Dynamic Channel Assignment and Link Scheduling in Multi-Radio Multi-Channel Wireless Mesh Networks

Hua Yu Prasant Mohapatra Xin Liu

Department of Computer Science, University of California, Davis
Email: {huayu, pmohapatra, xinliu}@ucdavis.edu

***Abstract*—Capacity limitation is one of the fundamental issues in wireless mesh networks. This paper addresses capacity improvement issues in multi-radio multi-channel wireless mesh networks. Our objective is to find a dynamic channel assignment and link schedule that maximizes the network capacity for ftp-type applications and video-type applications respectively. Specifically, we minimize the number of time slots needed to schedule all the flows for ftp-type applications and maximize the minimal link satisfaction ratio for video-type applications. The problems are formulated into linear programming models and we provide two heuristics to solve these problems. One heuristic uses a set covering strategy and the other uses a link-weight-adjusting strategy. We do a trade-off analysis between network performance and hardware cost based on the number of radios and channels in different topologies. This work provides valuable insights for wireless mesh network designers during network planning and deployment.**

I. Introduction

Wireless mesh networks (WMNs) have become a popular option for providing ubiquitous network access to users in the context of home, enterprise, and community networks. Infrastructure-based WMNs consist of statically positioned mesh routers. Such back-haul network architecture is reliable, scalable, cost-effective and easy to deploy [2]. However, the network capacity is limited. If all nodes communicate with a single channel in an IEEE 802.11-based WMN, the number of simultaneous transmissions is limited by signal interference. The system capacity also degrades due to the multi-hop nature of WMNs [7].

One approach to enhance the capacity is to take advantage of multiple channels that are available for use in the IEEE 802.11 a/b/g standards. To better exploit the multi-channel availability, multiple radios are equipped at each node and tuned to different frequencies. Most work in the literature propose heuristic channel assignment algorithms and/or transmission scheduling algorithms based on a fixed number of radios and channels [15, 18, 20]. The capacity limit on a multi-channel multi-radio wireless network has not been extensively studied, especially in scenarios using radios with fast switching capabilities. Bahl et al. [4] stated that the channel switching time could be decreased to approximately 80 microseconds in commercial IEEE 802.11 interfaces. Therefore, it is reasonable to assume that channel switching can be achieved in the time scale of packet transmissions and it is possible for each node to use a different channel in each time slot. In this work, we focus on determining the highest gain we can achieve from increasing the number of channels and radios with such capabilities under certain traffic demands.

We consider two different types of traffic demands based on different application models. One is an ftp-type application where the load can be expressed in terms of data volume. In this application, it is important to minimize the time to transport the load through the network. The other is a video-type application, which has real-time traffic with bandwidth requirements. The load can be expressed in terms of flow rate. In this application, it is more important to satisfy the bandwidth requirement to the extent possible. Since we consider infrastructure-based mesh networks with little topology change, the aggregate traffic load of each mesh router changes infrequently. With routing strategies to produce fixed routes, the aggregate traffic demand on each link can be estimated.

In this work, we first generate a max-flow graph and formulate it as an integer linear programming (ILP) problem by incorporating the constraints derived from the max-flow graph [19]. Then, we propose two algorithms to find a sub-optimal dynamic channel assignment and a centralized link schedule for both ftp-type and video-type application models given a multi-channel multi-radio wireless mesh network.

The paper has the following contributions. First, given a specific topology and the number of channels and radios, we provide a lower bound on the time to schedule all the flows in units of data size, and an upper bound on the minimal link satisfaction ratio, defined as the ratio of the flow rate to bandwidth requirement, among all links.

Second, we propose two application-oriented dynamic channel assignment and link scheduling algorithms. One finds the minimum number of time slots required to schedule all the flows in a given topology. The other maximizes the minimal satisfaction ratio. We find that our algorithms perform well compared to the bounds and the bounds can be reached for some specific traffic patterns. Contrary to most other works [3, 8], the achieved channel assignment and link schedule for each time time slot are feasible because they satisfy both radio and channel constraints. For each one-time-slot schedule, there exists at least one corresponding channel assignment. We select the one with minimum switching overhead.

Third, we evaluate the impact of the topology and the number of radios and channels on system performance. We find that both the number of radios and channels reach a saturating point in decreasing the number of time slots and increasing the link satisfaction ratio. In general, with a small number of channels, 2 radios work very well for most topologies. When more channels are available, adding more radios can help with ftp-type applications, but provides less benefit for video-type applications.This finding provides a guideline to help identify the appropriate number of radios to fully utilize the available channels and the number of channels that is fully utilized by the available radios in a specific topology.

