel assignment problems. In [12], Ramanathans proposed a progressive minimum neighbor first (PMNF) solution for the unified time/frequency/code domain channel assignment problem. In PMNF, vertexes are first labelled, then colored. After a vertex is colored, this vertex and the edges incident on this vertex are deleted while processing the rest of vertexes. In PMNF, the objective is to minimize the total number of colors that needed, hence the most difficult vertex (vertex with most neighbors) is treated first. [7] extended this scheme to color-sensitive graph coloring (CSGC), where vertexes are connected via multiple colored edges and colors have different reward. When a color \( m \) is assigned to a vertex \( v \), \( m \) is deleted from the color lists of vertex \( v \) and its \( m \)-color neighbors. And only the \( m \)-color edges incident on \( v \) are deleted.

We extend the CSGC schemes in [7] to solve our graph coloring subproblem. Other than minimizing the number of colors needed [12], nor maximizing the link-based utility [7], we aim at end-to-end throughput. The colors are assigned iteratively. In each iteration, a set of bottleneck links for all flows are found. Bottleneck links in above set are labelled according to their possible maximum reward by acquiring one feasible color. The link with the largest label, say \( j \) is colored with \( h \). Then the topology of the color-sensitive graph and the color lists of some nodes are adjusted in the same way as [7]. Next, we solve a LP problem to allocate every link’s rate to
flows which pass through that particular link. After this, the algorithm starts the next iteration to assign another color. It does not terminate until color lists of all vertexes are empty.

To present our flow-aware approach in detail, we first introduce **Link Rate Allocation Matrix**: \( \Phi = \{ \beta_{i,k} \}_{N \times K} \), an \( N \) by \( K \) matrix. An element \( \beta_{i,k} \) represents the rate that flow \( f_k \) is occupying on link \( l_i \). The rate of a multi-hop flow equals to the rate of its bottleneck link:

\[
\gamma_k = \min_{i \in f_k} \beta_{i,k}, \quad \forall k \in \mathcal{K}.
\]

So, constraint Eq.(2) is turned into:

\[
\sum_{k \in F_i} \beta_{i,k} \leq \sum_{m=1}^{M} a_{i,m} b_{i,m}, \quad \forall i \in \mathcal{N}. \tag{4}
\]

Following is the detail of our flow-aware heuristic scheme:

- **Step 0: Initiate** \( A, \Phi, L \)

At the beginning, channels have not been assigned yet, and the link rate allocation matrix has not been solve.

\[
A^0 = \{0\}_{N \times M}, \quad \Phi^0 = \{0\}_{N \times K}.
\]

We define an **Available Channel List** \( L_i \) for each link, which is an \( M \) by 1 vector. If \( L_i(m) = 1 \), channel \( m \) can be utilized by link \( l_i \). At the beginning, all links’ available channel lists are full:

\[
L_i^0 = \{1\}_{M \times 1}, \quad \forall i \in \mathcal{N}.
\]

- **Step 1: Find the Set of Bottleneck Links**

A set of bottleneck links for all flows, \( G^n = \{(i,k)\} \) are found. An element \((i,k)\) represents that link \( l_i \) is the bottleneck link of flow \( f_k \). Several links on a same flow may be bottleneck links at the same time. In such a tie, one link is randomly selected. As a bottleneck link with an empty available color list could not be improved by acquiring new colors, although it is still possible to get improved by robbing some rate from other flows sharing the same link in step 3), links with empty color list are removed from \( G^n \).

\[
(i,k) = \arg \min_{i \in F_k} \beta_{i,k}, \quad G^n = \{(i,k)\mid \beta_{i,k} = \min_{j \in F_k} \beta_{j,k}, \quad L^{n-1}_i \neq \emptyset \}, \quad \forall k \in \mathcal{K},
\]

in which \( \emptyset \) denotes an empty set.

