Minimum Interference Channel Assignment in Multi-Radio Wireless Mesh Networks

Anand Prabhu Subramanian, Rupa Krishnan, Samir R. Das, Himanshu Gupta
Computer Science Department
State University of New York at Stony Brook
Stony Brook, NY, 11794-4400, U.S.A.
anandps,krishnan,samir,hgupta@cs.sunysb.edu

ABSTRACT

We address the minimum interference channel assignment problem in multi-radio wireless mesh networks. Given K available channels and R_i radios in node i, we develop an efficient channel assignment algorithm that preserves the connectivity of the network as in the single channel case, and minimizes interference. We formulate this problem as a *constrained graph coloring* problem, develop an efficient TABU search based algorithm, and present initial performance results.

1. INTRODUCTION

There is an increasing interest in using wireless mesh networks as a broadband backbone network. However, it is well-known that the multihop nature of such wireless networks is fraught with capacity problems. These arise mainly due to wireless interference between the links of the network. Two interfering links can operate simultaneously if they operate in different non-interfering channels. When a node in the network is equipped with multiple radios, they can be assigned different channels so that neighboring links will operate on different channels as much as possible. This minimizes the interference in the network. However, the channel assignment problem is non-trivial as the assignment of channels to radios also influences the topology of network. While this problem can be formulated in many ways, our goal in this work is to assign channels to radios so that the original topology of the network (i.e., the topology when only a single channel is used) is preserved, but the wireless interference is minimized.

2. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a wireless mesh network with stationary wireless routers that are equipped with multiple radio interfaces. Each radio in the routers is half duplex, omnidirectional and have identical transmission ranges. We model the network as an undirected graph $G = (V, E)$ (referred to as the *communication graph*) where V is the set of vertices in the graph that represent the wireless router nodes in the network. An edge $e = (v, w) \in E$ if the routers represented by v and w are located within each others transmission range. We assume that there are K channels available, numbered from 1 to K, and that each node i has R_i radio interfaces, where $1 < R_i \leq K$. The assignment problem is trivial when $R_i > K$.

We model the channel assignment problem as a constrained edge coloring problem, where each edge $(v, w) \in E$ is assigned one of the K channels. This constraint preserves the network topology. When an edge $(v, w) \in E$ is assigned a channel κ, the nodes v and w must have at least one of their radios assigned to channel κ. Our goal is to develop a channel assignment algorithm that minimizes wireless interference. We represent interference in the network using a traditional conflict graph model [2]. When two links in the network interfere when they operate in the same channel (i.e., only one can be active at a time), they are said to conflict with each other. The conflict graph $G_c = (V_c, E_c)$ is constructed as follows. For each edge $e = (v, w) \in E$, there is a corresponding vertex $l_{vw} \in V_c$. There is an edge between l_{uv} and l_{xy} in the conflict graph if the edges (u, v) and $(x, y) \in E$ interfere with each other. The conflict graph model is quite general and can accommodate a variety of interference models in the PHY/MAC layers. With this conflict graph representation, minimizing interference is equivalent to minimizing the total number of edges in the conflict graph. This problem in known to be NP-complete [2]. However, efficient heuristic algorithms that provide solutions with a reasonable quality are of interest.

3. TABU SEARCH BASED CHANNEL ASSIGNMENT

The channel assignment algorithm consists of two phases. In the first phase we do not consider the constraint that each network node has a limited number (R_i) of radio interfaces (*interface constraint*). Our aim is to color the edges in G (vertices in G_c) using K colors. Thus, this phase partitions the nodes V_c in the conflict graph G_c into K sets. Any such K-partition, $s = \{S_1, ..., S_K\}$, represents a feasible solution in this phase. Let $I(S_i)$ be the group of edges in the conflict graph G_c with both endpoints in S_i. The following objective function represents the total number of conflicts that should be minimized:

$$f(s) = \sum_{i=1}^{K} |I(S_i)|. \tag{1}$$

We use a TABU search based technique for K-coloring [1] the conflict graph G_c. The initial solution is a random K-partition of the conflict graph. Let s be the initial solution and *bestCol* be its cost as presented in equation (1). Given a solution s, we generate a neighbor solution s_i by moving a random vertex from one partition to another random partition.