II. Related work

Recently, there has been a significant amount of research in the area of WMNs to enhance system capacity by proposing wireless protocols that utilize multiple channels. Some solutions are based on switching channels [4, 10, 17, 18], while other solutions are based on using multiple radios [1, 6, 10, 15, 16]. In SSCH [4], nodes switch channels synchronously in a pseudo-random sequence such that the neighboring nodes meet periodically at a common channel to communicate. In [17], every node is assigned a quiescent channel and listens to it. The sender switches to the receiver's channel to transmit. [1, 6, 15, 16] require a dedicated interface for each channel. [6] focuses on routing while [15, 16] focus on channel assignment. MUP [1] advocates unifying multiple radios and abstracting their use at higher layers. Recently, there are also some studies on the mechanism of partially overlapped channels [11–14], which permits sender and receiver to use non-orthogonal channels to communicate.

Optimal throughput in multi-channel multi-radios networks is studied in [3, 8, 9]. Apart from [8], the other works assumed the radio interface is not capable of fast switching. Both [3] and [8] assumed the source-destination pairs and considered routing. Li et al. used linear programming (LP) and integer linear programming (ILP) to find the maximum throughput and the corresponding routes of the network [3]. Kyasanur et al. studied the impact of the ratio between number of radios and channels on system performance in the asymptotic case [9]. Kodialam et al. focused on whether a given rate-demand vector can be achieved in the network [8]. Wei et al. proposed a general framework to find the maximum capacity for multi-radio multi-channel networks, which provides a basis for our work. They gave the maximum capacity without considering network traffic, which presents an upper bound on the maximum throughput of any given traffic pattern [19].

Contrary to [3, 8], our work is not focused on routing. Instead, we consider exploiting spectrum reuse to the extent possible given the channel and radio constraints and traffic demands. In addition, because our work does not involve routing, the complexity to find a numerical solution is much less significant. Furthermore, we also consider the impact of number of channels and radios. Compared to [9], which focuses on the asymptotic bound, we studied the relationship between the number of channels and radios.

III. System Model and Problem Formulation

We start with our underlying network model and explain the definitions and concepts used in the rest of the paper. We then formulate the MAC (Multiple Access Control) layer problem.

A. System Model

We consider a wireless mesh network $G = (V, E)$ with M nodes and L possible links, where $V = \{v|v$ is a mesh router $\}$, and $E = \{l|l$ is a link $(u, v), u, v \in V$ $\}$. Here we have $|V|$ = M and $|E|$ = L. If two nodes are in the transmission range, we assume that there is a link between them. Each node v has R radios with fast channel switching capability.

Suppose that there are K orthogonal channels in the system. There are 12 non-overlapping channels in IEEE 802.11a and 3 in IEEE 802.11b/g. Let C be the available channel set, so $C = \{c|c$ is an available channel in the system $\}$ and $|C|$ = K. Let $B(l, c)$ denote the channel capacity across a link $l = (u, v)$, which is the maximum data rate between node u and v on the channel c. We assume that the channel capacity is fixed for each link under each channel, independent on the number of channels and link locations. Then we use B to represent the channel capacity for all the links. Therefore, the aggregate data rate possible by using all K channels and R radios over a link is $min(K, R) \times B$. Our model can easily incorporate the heterogenous channel capacity for each link by replacing B with a link-rate vector $\vec{B}(l)$ where the channel capacity for each link is given.

We model the impact of interference by using the Gupta-Kumar model [7]. A transmission on channel c over link l is successful if all interferers in the neighborhood of both nodes on link l are silent on the channel c for the duration of the transmission. This protocol model of interference captures the behavior of the CSMA/CA protocol used in IEEE 802.11 standards, which follow a RTS-CTS-Data-ACK sequence to protect transmissions. We assume that the data transmissions on different channels do not interfere.

We assume that the system operates in a synchronous time-slotted mode where the length of a time slot is pre-defined as τ seconds. We adopt a time-division multiple access (TDMA) mechanism and schedule the links periodically. Let N_t be the TDMA frame size, i.e. the number of time slots in a period. Channels for the activated links are allocated at the beginning of each time slot. In each time slot, there is no interference among the transmissions of the links scheduled. Thus the performance we obtain will give an upper bound for systems using the IEEE 802.11 MAC protocol.

B. Definitions

As mentioned earlier, we consider two different types of traffic demands, which represent two different applications. First we consider ftp-type applications where the highest priority is to transmit all data in the shortest time. In this case, the traffic demand for each link is given in the form of data size vector $\vec{D^1} = \{d_l^1\}$. Each element denotes the aggregate flow size on all channels across a link $l = (u, v)$. Similarly, we use $\vec{F^1} = \{f_l^1\}$ to represent the scheduled aggregate flow size on all channels across each link. We define a required opportunity vector $\vec{D}_{opp} = \{d_l^{opp}\}$ transformed from $\vec{D}^1$ with each element $d_l^{opp} = d_l^1/(B\,\tau)$.

