A Topology Control Approach for Utilizing Multiple Channels in Multi-Radio Wireless Mesh Networks

Mahesh K. Marina
Computer Science Department
University of California, Los Angeles
Los Angeles, CA 90095-1596, U.S.A.
E-mail: mahesh@cs.ucla.edu

Samir R. Das and Anand Prabhu Subramanian
Computer Science Department
State University of New York at Stony Brook
Stony Brook, NY 11794-4400, U.S.A.
E-mail: samir,anandps@cs.sunysb.edu

Abstract

We consider the channel assignment problem in a multi-radio wireless mesh network that involves assigning channels to radio interfaces for achieving efficient channel utilization. We propose the notion of a traffic-independent *base channel assignment* to ease coordination and enable dynamic, efficient and flexible channel assignment. We present a novel formulation of the base channel assignment as a topology control problem, and show that the resulting optimization problem is NP-complete. We then develop a new greedy heuristic channel assignment algorithm (termed CLICA) for finding connected, low interference topologies by utilizing multiple channels. Our extensive simulation studies show that the proposed CLICA algorithm can provide large reduction in interference (even with a small number of radios per node), which in turn leads to significant gains in both link layer and multihop performance in 802.11-based multi-radio mesh networks.

1 Introduction

1.1 Background and Motivation

Wireless mesh networking is emerging as a promising technology for low-cost, ubiquitous broadband Internet access via reduced dependence on the wired infrastructure. In a wireless mesh network, a collection of *wireless access routers* provide connectivity to mobile clients akin to access points in a traditional wireless LAN; but access routers communicate with each other wirelessly, potentially over multiple hops; a small fraction of those access routers are wired to the Internet and serve as Internet gateways for the rest of the network. Mesh networks based on commodity 802.11 hardware and employing self-configuring ad hoc networking techniques can offer wider coverage with less expense and deployment ease. Furthermore, inherent redundancy in the mesh topology enhances reliability. Consequently, mesh networks enable a number of new application scenarios, including community wireless networking to provide affordable Internet access especially beneficial for low-income neighborhoods and scarcely populated areas. Refer to two recent survey articles on mesh networks [1, 2] for a detailed discussion on several application scenarios, and various community and commercial mesh network deployment efforts.

Deploying mesh networks as a simple multihop extension of 802.11 wireless LANs can lead to significant performance problems due to inefficient utilization of available spectrum, in turn degrading end-user throughput. Access points in 802.11 LANs typically have a single half-duplex radio and the MAC protocol in the 802.11 standard [3] is designed to use a single channel. Neither of these prevent wireless LANs from using all available channels (3 non-overlapping channels with IEEE 802.11b/g standards in 2.4GHz band; 12 non-overlapping channels with original IEEE 802.11a standard in 5GHz band) because neighboring access points can be configured to use different channels without the risk of losing connectivity with each other. In contrast, the use of one half-duplex 802.11 radio per node[1] for multihop wireless communication in a mesh forces all nodes to use the same channel to maintain connectivity. As a result, this mode of operation poorly utilizes available spectrum and suffers from the well-known capacity scaling problem with single-channel multihop wireless networks ([4, 5]), arising from the need to share the same channel among neighboring transmissions even those at neighboring hops of a multihop path.

Using multiple 802.11 channels with a single radio per node is possible (e.g., [6, 7]), but this approach has certain drawbacks. A common theme across these single-radio solutions is for each node to dynamically switch between channels, while coordinating with neighboring nodes to ensure communication over a common channel for some period. Such coordination, however, requires tight time synchronization among nodes. Slow switching between channels can reduce synchronization requirements and overheads, but increases end-to-end delays [8]. Besides, these solutions cannot be applied to mesh networks based on commodity 802.11 hardware as they require MAC or hardware modifications.

We therefore consider a mesh networking architecture with multiple radios interfaces per node for effective use of given spectrum [9, 10, 11]. This architecture can not only leverage inexpensive commodity 802.11 hardware, but also overcome deficiencies of single radio solutions. For instance, the use of multiple network interfaces per node allows simultaneous transmission and reception on different interfaces tuned to different channels, which can substantially improve multihop throughput.

A key issue to be addressed in a multi-radio mesh network architecture is the *channel assignment* problem that involves assigning (mapping) channels to radio interfaces to achieve efficient utilization of available channels. A simple approach to address this issue is common channel assignment (CCA) assumed in [9], where radio interfaces at

[1] We will use the terms “node” and “access router” synonymously.

each node are assigned to the same set of channels. Clearly, CCA leads to inefficient channel utilization in the typical case where number of interfaces per node are fewer relative to the number of channels.

Generally speaking, the "goodness" of a channel assignment rests on two factors: connectivity and interference (dependent on load). With multiple radio interfaces operating on different channels, two nodes can communicate only if each of them has an interface assigned to a common channel. Assigning many interfaces to a few channels can provide richer connectivity, but has the undesirable effect of increasing interference among transmissions on those channels. Thus, the channel assignment has to balance between minimizing interference (on any given channel) and maintaining sufficient connectivity. In this sense, the channel assignment in a multihop wireless network can be viewed as a *topology control* problem (similar to transmit power control, for example).

Channel assignment becomes much more challenging in a dynamic network environment because of low overhead inter-nodal coordination required to quickly to adapt the channel assignment in response to time-varying traffic demands and efficiently support a wide range of traffic patterns. It may not be feasible to predict the traffic demand profile *a priori* as in [10]. On the other hand, optimizing the channel assignment for a specific communication pattern to ease coordination [11] may lead to inefficient channel utilization in some cases (e.g., peer-to-peer traffic). Updating the channel assignment is also needed to adjust to spatio-temporal variations in number of available channels (e.g., due to presence of external uncoordinated interference sources in unlicensed bands such as microwave ovens, or opportunistic use of idle licensed spectrum), and to handle device failures/movement. As noted in [11], the main challenge for dynamic and adaptive channel assignment is the multihop coordination requirement that arises because of channel dependency among nodes. Such coordination gets more complex when efficient and flexible support for different communication patterns is desired.