- **Step 2: Labelling and Coloring Rules (Update \( A, L \)**

All vertexes in \( G^n \) are labelled according to the possible maximum rate that they could get by acquiring any feasible color in the available color list. These label values are solely dependent on the available channel (color) list \( L^{n-1}_i \) and the pre-defined channel (color) reward matrix \( B \).

\[
label_i = \max_{m \in L^{n-1}_i} b_{i,m} \tag{6}
\]

The algorithm selects the vertex with the highest valued label, say \( j \) and assigns the most beneficial color associated with the label, say \( h \) to the vertex.

\[
\text{color}_j = \arg \max_{m \in L^{n-1}_i} b_{i,m} \tag{7}
\]

As one color is successfully assigned, channel assignment matrix \( A^{n} \) is updated to be \( A^n \):

\[
A^n = \{a^n_{i,m}\}_{N \times M} = \begin{cases} 1, & \text{if } i = j, m = \text{color}_i \\ a^{n-1}_{i,m}, & \text{others} \end{cases}
\]

Then color \( h \) is remove from the color lists of vertex \( j \) and its constrained neighbors in available color list.

\[
L^p_n(m) = \begin{cases} 0, & m = h \\ L^{n-1}_p(m), & m \neq h \end{cases} \forall p \in \{j, \text{constrained neighbors of } j\}
\]

The \( h \)-color edges incident on \( j \) are also removed. So the topology of the conflict graph keeps changing as vertexes are processed in each iteration.

- **Step 3: Link Rate Allocation (Solve \( \Phi \)**

After \( A^n \) is acquired, we turn to solve the link rate allocation matrix \( \Phi^n \), which is presented in a LP formulation:

\[
\max \sum_{k=1}^{K} \gamma_k
\]

s.t. \( \gamma_k \leq \beta_{i,k}, \quad \forall i \in f_k, \forall k \in \mathcal{K}, \tag{10} \)

\[
\sum_{k \in F_i} \beta_{i,k} \leq \sum_{m=1}^{M} a_{i,m} b_{i,m}, \quad \forall i \in \mathcal{N}.
\]

Solving this LP problem, we get a new link rate allocation matrix \( \Phi^n \). Redirecting to step 1, a new set of bottleneck links \( G^{n+1}(A^n, \Phi^n) \) for next iteration can be determined. Such iteration does not end until the available color list of all vertexes are empty.

According to the graph coloring procedures, our flow-aware heuristic scheme has no more than \( O(MN) \) iterations, where \( N \) is the number of vertexes, and the number of colors \( M \) is the upper bound of the available color list for each vertex. As we all know, an optimal solution generated by exhaustive search has to finish \( O(M^N) \) iterations.

**IV. Performance Evaluation**

In this section, we study the end-to-end network throughput gain obtained by our algorithm in ring and grid topologies. We
compare our heuristic algorithm with three other algorithms. The baseline is a random channel assignment and the upper bound is the optimal channel assignment generated by exhaustive search. The last reference is a link based heuristic scheme proposed in [7], which aims at maximizing the total link rate. For easy notation, we use RAND, OPT, FL_Unaware, FL_Aware to represent random, global optimal, link based flow-unaware and our flow-aware channel assignments. In the case of RAND and FL_Unaware, the flows passing through a common link are assumed to share this link’s bandwidth evenly. Given the complexity of the optimal solution scales exponentially with the number of links and channels, we do not present OPT in grid topology. In all the simulation, we assume that each link on a given channel has a link rate of 1Mbps, 2Mbps, 5.5Mbps, or 11Mbps with equal probability. All our results are averaged by 100 runs.

A. Ring Topology

In Fig.1(a), N nodes are uniformly distributed on the ring. Each transmission interferes with adjacent 4 links. For example, link \( l_i \) interferes with \( l_{(i+1) \mod N} \), \( l_{(i+2) \mod N} \), \( l_{(i-1) \mod N} \) and \( l_{(i-2) \mod N} \). All results in ring topology are normalized by the largest optimal value. We first examine the relationship between end-to-end network throughput and the number of available channels. In Fig.2(a), two flows of maximum length (4 hops) are randomly generated on a 8 node ring. The available channels varies from 4 to 9. Four algorithms grow as the number of available channels increases. The RAND scales poorly, and the FL_Unaware scales slowly. Our FL_Aware outperforms FL_Unaware at least 65% (in the case of 9 channels) and approaches the OPT. We notice that zero rate multi-hop flows frequently appears in both FL_Unaware and RAND when the number of channels is small. This is because they fail to assign limited number of channels to the key links. In contrast, our FL_Aware performs extraordinarily by hitting right on the most important links.