The other type of application we consider is video-type applications where the highest priority is to satisfy the bandwidth requirement. In this case, the traffic demand for each link is given in the form of data rate vector $\vec{D^2} = \{d_l^2\}$. Each element denotes the aggregate flow rate at which traffic is transmitted between node u and v on all channels across a link $l = (u, v)$. Similarly, we use $\vec{F^2} = \{f_l^2\}$ to represent the scheduled aggregate flow rate on all channels across each link. We define a required link utilization vector $\vec{D}_{util} = \{d_l^{util}\}$ transformed from $\vec{D}^2$ with each element $d_l^{util} = d_l^2/B$.

Our algorithm produces two types of matrices. The first type is channel assignment matrices (CMs), which consist of a corresponding $K \times L$ channel assignment matrix (CMT) CM^t for each time slot. Each element in CM^t indicates whether a channel c is used by a link l or not. $CM^t = \{\delta_{cl}^t\}$ where

$$\delta_{cl}^t = \begin{cases} 1 & \text{if channel } c \text{ is used by link } l \text{ at time slot } t \\ 0 & \text{otherwise} \end{cases}.$$

The second type is a link activation matrix (LM). Each element in this $N_t \times L$ matrix denotes the number of activations for a link l at a time slot t. Each row indicates a **one-time-slot link schedule (OTSLS)** for each link. $LM = \{\theta_l^t\}$ where

$$\theta_l^t = \begin{cases} \alpha & \text{if link } l \text{ is scheduled } \alpha \text{ times at time slot } t \\ 0 & \text{if link } l \text{ is not scheduled at time slot } t \end{cases}.$$

Given this notation, we define an opportunity vector $\vec{S}^T = \{s_l^T\}$, where $s_l^T = \sum_{t=1}^{T} \theta_l^t, \forall l \in E$. Each s_l^T denotes the total scheduled chances for a period length of T time slots. We denote $\frac{s_l^T}{T}$ as the aggregate link utilization ratio on all channels. It corresponds to the fraction of the channel capacity can be achieved. Note that it can be greater than 100% because of the use of multiple radios.

Note here the number of channels used by a link will also be θ_l^t if the link has been activated θ_l^t times, i.e.,

$$\theta_l^t = \sum_{c=1}^{K} \delta_{cl}^t. \tag{1}$$

This is because multiple simultaneous transmissions on a link usually do not share the same channel due to interference. Therefore, the LM can be derived from all the CMTs. A row in LM is just the sum of all the rows in CM^t on corresponding links.

C. MAC layer Problem formulation

For the ftp-type applications, our goal is to transmit all the data through the network as fast as possible. Thus we minimize the number of time slots to schedule all the flows, i.e. $arg\min_{N_t} f_l^1/d_l^1 = 1,\ \forall l \in E$. Note that the scheduled aggregate flow size on all channels across each link f_l^1 is proportional to its total scheduled chances $s_l^{N_t}$ with fixed channel capacity B and time slot length τ, i.e. $f_l^1 = \sum_{t=1}^{N_t} \theta_l^t\, B\,\tau = s_l^{N_t}\, B\,\tau,\ \forall l \in E$. By scaling with $B\tau$, we formally state the problem as follows.

$$Objective: arg\min_{N_t} s_l^{N_t}/d_l^{opp} = 1,\ \forall l \in E \tag{2}$$

subject to

$$\sum_{l \in adj(v)} \theta_l^t \le R, \forall v \in V, \forall t, \tag{3}$$

$$\theta_l^t \le K, \forall l \in E, \forall t. \tag{4}$$

The first constraint is node-radio constraint. At any time slot, a node can use at most R radios to communicate with its neighbors. Here l is the link adjacent with node v. This constraint implies $\theta_l^t \le R$. The second one is a link-channel constraint. At any time slot, a link can be activated on at most K channels. Because of the definition of 0-1 variable δ_{cl}^t, the following equation is always satisfiable: $\sum_{c=1}^{K} \delta_{cl}^t \le K, \forall l \in E, \forall t$. Then by Eqn. 1, the Constraint 4 is always satisfiable in our formulation.