1.2 Our Approach and Contributions

We propose the notion of a *base channel assignment* to ease coordination and enable dynamic, efficient and flexible channel assignment in multi-radio mesh networks. Main idea is to assign channels to radio interfaces to obtain an initial, well-connected topology in a *traffic-independent* manner such that a pair of neighboring nodes have a common *default channel* to communicate with each other, which we refer to as the base channel assignment. Note that different pairs of nodes are allowed to use different default channels for even distribution of load across channels. Also base channel assignment can exploit the usually static nature of access routers in a mesh to update the assignment at relatively large timescales.

The base channel assignment can facilitate different adaptation modes depending on the channel switching delays and relative number of available channels and radio interfaces per node. When channel switching delays are large, as is the case with current 802.11 hardware, base assignment itself can be used for all communications with proper support from higher layers in the form of an adaptive, multipath routing protocol. While this mode of operation is practical and more along the lines of the traditional layered protocol architecture, it may not be the most efficient (as illustrated in an analogous situation in [12]). However, channel switching delays are expected to reduce in the future thereby making highly dynamic channel assignment feasible. Base channel assignment can assist such small-timescale dynamic assignment (e.g., in the order of flow arrival times) in two ways: (i) once the base assignment is available, nodes can use their default channel for subsequent coordination when reassigning channels to interfaces in response to various dynamics (e.g., load variations) as well as to register the new assignment with each other to permit future communication; (ii) alternatively, when nodes have sufficient number of radio interfaces (but still much less than available channels), few interfaces at a node can be devoted for base assignment (long-term) and the rest can be dynamically assigned by coordination via interfaces using the base assignment. This latter approach leads to a hybrid channel assignment somewhat similar in spirit to the one proposed in [8].

In this paper, we focus on the aforementioned base channel assignment problem; we leave the detailed specification of dynamic techniques that leverage the base assignment and evaluation of their relative effectiveness for a separate paper. This paper makes two main contributions. First, we formulate the base channel assignment as a topology control optimization problem that has the goal of minimizing a network-wide measure of interference under the connectivity constraint, and show that it is NP-complete (Section II). Second, we present a greedy heuristic for base channel assignment (termed CLICA) to find connected and low interference topologies (Section III); we then evaluate the performance of this heuristic using an extensive set of simulations (Section IV). In the rest of the paper, unless explicitly mentioned otherwise, we refer to base channel assignment as channel assignment for brevity.

2 Models and Assumptions

Consider a wireless mesh network with access routers arbitrarily distributed on a plane. Each access router is equipped with one or more radio interfaces. We assume that all radio interfaces across the network are half-duplex, use omni-directional antennas and have identical transmission ranges (denoted by R). We model the connectivity between access routers by an undirected graph (henceforth referred to as the *connectivity graph*). A pair of nodes have a link between them in the connectivity graph, if they are located within each other's transmission range.

Suppose that there are K distinct wireless channels denoted by $c_1, c_2, ..., c_K$. Let RI_i denote the number of radio interfaces at node i, where $1 \leq RI_i \leq K$. Note that different nodes may have different number of radio interfaces.

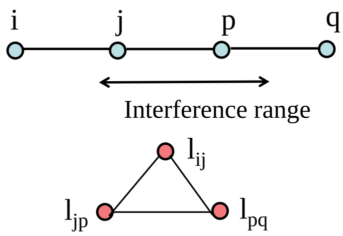


Figure 1: A four node network (above) and the corresponding conflict graph (below). The edge weights in the conflict graph are unity in the base case.

The assignment of channels to radios induces the *network topology*. The network topology and the connectivity graph, in general, may not be identical. This can happen because of two reasons. First, a link in the connectivity graph may be absent in the network topology graph if the nodes at the end points of this link do not have any radios assigned to the same channel. Second, it may have several corresponding links in the network topology if the nodes at the end points have more than one radio each with common channels.

Because of the broadcast nature of the wireless medium, the success of a transmission is greatly influenced by the amount of multiple access interference. We represent such interference using a *weighted conflict graph* [4]. Corresponding to every link $i - j$ in the network topology between nodes i and j, the conflict graph contains a vertex (denoted by l_{ij}). We place an edge[2] between two nodes (say, l_{ij} and l_{pq}) in the conflict graph if the corresponding links ($i-j$ and $p-q$) in the network interfere; the weight of the edge indicates the extent of interference between those links. The existence and extent of interference between a pair of links in the network is determined by an *interference model*. There are two commonly used models for this purpose [5]: (i) Protocol Model; (ii) Physical Model. Because of space limitation, we only discuss the Protocol Model in this paper. We refer the reader to [4] for the Physical Model and a method to find edge weights for conflict graph with that model.

Protocol Model. This model defines interference within the same channel (i.e., co-channel or intra-channel interference). It associates an interference range for each node (typically, larger than the transmission range) and defines a range up to which a transmitter may interfere with the reception in the another receiver. Suppose that all nodes have identical interference ranges (denoted by $R^{'} \geq R$). A transmission from node i to node j is successful provided no other node located within a distance $R^{'}$ from j transmits at the same time. For reliable unicast transmission (e.g., in 802.11 RTS/CTS/DATA/ACK exchange) transmissions in both directions must be successful. Thus, it is additionally required that all nodes located within $R^{'}$ from i refrain from transmitting as well. Thus, the conflict graph for this

[2]As in [4], we associate the terms "node" and "link" with the network topology and the connectivity graph, and use the terms "vertex" and "edge" for conflict graph.

model contains an edge between two vertices (l_{ij} and l_{pq}) if either nodes i or j are located within distance $R^{'}$ from p or q. In the base case, all edges have unit weight. However, it can be further refined as described below. Fig. 1 shows an example illustrating this model. With multiple channels, we need to consider inter-channel interference as well. Although it is straightforward to extend our model to accommodate interference between channels, due to space constraints, we will limit our attention to multiple non-interfering channels in this paper.