Then, we examine the impact of the number of flows. In Fig.2(b), there is a 8 node ring. The number of 4 hop flows varies from 1 to 3. Totally 7 channels are assigned to these flows. As the number of flows increase, the OPT keeps increasing, but the increment becomes smaller and smaller. The reason is that the OPT could always feed flows in best condition, while starve those in bad condition on purpose. Our FL_Aware achieves more than 90% of the optimal value on average.

B. Grid Topology

The result in simple ring topology has given an intuitive impression that our flow-aware algorithm excels random and flow-unaware channel assignments, and approximates the global optimal solution. We now evaluate our algorithm in a \( \times \) \( \times \) grid (Fig. 1(b)), which consists of \( 2N(N+1) \) links. We assume that each link conflicts with links within two hops. Several source-destination pairs are randomly selected to generated multi-hop flows. A shortest route is randomly generated for each source-destination pair. In the following, scenarios with different number of channels, flows and various flow length are discussed.

**Vary Number of Channels:** In a \( \times \) \( \times \) grid, we randomly generate \( \frac{N}{2} \) flows with the length of \( N \) hops. The number of available channels varies from 1 to \( \frac{N^2}{2} \). As Fig.3 shows, in the case of 1 or 2 channels, no algorithms could acquire any end-to-end throughput, simply because the minimum conflict set is 3 successive links on any multi-hop flow. As the number of available channels increases, our FL_Aware scale much better than RAND and FL_Unaware by hitting right on the bottleneck links. Both FL_Aware and FL_Unaware are near linear, the slope of FL_Aware is approximately 1.7 times of that of FL_Unaware. The benefit of our FL_Aware channel assignment here is much more than that in a simple ring, barely for the reason that the flow length in grid topology is larger than that in the ring topology. This is further discussed in Fig.5.

**Vary Number of Flows:** There are \( \frac{N^2}{2} \) channels in a \( \times \) \( \times \) grid. We vary the number of \( N \) hop flows from 1 to \( 2N \). As Fig.4 shows, the RAND does not scale with the number of flows because of its random assignment. The FL_Unaware increases at the beginning, but soon ceases
the overall end-to-end throughput. In order to solve such a NP-hard problem, we have proposed a centralized flow-aware heuristic algorithm which assigns the available channels to the bottleneck links of multi-hop flows iteratively. Our simulation in ring and grid topologies has shown the effectiveness of the algorithm in increasing the network throughput. Future work includes distributed algorithm, and introducing fairness considerations and collaborative issues to further explore the potential of multi-channel mesh networks.

REFERENCES


Figure 3: The impact of number of channels in the grid topology

Figure 4: The impact of number of flows in the grid topology

Figure 5: The impact of flow length in the grid topology

scaling up. The reason behind this is simple: at the very beginning, new transmission links appears in the grid. They give the FL_Unaware a better opportunity to enable concurrent transmissions, which increases the spatial reuse. As more and more candidate links appear, FL_Unaware’s decision departs farther and farther from the optimal. Our scheme keeps exploring the spatial reuse gain although the increment decreases as number of flows increases.

*Vary Flow Length:* In a \( N \times N (N = 8) \) grid, there are \( \frac{N^2}{2} \) channels and \( \frac{N}{2} \) flows. The flow length is varied from 1 to 16. As Fig. 5 shows, if flows are only one hop, maximizing total link rate is equivalent to maximizing total flow rate, therefore FL_Unaware and FL_Aware get the same result when flow length is 1. It is intuitively understandable that the network throughput decreases as flow length increases. However, by focusing on the end-to-end flow rate and assigning channels to the weakest part of each flow, our flow-aware algorithm has much better scalability as flow length increases. For example, FL_Aware outperforms FL_Unaware 40\%, 116\%, 206\% when the flow length are 2, 5 and 8 separately.

**V. Conclusion**

End-to-end network throughput is a key metric in wireless mesh networks. In this paper, we have presented an ILP formulation of the channel assignment problem for static multi-radio wireless mesh networks with the goal of maximizing network throughput.