For real-time video-type applications in multi-hop WMNs, maximizing the total flow rates on all the links may not achieve efficient system throughput if some link shared by many end-to-end flows cannot obtain resources. Thus, our goal is to allocate resources to different links proportional to their bandwidth requirement to the extent possible. We denote the link satisfaction ratio as the ratio of the flow rate to the required bandwidth on a link. Then the objective is to maximize the minimal link satisfaction ratio of all links, i.e., $\max\ \min\ f_l^2/d_l^2,\ \forall l \in E$. Note that the scheduled aggregate flow rate on all channels across each link f_l^2 is proportional to its aggregate link utilization ratio $\frac{s_l^{N_t}}{N_t}$ given fixed channel capacity, i.e. $f_l^2 = \frac{\sum_{t=1}^{N_t} \theta_l^t}{N_t} B = \frac{s_l^{N_t}}{N_t}\ B,\ \forall l \in E$. If we scale both f_1^2 and d_1^2 with B, then the link satisfaction ratio can be expressed as the ratio of the aggregate link utilization to the required one. The problem formulation is the same as that for ftp-type applications except the objective becomes

$$Objective: \max\ \min\ \frac{1}{d_l^{util}} \frac{s_l^{N_t}}{N_t},\ \forall l \in E. \tag{5}$$

IV. Dynamic Channel assignment and link scheduling algorithms

Our link scheduling and channel assignment algorithm has three steps. First, we generate the framework to capture the objective and constraints in the max-flow graph. Second, we find the dynamic link schedule according to different traffic demands based on the framework. For ftp-type applications, we use a greedy set-covering strategy to schedule all the flows as fast as possible (function *ScheduleDym1*). For the video-type applications, we use a link weight adjusting strategy to increase the minimal link satisfaction ratio (function *ScheduleDym2*). For both cases we transform the traffic demand accordingly, as mentioned in Section III-C. Lastly, we assign channels to each activated link at each time slot according to the link schedule (function *ChannelAssignmentDym*).

A. Framework Generation

Based on our prior work [19], we include the weights and edge capacity constraints in our modified framework. The objective is to maximize the total weighted capacity on the links subject to both the radio and channel constraints. For example, we may achieve a maximum capacity of $3B$ with links $\{a, b, d\}$ activated for the topology in Fig. 1 under the constraint of 3 channels and 2 radios with all the link weight values equal to 1. This schedule can be represented by the row vector [1 1 0 1 0] in link activation matrix LM with each element indicating the number of activations for a link. This OTSLS also corresponds to a maximum flow size of $3B\tau$ for a given time slot.

We first generate [19] a flow interference graph G^c (Fig. 2) based on the topology graph G (Fig. 1) and the Gupta-Kumar interference model. Each vertex in G^c represents a link in G. Based on the interference graph, we generate the resource contention graph RCG (Level 2 and 3 in Fig. 3). A RCG captures various contention regions in the network topology by identifying all the maximum cliques in the interference graph. There is an edge between a resource vertex and a link vertex if this link belongs to the contention graph represented by the resource vertex.

We extend the RCG to a max-flow graph MG (Fig. 3) by adding a set of image link vertices $(a', \cdots, e')$, a set of node vertices $(A, \cdots, F)$, a source vertex, s, and a sink vertex, t. An edge between the link vertex and its corresponding image link vertex is added. The image link vertices are connected with the node vertices according to the topology graph. Then the node vertices are connected to the sink vertex. The edge capacity for the first three levels is K, which is the number of channels. The edges of the last two levels have a capacity of R, which is the number of radios. For a heterogenous network where different number of radios may be equipped for each node, it is easy to reflect this non-uniformity by setting the edge capacity of the last two levels according to a node-radio vector $\vec{R}(v)$ where the number of radios for each node is given.

Let E' be the set of image links $\{l' | \forall l \in E\}$ and $N(x)$ be the set of neighbors of a vertex x in the max-flow graph. Let f_{ij} be the edge flow value between vertices i and j, where $f_{ij} \geq 0$. To simplify notations, we denote $f_{ll'}$ as f_l. So $F = \{f_l | l \in E\}$ records all the edge flow values for each link in the network at a time slot. Because of the setting of the edge capacities, the edge flow value f_l is at most $min(R, K)$. In addition to the fixed edge capacity on edge ll', we introduce another edge capacity vector $edgeCap$ for each link in E. Let W be the weight vector for each link in E: $W = \{w_l \leq 0 | l \in E\}$. $edgeCap$ and W are known variables, which are dynamically generated by our link scheduling algorithms described later.