3 Multi-Radio Channel Assignment Problem

The channel assignment problem involves obtaining a mapping between radios and channels. We formulate this problem as a topology control problem. Unlike a wired network, links in a wireless network are flexible entities which can be configured or tuned [13]. There are a wide variety of tunable link parameters including transmission power, bit rate, frequency band/channel and beam direction (if directional antennas are used). In a broad sense, topology control is a way to exploit such link controllability to obtain a desired topology. Typically, topology control is targeted towards reducing interference, or improving energy efficiency. Our focus here is on the former. While much of the topology control work is aimed at transmit power control [14, 15], we will look at channel assignment as yet another way to do topology control. Our main goal is to facilitate effective utilization of available channels, which is equivalent to reducing the interference on any given channel by distributing it across as many channels as possible.

With the above view, channel assignment becomes an optimization problem, where "some interference measure" defined over the whole network according to a given interference model is optimized, with the constraint that "some notion of connectivity" is preserved. Now, there are three sets of parameters that need to be instantiated to define the problem more concretely, and they all serve as inputs to the problem: (i) the interference model. We will use the Protocol Model (described earlier) in this paper for ease of exposition. However, our algorithm is general enough to account for other interference models including the Physical Model [5], by virtue of using conflict graphs [4] to represent interference. Interference is modeled by the *link conflict weight*, which is simply the sum of weights of the edges incident to the vertex in the conflict graph corresponding to a network link. For the link $i-j$ assigned to channel c, the link conflict weight is denoted by $W(l^c_{ij})$. (ii) the measure of interference that serves as the objective function. In this work, we choose to minimize the maximum interference (i.e., link conflict weight) at any link. Alternatively, we can also minimize the average interference over all links. (iii) The connectivity constraint serves as the third parameter of the problem. One reasonable choice for this constraint is to require that all links in the connectivity graph are still "preserved" in the network topology after the channel assignment is complete. This ensures that the shortest path length (in number of hops) between any two network nodes does not increase because the network topology becomes different from the original connectivity graph due to channel assignment.

We refer to the decision problem equivalent of the above optimization problem as *Connectivity-preserving Interference-bounded Channel Assignment.* The decision version can be stated as follows.

INSTANCE: Connectivity graph $G = (V, E)$; K distinct channels; RI_i radio interfaces at node i; Interference model; non-negative integer M.

QUESTION: Is there a connectivity-preserving assignment of channels to radios such that the maximum (or, average) link conflict weight in the resultant network topology $\leq M$?

3.1 Analysis of Complexity

Most of the traditional channel assignment problems for wireless networks are known to be difficult and have close relationship with graph coloring problems [16, 17]. Even though at first sight, there does not seem to be any connection between these problems and the above channel assignment problem, a closer look does reveal that our problem is in fact a generalized version of a well-known graph edge coloring problem. Unlike these other problems which either seek proper coloring or conflict-free channel assignment, we attempt to minimize a measure of conflict. Except for the extreme and trivial cases where there is one radio per node or as many radios as number of channels, the optimal solution is intractable. Below, we show that it belongs to the class of NP-complete problems.

Theorem 1. *The connectivity-preserving interference-bounded channel assignment problem as stated above is NP-complete.*

Proof. The problem is clearly in NP since an assignment can be verified in polynomial time.

The rest of the proof is by restriction [18]. We show that the above channel assignment problem contains a known NP-complete problem *minimum edge coloring* (also called *minimum chromatic index*) [19] as a special case. For clarity, *minimum edge coloring* is restated below from [18](OPEN5).

INSTANCE: Graph $G = (V, E)$ and a positive integer K.

QUESTION: Does G have a chromatic index $\leq K$, i.e., can E be partitioned into disjoint sets $E_1, E_2, .., E_k$ (each set denoting a particular color, or in our case, channel), with $k \leq K$, such that, for $1 \leq i \leq k$, no two edges in E_i share a common end point in G?

Now we note that a specific instance of the *connectivity-preserving interference-bounded channel assignment* problem is identical to the *minimum edge coloring* problem. The following conditions hold for this instance: (i) K = maximum node degree in G; (ii) RI_i = degree of node i; (iii) a simple one-hop interference model holds, where two edges in E interfere only if they have the same color (i.e., assigned to the same channel) and share a common end point in G; (iv) $M = 0$. $\square$

The problem remains NP-complete even when the number of channels, $K = 3$. So, the difficulty does not arise from having more channels. Also, the problem is strongly NP-complete.

4 CLICA Algorithm

In this section, we develop a polynomial-time heuristic called *Connected Low Interference Channel Assignment* or CLICA for assigning channels to radios. Based on our discussion in the previous section about the close relationship between channel assignment and graph coloring, from this point on we sometimes use the term "color" in place of "channel" for ease of exposition.

Before we present our algorithm, let us first look at the pitfall of arbitrarily coloring radios and links using a simple example. Fig. 3 shows a 4 node connectivity graph with one radio per node. Suppose we are given 2 colors: c_1 and c_2. If we first color the link $a - b$ with c_1 (by assigning channel c_1 to the radios at a and b) and later color the link $c - d$ with c_2 in a similar fashion, then we end up with a partitioned network where nodes a and b are disconnected from nodes c and d. This simple example shows that a coloring decision constrains the flexibility for future coloring decisions if we want to preserve the network connectivity. For example, nodes c and d are precluded from using color c_2 because of an earlier choice to use color c_1 for link $a - b$. However, adding more radio interfaces to nodes will provide more flexibility in coloring.

The central idea in the CLICA algorithm is to use that degree of flexibility as a guide in determining the *order* of future coloring decisions. Specifically, each node is associated with a priority, and coloring decisions are made on a node-by-node basis in the order of this priority. The set of coloring decisions at a node i include choosing colors for radios at i and its adjacent nodes in order to color all links incident to i in the connectivity graph. At the beginning of the algorithm, each node is given a priority based on some criterion. These priorities determine the default order for making coloring decisions. However, the algorithm, in the midst of its execution, may override that order by setting priority of a subset of nodes to a value greater than the maximum priority (over all nodes) to reflect the lack of flexibility for coloring radios at nodes in that subset. This characteristic of the CLICA algorithm to alter a node's priority during the course of its execution makes it an *adaptive priority algorithm* [20]. Here, we present CLICA in a generic form. Depending on the specific criteria used to determine the initial node priorities, specific heuristics can be realized[3].