Algorithm 1: Phase I - TABU Search algorithm for K-coloring of Conflict Graph

Input : (1) The conflict graph $G_c = (V_c, E_c)$
(2) K - Number of distinct colors
(3) T - The size of the tabu list
(4) η - No. of neigbors in each iteration
(5) *maxIter* - Max iterations before objective funtion improves

Output: (1) The K-coloring $sol = (S_1, ..., S_k)$
(2) Value of the objective function *bestCol*

Start with any random partition $s = (S_1, ..., S_k)$;
$bestCol = f(s)$;
$sol = s$; τ = Null;
$iter = 0$;
while $f(s) > 0$ **and** $iter = maxIter$ **do**
    Generate η neighbors s_i of s with moves $s \rightarrow s_i \notin \tau$ or $f(s_i) < f(s)$;
    Let s' be the best neighbor generated;
    Add the move $s \rightarrow s'$ to τ and remove the oldest move if the list is full;
    if $f(s') < bestCol$ **then**
        $bestCol = f(s')$;
        $sol = s = s'$;
        $iter = 0$;
    else
        $iter{+}{+}$;
        $s{=}s'$;
end while

For every iteration η such neighbors are generated. Let s' denote the neighbor with the lowest cost. If the cost of s' is less than *bestCol*, then this solution is remembered and *bestCol* is updated. The move from solution $s \rightarrow s'$ is recorded in the tabu list τ. This list is useful so that moves are not repeated. The algorithm terminates when *maxIter* iterations have been done without improving *bestCol*. In our simulations, *maxIter* was set to the number of nodes in the conflict graph. The algorithm description is shown in Algorithm 1.

By the end of this phase, all nodes in the conflict graph G_c (edges in G) are colored. The number of colors assigned to a node in the communication graph G is the number of distinct colors assigned to its edges. Let $\gamma(i)$ denote the number of colors assigned to node $i \in G$ by the end of the first phase. This may be more than the number of radio interfaces (R_i) in that node. We define a *violation* metric $\psi(i)$ in each node i as follows:

$$\psi(i) = \gamma(i) - R_i. \quad (2)$$

In the second phase, the nodes in the edge-colored communication graph are sorted in the non-increasing order of the violations. We define edge-connected components in the edge-colored communication graph as connected components where all edges have the same color. For each node

Algorithm 2: Phase II - Satisfying the Interface Constraint

Sort the nodes in G in non-increasing order of *violations* ψ;
for each node $i \in V$ **do**
    while $\psi(i) > 0$ **do**
        Merge two edge-connected components of node i which will give least increase in the number of conflict;
        $\psi(i) = \psi(i)$ - 1;
    end while
end for

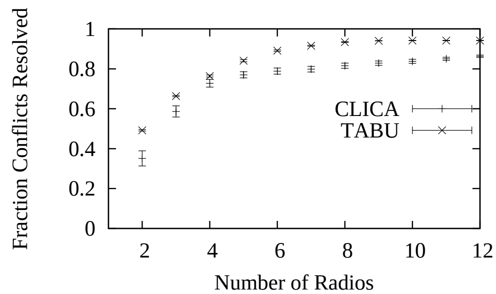

Figure 1: Fraction of Conflicts resolved with 12 channels for a 50 node network.

i in the above order, two of its edge-connected components are merged (i.e., colored with the same color as one of the components) so that it will cause the least increase in the number of conflicts. When two edge-connected components of a node i are combined, $\psi(i)$ decreases by 1 and the violations of other nodes involved in the connected components merged will decrease or remain the same. This algorithm is a one-pass algorithm which greedily merges connected components to satisfy the interface constraints in each node. This phase is described in Algorithm 2.

4. PERFORMANCE EVALUATION

We have evaluated the performance of our channel assignment algorithm using extensive graph-based evaluations. Figure 1 shows the fraction of conflicts resolved (over a single channel case) compared to the CLICA algorithm in [2]. These results correspond to a 50 node network in a 300m $\times$ 300m area, with transmission range equal to 150m. An 802.11 like MAC protocol is assumed, which means that every link in the communication graph interferes with all links within two hops. The 95% confidence interval is shown. We can see that with 5-6 radios and 12 channels, around 90% interference is reduced. Evaluations not reported here show that lesser number of radios are required to achieve similar performance when the network is sparser.

5. REFERENCES

[1] A. Hertz and D. de Werra. Using tabu search techniques for graph coloring. *Computing,39(4)*, pages 345–351, 1987.

[2] M.K.Marina and S.R.Das. A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks. In *Broadnets 2005 Symposium*, 2005.