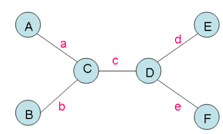

Fig. 1. Topology 1

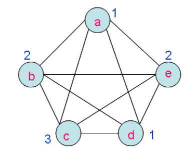

Fig. 2. Flow interference graph of topology 1

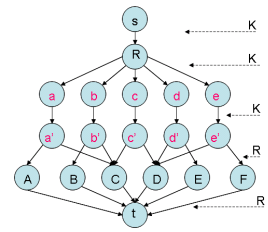

Fig. 3. Framework of topology 1

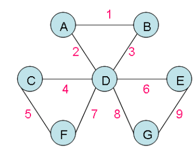

Fig. 4. Topology 2

The ILP problem is formulated as shown in the equations below.

$$Objective: maximize \sum_{l=1}^{L} (w_l * f_l) \tag{6}$$

subject to

$$f_{ie} = f_{ej}, \forall i, j \in N(e), \forall e \in E \cup E', \tag{7}$$

$$\sum_{i \in N(v)} f_{iv} = \sum_{j \in N(v)} f_{vj}, \forall v \notin E \cup E' \cup \{s, t\}, \tag{8}$$

$$f_l \diamondsuit edgeCap(l), \forall l \in E, \diamondsuit \in \{\leq, \geq\}. \tag{9}$$

The first constraint (Eqn. 7) models both the radio and channel constraint. For each link, it is allocated a time slot if and only if it owns resources in all the contention regions it belongs to. That is, any x allocated channels needs to take x units of resource from all of its resource contention regions. It also needs to consume x radios at the end nodes of the link. The second constraint (Eqn. 8) is the flow conservation constraint for all other nodes. The edge capacity constraint (Eqn. 9) is dynamically generated by the function *GetOneSolution* with the quantity relationship specified in the function *ScheduleDym1* and function *ScheduleDym2*.

With the above framework, the achieved solution F is actually OTSLS. In the following sections, the terms “solution” and “OTSLS” are used interchangeably. The solution F under the link weight w_l of value 1 achieves the maximal capacity to satisfy both the radio constraint and channel constraint under certain edge flow capacity constraints. Here, F provides available links that can transmit simultaneously. The value of the variable f_l is the scheduled chance for link l. We describe in the next two sections how to achieve the periodic schedule that maximizes the network capacity under certain traffic demands based on these feasible OTSLSs.

B. Link Scheduling for Ftp-type Application

For ftp-type applications, the objective of the link scheduling algorithm is to find a link schedule that minimizes the number of time slots required to satisfy all the flows. With the transformation mentioned in Section III-C, it suffices that the total scheduled opportunities meet the required opportunities within minimal time slots.

Note that the problem of obtaining all the possible OTSLS, that is, finding minimum time slot schedules to satisfy all the flows, is NP-hard. Thus, we use a greedy set-covering strategy to find a sub-optimal solution to schedule all the flows. The idea of the set-covering strategy is to pick, at each stage, the set that covers the greatest number of remaining elements that are uncovered.

For example, with a required opportunity vector [1 1 3 1 5] corresponding to each link [a b c d e] for topology 1 under 4 radios and 12 channels, we can have a schedule including three OTSLSs [1 1 1 1 2], [0 0 2 0 2] and [0 0 0 0 1]. Consider each opportunity as a covering, so there are 11 opportunities to be covered. Each OTSLS has covered 6, 4 and 2 opportunities, so the schedule satisfies the total required opportunities. Therefore, we make each OTSLS cover as many opportunities as possible until the whole schedule covers the total opportunities. This can be done using the framework presented in Section IV-A. Each time, we set the weight value for each link to 1. In addition to the radio and channel constraint, we impose the edge capacity constraint by setting the scheduled chance no greater than the remaining covering for each link. Then the edge flow value on each link is at most $min(R, K, edgeCap(l))$. Giving a link l fewer chances ($edgeCap(l)$) than what can be allowed ($min(R, K)$) potentially provides more chances to other links who require more coverings if $edgeCap(l)$ is smaller than $min(R, K)$, which saves time in scheduling all the flows.

The algorithm works as follows (function *ScheduleDym1*). Each element of the vector W denotes the weight of each link l, which corresponds to w_l in Equation 6. We initialize the edge capacity for each link $edgeCap(l)$ as the required opportunity $D_{opp}(l)$. The algorithm then works by choosing, at each stage, the OTSLS that has the greatest number of remaining opportunities that are unsatisfied. At each time slot, we generate the ILP problem based on the link weight and edge capacity vector. After achieving a solution F (line 3), we update the opportunity vector S and edge capacity vector $edgeCap$ with the current OTSLS F for all the links, and add F to the set LM (line 4). This process stops when all the flows are satisfied (line 2). If there is a predefined TDMA frame size $MaxT$, we can scale down the traffic demand to meet this requirement. The scaled-up time for the original flow may increase because the value of link flow is limited by the scaled-down demand.