[3]We experimented with various possibilities, but for simulation results presented in this paper, we assign initial node priorities as follows: a node

ALGORITHM CLICA

Input: (1) Connectivity graph $G = (V, E)$,
(2) K distinct channels $< c_1, c_2, .., c_K >$,
(3) $\forall i \in V$, $RI_i (1 \leq RI_i \leq K)$ radio interfaces $< r^i_1, r^i_2, .., r^i_{RI_i} >$,
(4) Interference model.
Output: Assignment of channels to radio interfaces at each node that preserves the connectivity specified by G.
This assignment determines the actual network topology.

▷ Phase 1: color all links in G.
Assign a priority to each node in G based on *some* criterion
Order nodes in non-increasing order of their priorities
for each node v in this order
    do CLICA-VISIT-NODE($v, \emptyset$, NIL)

PROCEDURE CLICA-VISIT-NODE($v, NodeSet, c'$)

for each uncolored link $< v, w >$ in G
    do if each of the nodes v and w have a radio with a common color "c"
        then color the link $< v, w >$ with c
            update conflict graph
if $NodeSet \neq \emptyset$ and $c' \neq$ NIL
  then while $\exists$ an uncolored path from v to a node x in $NodeSet$ via a neighbor w
            such that each intermediate node on the path has only one uncolored radio
        do assign c' to the uncolored radio at w
            color the link $< v, w >$ with c'
            update conflict graph
            CLICA-VISIT-NODE($w, NodeSet \cup \{v\}, c'$)
while v has an uncolored incident link $< v, w >$ in G
    do pick a color "c" *greedily* based on objective function evaluated on the conflict graph
        ▷ If all radios at v are already colored, then c is chosen from among currently assigned colors.
        ▷ Otherwise, it is chosen from among unused colors.
        if not all radios at v are colored
            then assign c to an uncolored radio at v
        assign c to an uncolored radio at w
        color the link $< v, w >$ with c
        update conflict graph
        if all radios at w are colored
            then
                ▷ This recursive step indicates implicit increase in priority of node w.
                if all radios at v are colored
                    then CLICA-VISIT-NODE($w, \{v\}, c$)
                    else CLICA-VISIT-NODE($w, \emptyset$, NIL)

▷ Phase 2 (optional): color any remaining uncolored radios from phase 1.

Figure 2: Pseudo-code of the CLICA Algorithm. The algorithm visits the nodes in the connectivity graph and chooses a color (channel) for each radio on the nodes such the original connectivity is preserved. The algorithm maintains the conflict graph on the side, which is used to model interference and guide the greedy heuristic choice of colors.

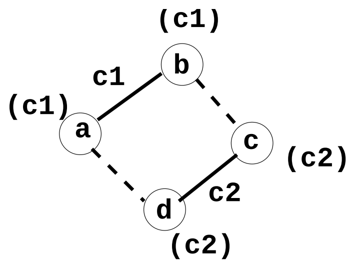


Figure 3: Example illustrating how the connectivity constraint limits coloring choices.

Going back to the example in Fig. 3, suppose that initial order of priorities is a, d, c and b. So, CLICA starts at a to color its incident links. Suppose it chooses c_1 to color the link $a - b$. As a result, both a and b lose further flexibility in choosing colors for their other incident links. So, CLICA additionally bumps b's priority to the highest. Moreover, it recursively starts coloring at b to retain links on other paths connecting a and b (only one path in this example: $b - c - d - a$), which results in node b reusing color c_1 for link $b - c$. Same procedure as above (i.e., priority increase followed by recursive color reuse) repeats itself at node c forcing link $c - d$ to use c_1, which in turn increases the priority of d. At d, since there is already a common color (c_1) with node a, the link $a - d$ is colored with c_1. At this point, CLICA comes out of recursion and terminates. Now suppose that nodes a and d have two radios and the algorithm starts like before at a by coloring link $a - b$ with c_1. Even in this case, the algorithm goes through recursion to color b and c ahead of d; however unlike in previous case the algorithm colors the link $a - d$ with c_2 by using the additional radios. The above two cases are distinguished by lines 21 and 22 in the CLICA pseudocode shown in Fig. 2. Note that CLICA is naturally *recursive* and follows a chain of the least flexible nodes to maintain network connectivity. Also note that it is a *one-pass* algorithm in that coloring decisions once made are not reversed later in the algorithm execution.

Each coloring decision is made in a *greedy* fashion: node i, when faced with a decision to pick a color for an incident link $i - j$, makes a locally optimal choice from among the feasible set of colors: the color that minimizes the maximum link conflict weight over all links that can interfere with the link $i - j$ including itself (GreedyMax). Alternatively, we can pick the color that minimizes the link conflict weight for the link $i - j$ (GreedyAvg)[4].

The algorithm described so far forms the first phase of the algorithm (see Fig. 2). The goal in this phase is to color all links in the connectivity graph to lower overall interference, while satisfying the connectivity constraint (see

is randomly chosen and assigned the highest priority; other nodes are assigned lower priorities determined by the order in which they are discovered during a depth-first search of the connectivity graph starting from the highest priority node — a node i is assigned a higher priority than a node j if i is discovered earlier than j.

[4]Simulation results in this paper use the former method (i.e., GreedyMax).

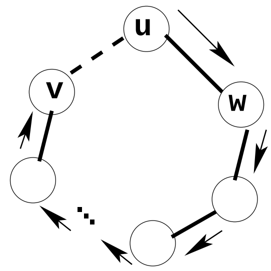


Figure 4: Illustration helpful for proving the connectivity preservation aspect of the CLICA algorithm. See description in text.

Section II). Now we show that first phase of the algorithm preserves connectivity.

Theorem 2. *CLICA (Phase 1) algorithm yields a connectivity preserving color assignment.*

Proof. Recall that a connectivity preserving assignment implies that for every link in the connectivity graph, radios at the end nodes are assigned at least one common channel (color).

Let us consider an arbitrary link $< u, v >$. Without loss of generality, suppose that node u is assigned a higher priority than node v at the start of the algorithm.

Consider the case when u is visited before v (via a call to the procedure CLICA-VISIT-NODE in line 4 of ALGORITHM CLICA). There are two sub-cases within this case. In the first sub-case, u colors the link $< u, v >$. Here we claim that v has at least one uncolored radio (see line 15 in CLICA-VISIT-NODE). Otherwise, if all radios at v are already colored (because of previous calls to CLICA-VISIT-NODE and execution of lines 7, 15) then v would have been visited prior to u via a call to CLICA-VISIT-NODE (see lines 10, 21-22), a contradiction. Thus, both u and v have will end up having a radio with a common color that matches the color of the link $< u, v >$. In the second sub-case, v colors the link $< u, v >$. This could happen in the following scenario. Node u colors its own last uncolored radio and that of a neighboring node w (see lines 20-21). This in turn may trigger further reuse of that color recursively among a set of nodes (including v) starting from w to preserve links on all paths between those nodes (see Fig. 4 and lines 5-10). Because of the recursive color reuse, node v ends up coloring the link $< u, v >$ with a color common to both u and v (see lines 1-4 in CLICA-VISIT-NODE).

Alternatively, v can be visited prior to u. This can happen because of a recursive call to CLICA-VISIT-NODE in lines 10, 21-22. Similar arguments as above apply in this case with roles of u and v interchanged. □

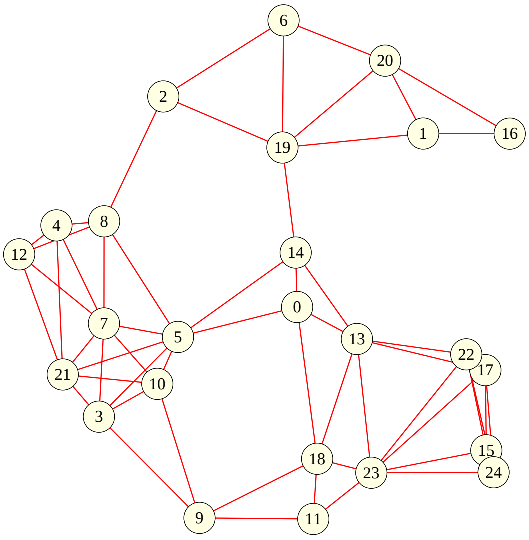
During the first phase, it is possible that radios at two neighboring nodes share more than one common color. We represent this by adding multiple links in the network topology between such nodes. This augmented graph represents the resultant network topology (denoted by T) after phase 1. At the end of first phase, there may still remain some nodes with uncolored radios because each of those nodes have more radios than their respective degree. There are several possible ways to assign channels to uncolored radios and use them. A straightforward method is outlined below. Each uncolored radio at a node is paired with another uncolored radio at a neighboring node, if available; otherwise, it is paired with an already colored radio at a neighboring node. In either case, a new link gets added to T. A color for this link and the corresponding uncolored radio is chosen in a greedy manner as in phase 1. This procedure is repeated as long as there are uncolored radios remaining. Fig. 5 graphically illustrates the topologies generated by CLICA. An alternative and a potentially better way of handling the aforementioned second phase is to assign channels to remaining set of uncolored radios dynamically based on traffic load. This is similar to hybrid channel assignment mentioned at the outset. In fact, we can explicitly control the number of radios available at each node for dynamic assignment by putting a limit on the number of radios that are allowed to participate in the first phase of the CLICA algorithm.

Although our description of the algorithm assumes a centralized setting, it is possible to implement a specific instance of the algorithm in a distributed manner (albeit with limited scalability). Specifically, the first phase of the algorithm in that implementation explores the connectivity graph via distributed depth-first search like procedure (similar to [21]) with token-passing starting from a designated node (e.g., a gateway node wired to Internet). Once first phase is completed, the second phase can be implemented (if needed) in a purely local fashion as there is no risk of breaking connectivity by a color choice. Also note that in such an implementation each node makes coloring decisions based on its own "view" of the conflict graph which could lead to coloring somewhat worse than a centralized solution. We leave further investigation of this issue for future.

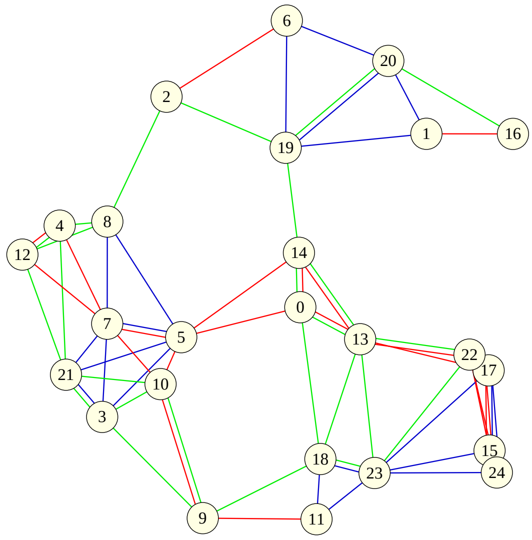
5 Simulation Results

In this section, we study the average-case performance of the CLICA algorithm using simulations. The single channel case serves as a baseline in all our comparisons. We also consider the common channel assignment (CCA) algorithm used in [9], which is the only other traffic-independent scheme in our knowledge. As mentioned earlier, CCA assigns channels to radio interfaces at all nodes identically. More precisely, using the notation from our model, CCA assigns the j^{th} interface at a node i $(1 \leq j \leq RI_i)$ to the j^{th} channel.