The OTSLS sets LM contains the whole schedule that can satisfy all the flows. Because of the edge capacity, some links get fewer opportunities than what can be allowed. Lines 5 to 7 give the part of the algorithm that better utilizes the spectrum and allows for variation in estimation of traffic demands. It works by setting the scheduled chance to no less than the existing one for each link (line 6). Then the edge flow value on each link is at least $edgeCap(l)$ and at most $min(R, K)$. For example, we get the final schedule consisting of [2 1 1 1 2], [1 1 2 0 2] and [0 3 1 0 3].

The set covering strategy we used is a polynomial-time $(ln(max\{|OTSLS|\} + 1)$-approximation algorithm [5] as each OTSLS is a covering set in standard set covering problem. The maximum size of OTSLS is fixed for a specific topology with a certain number of channels and radios under any traffic pattern, which is achieved by setting all weights to one and skipping edge capacity constraint we imposed here [19]. So considering the traffic demand, the lower bound for the number of time slots to schedule all the flows can be calculated by dividing the sum of required opportunities by the maximum covering size of OTSLS. We plot the bounds in Section V.

Function `ScheduleDym1(`MG,D_{opp}`)`

Input: Max flow graph MG, Required opportunity D_{opp}
Output: link scheduling matrix LM_d

1 `Initialize` ()
2 **while** $\exists S(l) < D_{opp}(l)$ **do**
3 $\quad F \leftarrow$ `GetOTSLS(`*MG,W,'le',edgeCap*`)`
$\quad S \leftarrow S + F$
$\quad edgeCap \leftarrow edgeCap - F$
4 $\quad LM \leftarrow LM \cup F$
5 **foreach** $Result \in LM$ **do**
6 $\quad edgeCap \leftarrow Result$
$\quad F \leftarrow$ `GetOTSLS(`*MG,W,'ge',edgeCap*`)`
7 $\quad LM_d \leftarrow F \cup LM_d$

C. Link Scheduling for Video-type Application

For video-type applications with bandwidth requirements, the bandwidth requirements may not be satisfied because of the constraints on channel capacity and the number of radios and channels. The objective of the link scheduling algorithm is to increase the minimal link satisfaction ratio of the flow rate to the bandwidth requirement on each link. With the transformation mentioned in Section III-C, it suffices to find a link schedule that maximizes the minimal satisfaction ratio of link utilization across all links.

Similarly, note that the problem of obtaining all the possible OTSLS is NP-hard. One intuitive way is to use the method in previous section, which gives a satisfaction ratio no less than $1/|LM_d|$. In this section, we propose another algorithm using weight adjusting strategy and imposing edge flow capacities. We call the first approach the "time-based algorithm" and show the performance difference in Section V.

Our algorithm works by looking, at each stage, for the OTSLS that can increase the current minimal link satisfaction ratio if added. At step T, we calculate the minimal scheduling chance F for each link that maintains the same minimal satisfaction ratio at step $T + 1$ using the equation $minSat * Dutil = (F + S)/(T + 1)$. To find a schedule that can increase the satisfaction ratio, we set the schedule chance for each link at step $T+1$ to no less than $edgeCap(l) = \lfloor minSat * (TSize+1) * D_{util}(l) - S(l) \rfloor + 1$ due to the integrality of OTSLS (line 12). Then the edge flow value on each link is in the range of $(edgeCap(l), min(R, K))$. If no such OTSLS is found, we set the schedule chance for each link at step $T+1$ to no less than $minSat*(TSize+1)*D_{util}(l)-S(l)$ to allow for the same zero satisfaction ratio (line 14). If a positive ratio has been reached and there is no such OTSLS, we stop the search. The link weight W is initialized as the required link utilization D_{util}. At each step we update the weight by decreasing the current schedule chances F. If the maximum weight is less than or equal to zero, we proportionally adjust the weights to keep the relationship of the required link utilization among all the links (line 11). In this way, more scheduling chances will be given to the links who demand more, or many non-bottleneck links that demand less because the ILP is maximizing the total weighted scheduling chances. Here we say a link is a bottleneck link if the node degrees of the end points of the link is high. If a bottleneck link is scheduled, fewer simultaneous transmissions are possible.