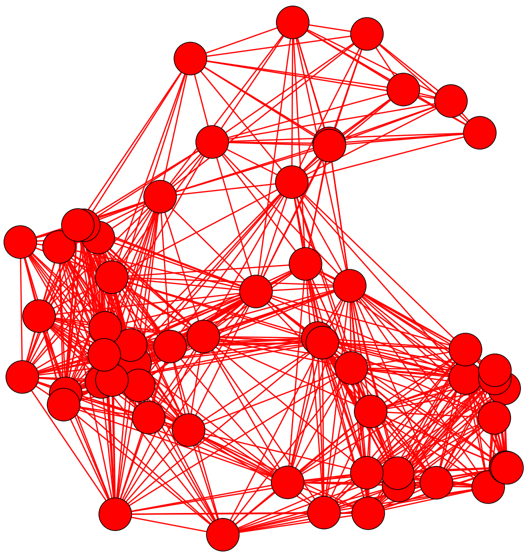
Our evaluation uses a combination of graph-based simulations and ns-2 simulations. Graph-based simulations compare interference and capacity properties of topologies generated by different channel assignment algorithms independent of protocol overheads and interactions — no protocol is modeled in these simulations. As a measure of network-wide interference, we use the maximum link conflict weight metric discussed earlier. Additionally, we use



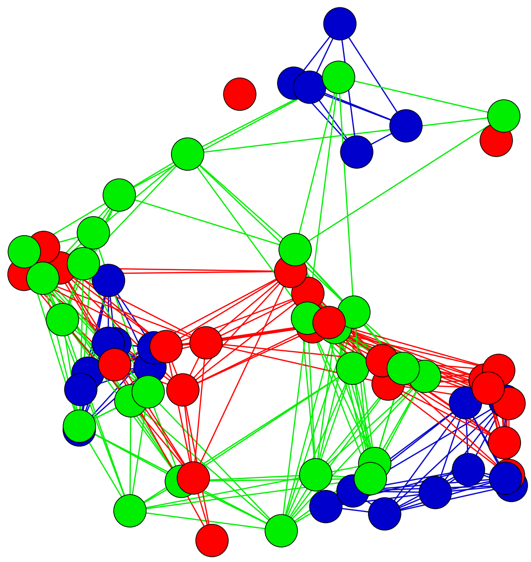
(a) Network topology (single channel)



(b) Network topology (3 channels, 2 radios per node)



(c) Conflict graph (single channel)



(d) Conflict graph (3 channels, 2 radios per node)

Figure 5: Example showing the ability of the CLICA algorithm to generate connected and low interference topologies. This scenario corresponds to 25 randomly distributed nodes with 150m transmission range in a 500mx500m field, protocol interference model with identical transmission and interference ranges, and non-interfering channels. Note that the interference (maximum link conflict weight) for the multi-channel case in this example is reduced by a factor of 3 relative to the single channel with only 2 radios.

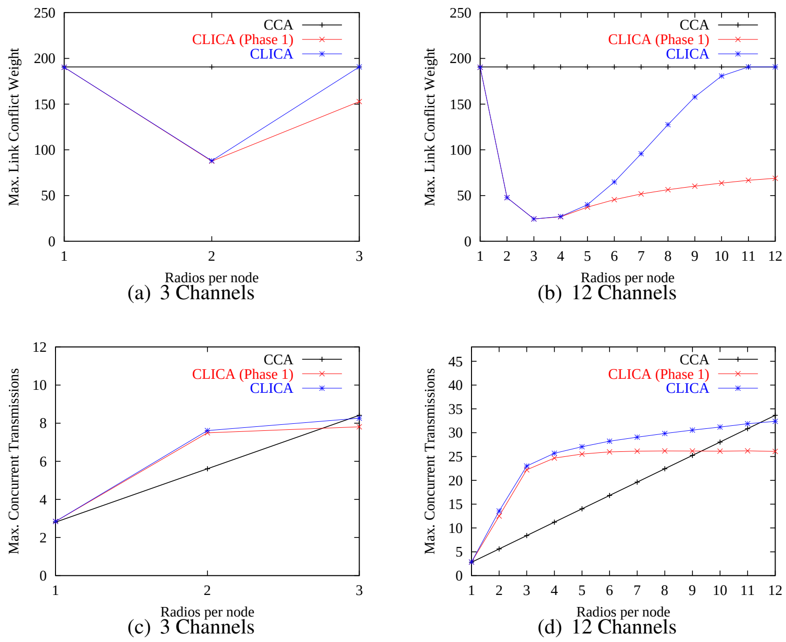


Figure 6: Topologies properties for CLICA and CCA algorithms (interference in left column and capacity in right column) with varying number of radios per node and channels.

the maximum number of concurrent transmissions (calculated by computing maximum independent set in the conflict graph) as a measure of the total one-hop capacity. Since maximum independent set problem is itself NP-complete, we use a greedy $O(1)$ approximation algorithm mentioned in [22]. Later, we conduct detailed ns-2 simulations to evaluate CLICA performance in a 802.11-based multi-radio mesh network using the standard performance metrics of aggregate throughput and average delay. Throughout we consider networks of 50 randomly placed nodes with 250m transmission range in a 1000m$\times$1000m field. Note that these parameters are chosen based on the analysis in [23] to obtain sparse and connected topologies. Such topologies can be seen as the result of applying transmit power control to reduce overall interference in a single channel network. Consequently, they are a good basis to evaluate additional interference reduction possible via intelligent channel assignment. We assume non-interfering channels. For interference within a channel, we assume the Protocol Model [5] with 550m interference range.

5.1 Topology Properties

To study topology properties, we generate a large number of random multihop wireless network topologies (over a thousand for each data point) with varying number of channels and number of radios per node, and apply different channel assignment algorithms on them. All nodes have the same number of radios (varied from 1 to the total number of channels). Fig. 6 shows the results. Note that we present two variations for CLICA – one with Phase 1 alone and the other which includes both phases. Recall that Phase 1 stops when the all links in the original connectivity graph have been assigned channels; however, at the end of Phase 1, some radios may be left unassigned.

CCA interference (maximum link conflict weight) performance (Fig. 6 (a, b)) is unaffected by the number of channels and radios per node, whereas its capacity (maximum number of concurrent transmissions) performance (Fig. 6 (c, d)) shows a linear growth with increase in radios. Both these trends are expected given the way CCA assigns channels to radios. Recall that CCA does identical channel assignment at all nodes. As a result, the number of radios in the network assigned to a channel is same as in the single channel case regardless of the number of radios available per node, which means potential interference remains identical to the single channel scenario. However, each additional radio at a node allows CCA to use an additional channel, hence proportional increase in the capacity. Note that with as many radios per node as the number of channels, CCA capacity is optimal.

CLICA performance is markedly different from CCA. With CLICA, the interference (Fig. 6 (a, b)) reaches a minimum value for a small number of radios per node regardless of number of channels, but the absolute value for the minimum is smaller with more channels. The initial decrease in interference is expected because of the added flexibility in choosing diverse channels with more radios. But the increase beyond a certain point needs some explanation. Observe that network connectivity and interference are inter-related and both are determined by the channel assignment. To obtain a low interference topology, network (and the conflict graph) has to be sparse. However, use of more radios to choose diverse channels acts against this goal after a point through the reuse of channels and consequent addition of new links into the topology. Similar reasoning can be applied for the capacity performance with the CLICA algorithm (Fig. 6 (c, d)) to explain the initial super-linear increase and marginal improvement thereafter. Note that the minimum interference value obtained with CLICA Phase 1 alone matches its two phase variant. This shows that opting to color all available radios can be harmful from the interference perspective. Interestingly, even Phase 1 tends to have increased interference with more radios after a point even though it does not attempt to color all radios. This is due to the underlying greedy nature of the channel selection mechanism that can make poor choices, in turn forcing channel reuse to preserve connectivity. Also note that the point at which interference reaches a minimum does not match that of maximum capacity. For example, with 12 channels and 3 radios per node, CLICA has about 25% lower capacity than the maximum (occurs when number of radios equals 12, the number of channels). We believe this is a consequence of the heuristic nature of the CLICA algorithm. Besides, the two phase CLICA achieves more capacity than with Phase 1 alone, as more operational radios always make more concurrent transmissions possible.

Overall the results in Fig. 6 show that CLICA can generate topologies with low interference with a small number of radios per node, while coming close to achieving optimal capacity. The exact number of radios required, however, depends on the number of channels and node density. This issue requires further study.

5.2 802.11-based Multi-Radio Mesh: Single Hop Performance

We now use ns-2 simulations to evaluate the link layer performance of CLICA in a 802.11-based multi-radio mesh network, in terms of aggregate one-hop throughput and average delay (Fig. 7). We present data for 3 and 12 non-overlapping (and non-interfering) channels to study performance representative of 802.11b and 802.11a networks, respectively. These simulations use a commonly used 802.11 physical layer model in ns-2 that operates at a fixed data rate of 2Mbps. Even though higher data rates are available with 802.11b and 802.11a, the issue of physical layer data rate is orthogonal to our interest here – we are interested in relative performance improvements. The traffic model consists of unicast data with identical poisson packet arrivals between every pair of neighboring nodes in the network. Mean packet arrival rate is varied to obtain different offered loads, while keeping the packet size fixed (1KB). RTS/CTS mechanism is enabled. With CCA and CLICA, when a node can communicate with its neighbor via multiple radio interfaces tuned to different channels, we randomly stripe data across those interfaces (like one of the schemes in [9]). Each point in the plots is an average of five runs with different randomly generated node locations.

As seen from Fig. 7 (a, b), CLICA provides a significant improvement in throughput with multiple radios and channels compared to the single channel case – up to a factor of 3 with 3 channels and 2 radios, and a factor of 9 with 12 channels and 3 radios. For a comparable improvement, CCA needs as many radios as the number of channels. CCA throughput is much lower relative to CLICA with fewer radios per node because the number of channels it can use is limited by the number of radios. The improvement in delay (Fig. 7(c),(d)) over single channel case is often dramatic with CLICA going up to a factor of 100 or more (not easily obvious due to scale of delay plots) for low to moderate traffic loads. This improvement factor is much more than the additional resources (radios and channels) used, which go up to only a factor of 12 in these experiments. The primary reason behind this huge reduction in delay is due to the reduced interference (contention) and collisions, in turn leading to lesser channel access delays (including back-offs and retransmission) in the 802.11 MAC and consequently lesser queueing delays.

5.3 802.11-based Multi-Radio Mesh: Multihop Performance

We again use ns-2 simulations to evaluate the effectiveness of CLICA algorithm for multihop communication, where inter-hop interference is a key limiting factor. Use of multiple channels and radios can reduce such interference through

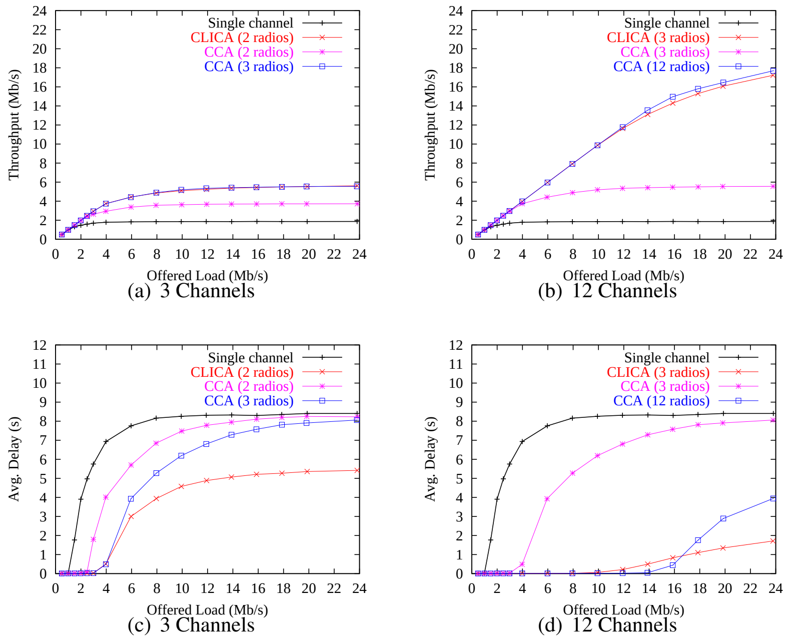
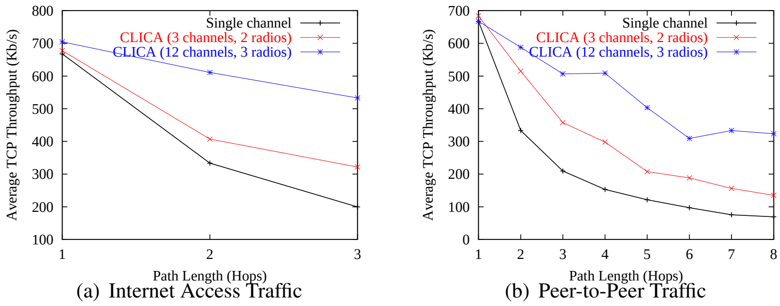