Function `ScheduleDym2`(MG,D_{util})

Input: max flow graph MG, Traffic demand D_{util}
Output: link scheduling matrix LM_d

8 `Initialize` ()
9 $F \leftarrow$ `GetOTSLS`(*MG,W,'ge',edgeCap*)
10 **while** $getMore = 1$ **do**
  $LM \leftarrow LM \cup F$, $TSize \leftarrow TSize + 1$
  $S \leftarrow S + F$, $W \leftarrow W - F$
11 **if** $max(W) < 0$ **then**
    $W \leftarrow (abs(min(W)) + 1) * Dutil + W$
  $F_{util} \leftarrow S/TSize$, $sat \leftarrow F_{util}/D_{util}$
  $preMinSat \leftarrow minSat$, $minSat \leftarrow min(sat)$
12 $edgeCap \leftarrow \lfloor minSat * (TSize+1) * D_{util} - S \rfloor + 1$
13 $F \leftarrow$ `GetOTSLS`(*MG,W,'ge',edgeCap*)
  if $\nexists$ *optimal solution* F **then**
    if $minSat = 0$ **then**
14      $edgeCap \leftarrow minSat*(TSize+1)*D_{util} - S$
15      $F \leftarrow$ `GetOTSLS`(*MG,W,'ge',edgeCap*)
    else
16      $getMore = 0$

17 **foreach** $Result \in LM$ **do**
  $edgeCap \leftarrow Result$
  $F \leftarrow$ `GetOTSLS`(*MG,W,'ge',edgeCap*)
18 $LM_d \leftarrow F \cup LM_d$

The algorithm is depicted in the function *ScheduleDym2*. The loop from line 10 to 16 tries to obtain the periodic schedule by considering the time slots one by one. At each time slot, we achieve a current OTSLS F and update the opportunity vector S, link weight vector W and the edge capacity vector $edgeCap$. Then we generate the ILP problem according to the updated weight and edge capacity vector. If there is such a schedule, we loop again and try to see whether we can increase the satisfaction ratio by adding more time slots. Otherwise, we allow for the same zero satisfaction ratio by setting the edge capacity vector as in line 14 or break out of the loop if a positive satisfaction ratio is reached (line 16). We can run the algorithm at most $MaxT$ times if there is a predefined TDMA frame size $MaxT$. Because of the existence of zero weight, the corresponding link may get fewer opportunities than what can be allowed. As in previous algorithm, Lines 17 to 18 give the part of the algorithm that better utilizes the spectrum and allows for variation in estimation of traffic demands.

To evaluate our algorithm performance, we calculate the upper bound as follows. Due to the setting of the edge capacities, the edge flow value on each link is at most $min(R, K)$ for any time slot. Thus, the upper bound for the minimal link satisfaction ratio is $\frac{min(R,K)*N_t}{N_t}\frac{1}{max(D_{util})} = \frac{min(R,K)}{max(D_{util})}$. We plot the bounds in Section V.

D. Channel Assignment

Function `ChannelAssignmentDym`(LM_d)

Input: link scheduling matrix LM_d
Output: channel assignment matrix $CM = \{CT^t\}$;

$mIS =$ `IdentifyMaxIndSets`(G)
$t_d \leftarrow sizeof(LM_d)$
19 **while** $t_d > 0$ **do**
  $S \leftarrow LM_d(t_d)$
  $CT^{t_d}(c,l) \leftarrow 0 \; \forall l \in E, \forall c \in C$
  $C \leftarrow \{1, 2, \cdots, K\}$; $Assigned(l) \leftarrow 0 \; \forall l \in E$
20 **while** $\exists S(l) \neq 0$ **do**
    if ($Assigned(l) > 0$ **and** $Assigned(l) \in C$) **then**
      $c = Assigned(l)$
    else
      pick a channel c from C
    forall j *in mIS(l)* **do**
      $CT^{t_d}(c,j) \leftarrow 1; S(j) \leftarrow S(j) - 1$
      $Assigned(j) = c$
    $C \leftarrow C - \{c\}$
21 $t_d \leftarrow t_d - 1$

The function *ChannelAssignmentDym* depicts the algorithm that assigns channels to each activated link for each time slot according to the link schedule given by the function *ScheduleDym1* or *ScheduleDym2*. The channel assignment (CT^t) is dynamic, and thus, independent for each time slot. At each time (line 21), we first obtain a one-time-slot schedule S and initialize the channel assignment matrix CT^t to zero. Then we assign a different channel (c) to all the links in one of the maximal independent sets until all the activated links in S get a channel (line 20). To minimize the switching overhead, the vector $Assigned$ records which channel was recently assigned to each link. If a link has been assigned to some channel and the channel is available in the channel pool C, then the same channel is assigned to this link; otherwise, a channel c is picked from the channel pool. Note that a link may be assigned to several different channels because of multiple radios. The process stops when the link schedules for all the time slots

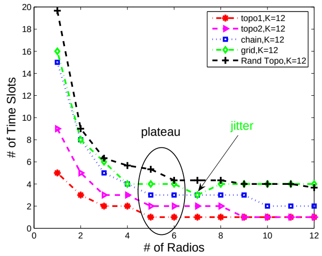

Fig. 5. Time for different topologies with different number of radios

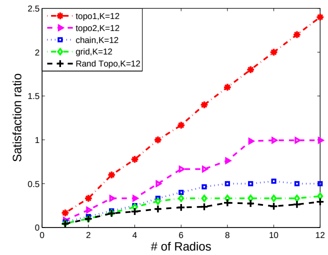

Fig. 6. Link utilization satisfaction ratio for different topologies with different number of radios