Figure 7: Link layer performance (aggregate one-hop throughput and average delay) with CLICA and CCA algorithms in a 802.11 network relative to the single channel case.

the use of diverse channels at each hop, and allowing simultaneous reception and transmission on different radios at intermediate nodes. In these simulations, we use average end-to-end TCP throughput of a multihop path as the metric. For the application, we use 50 second one-way bulk transfer with FTP. We consider two different traffic patterns. For the Internet access pattern, we assume four randomly located Internet gateway nodes and simulate a data transfer to each non-gateway node from its nearest gateway node; nearness is determined by the shortest path length in hops, a good choice when all links have similar loss characteristics. For the peer-to-peer traffic pattern, we separately simulate a data transfer between 100 randomly chosen node pairs. In both experiments, we disabled striping ability with CLICA in fairness to single channel case and to isolate the benefit of using channel diverse paths. We use 802.11 MAC as before.

Fig. 8 shows the results for the Internet access and peer-to-peer traffic patterns, with data averaged across samples with same path length. As expected, all cases have similar performance for one-hop transfers. With longer paths and more number of channels and radios, CLICA provides larger throughput improvements over single channel case (up to a factor of five).



(a) Internet Access Traffic (b) Peer-to-Peer Traffic

Figure 8: Multihop TCP throughput performance for CLICA algorithm and single channel cases with a single flow and two different traffic patterns (Internet access and peer-to-peer).

6 Conclusions

In this paper, we have considered the channel assignment (radio-channel mapping) problem in multi-radio wireless mesh networks. We have argued that a traffic-independent channel assignment that provides a connected and low interference topology can serve as a basis for dynamic, efficient and flexible utilization of available channels and radios; it can also be readily used to work with current commodity 802.11 hardware with large switching delays to improve utilization of available spectrum. We have formulated this base channel assignment as a topology control optimization problem where we have sought to minimize an overall measure of interference while preserving network connectivity, and showed it to be NP-complete. We have then developed a greedy heuristic channel assignment algorithm called CLICA to find connected and low interference topologies. Extensive set of simulations demonstrate the effectiveness of the CLICA algorithm in exploiting channel diversity for reducing interference with a small number of radios per node, and resulting significant performance benefits in a 802.11-based multi-radio mesh network with single hop as well as multihop workloads.

Our future work will focus on more comprehensive evaluation of the CLICA algorithm, including theoretical performance characterization and real-world performance in a multi-radio mesh testbed. We will also develop scalable and dynamic channel assignment techniques that build on the notion of base channel assignment outlined in this paper.

Acknowledgments

This work was supported in part by NSF grants (award numbers: ANI-0308631 and ANI-0335302). The first author thanks Esther Arkin for helpful discussions and reading an earlier draft.

References

[1] I. Akyildiz, X. Wang, and W. Wang. Wireless Mesh Networks: A Survey. *Elsevier Computer Networks*, 2005. To appear.

[2] R. Bruno, M. Conti, and E. Gregori. Mesh Networks: Commodity Multihop Ad Hoc Networks. *IEEE Communications*, 43 (3), 2005.

[3] IEEE Std. 802.11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.

[4] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of Interference on Multi-hop Wireless Network Performance. In *Proc. IEEE/ACM MobiCom*, 2003.

[5] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, 46(2), 2000.

[6] J. So and N. Vaidya. Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using A Single Transceiver. In *Proc. ACM MobiHoc*, 2004.

[7] P. Bahl, R. Chandra, and J. Dunagan. SSCH: Slotted seeded channel hopping for capacity improvement in ieee 802.11 ad-hoc wireless networks. In *Proc. IEEE/ACM MobiCom*, 2004.

[8] P. Kyasanur and N. Vaidya. Routing and Interface Assignment in Multi-Channel Multi-Interface Wireless Networks. In *Proc. WCNC*, 2005.

[9] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou. A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks. In *Proc. BroadNets*, 2004.

[10] A. Raniwala, K. Gopalan, and T. Chiueh. Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks. *ACM SIGMOBILE MC2R*, 8(2), 2004.

[11] A. Raniwala and T. Chiueh. Architecture and Algorithms for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network. In *Proc. IEEE Infocom*, 2005.

[12] E. Setton, T. Yoo, X. Zhu, A. Goldsmith, and B. Girod. Cross-Layer Design of Ad Hoc Networks for Real-Time Video Streaming. *IEEE Wireless Communications*. In print.

[13] A. Ephremides. Ad hoc networks: not an ad hoc field anymore. *Wireless Communications and Mobile Computing*, 2(5), 2002.

[14] E. Lloyd et al. Algorithmic Aspects of Topology Control Problems for Ad hoc Networks. In *Proc. ACM MobiHoc*, 2002.

[15] M. Burkhart et al. Does Topology Control Reduce Interference? In *Proc. ACM MobiHoc*, 2004.

[16] S. Ramanathan. A Unified Framework and Algorithm for Channel Assignment in Wireless Networks. *Wireless Networks*, 5 (2), 1999.

[17] I. Katzela and M. Naghshineh. Channel Assignment Schemes for Cellular Mobile Telecommunication Systems: A Comprehensive Survey. *IEEE Personal Communications*, 3(3), 1996.

[18] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[19] I. Holyer. The NP-completeness of Edge-coloring. *SIAM Journal of Computing*, 10, 1981.

[20] S. Davis and R. Impagliazzo. Models of Greedy Algorithms for Graph Problems. In *Proc. ACM-SIAM SODA*, 2004.

[21] R. Ramaswami and K. K. Parhi. Distributed Scheduling of Broadcasts in Radio Network. In *Proc. IEEE Infocom*, 1989.

[22] H. Balakrishnan et al. The Distance-2 Matching Problem and its Relationship to the MAC-layer Capacity of Ad hoc Wireless Networks. *IEEE JSAC*, 22(6), 2004.

[23] C. Bettstetter. On the Minimum Node Degree and Connectivity of a Wireless Multihop Network. In *Proc. ACM MobiHoc*, 2002.