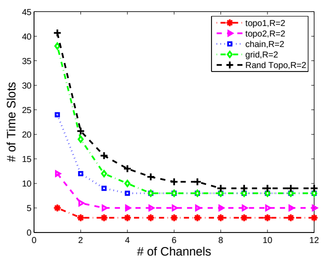

Fig. 7. Time for different topologies with different number of channels

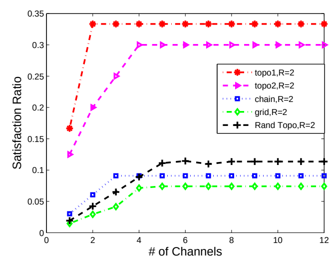

Fig. 8. Link utilization satisfaction ratio for different topologies with different number of channels

are checked (line 19).

V. Performance Evaluation

In this section, we evaluate the impact of the numbers of channels and radios as well as topology on the dynamic channel assignment and link scheduling algorithm for both application models. The reported results are based on the following parameters. For each case, we evaluated five different topologies. These are topology 1 (Fig. 1), topology 2 (Fig. 4), a chain topology, a grid topology and a random topology. The chain topology consists of 20 nodes uniformly distributed on a line. The grid topology is a 4*4 grid. For random topologies, we uniformly and randomly placed 20 nodes in $1000m * 1000m$ square area. We assume two nodes are connected if they are within the transmission range of each other, which is set to 300 meters. This leads to approximately 50 links for a random topology. The results of the random topology shown in the figures are averaged over three different random topologies. For the topologies 1 and 2 (small topology), we randomly generate 5 unit flows each with at most 5 hops; For the last three topologies (large topology), we randomly generate 20 unit flows each within 10 hops. The traffic demands are scaled to B and $B\tau$ respectively for the above two application models and fixed for the same topology in order to compare them.

A. Impact of Number of Radios and Topology

As there are 12 orthogonal channels available in 802.11a, we set the number of channels to 12 in this evaluation. From Fig. 5 and Fig. 6, we see that the number of times slots required to schedule all the flows decreases and the minimal link satisfaction ratio among all links increases with an increase in the number of radios. Second, it can be observed that the number of time slots plateaus at 5 radios. However, the minimal link satisfaction ratio is always increasing with an increase in the number of radios. So adding more radios is more suitable for video-type applications. The little jitter shown for the grid topology in Fig. 5 reflects the approximation of the algorithm. Third, adding a second radio can significantly decrease the required time slots, as shown by the steep slope in Fig. 5. As for increasing satisfaction ratio, adding one more radio almost has the same effect for all topologies.

B. Impact of Number of Channels and Topology

From the previous section, we observe that by using 2 radios instead of 1 has no less improvement than that of adding one more radio both on decreasing the number of time slots or increasing the link satisfaction ratio. Thus, we set the number of radios to 2 in the following simulations. As shown in Fig. 7 and Fig. 8, the number of time slots required to schedule all the flows decreases and the minimal link satisfaction ratio among all links increases with an increase in the number of channels. These trends are similar to the impact of number of radios. Second, it can be observed that the number of time slots plateaus approximately at 4 channels. Different than the impact of the number of radios, the minimal link satisfaction ratio also has a saturating point at approximately 3 channels. This is because we use 2 radios in our simulation. With only 2 radios, most topologies can not utilize more than 3 channels. Third, considering the improvement of adding one more channel on decreasing time and increasing link satisfaction ratio, that of 2 channels over 1 is significant as shown with the large difference of the first two values on each line in both figures. This justifies the use of multiple channels, which greatly increases the possibility of simultaneous transmissions.

C. Relationship between Number of Radios and Channels

In this section, we study the relationship between the number of radios and channels. We vary the number of radios and channels from 1 to 12 to get various combinations of number of radios and channels. Fig. 9 and Fig. 10 show the evaluation results for the grid topology. We observe that the trend is similar to that in the last two sections. With more radios, the saturating point increases with an increase in the number of channels. So with more channels available, more radios can be equipped to exploit the resources. Fig. 9 and Fig. 10 also verified our inference that a small number of radios and channels can achieve favorable results. With 1 radio and 1 channel, the number of required time slots is 38 and the satisfaction ratio is 0.0145. With 2 radios and 3 channels, the number of time slots is decreased to 12, a decrease of 68% and the link satisfaction ratio is increased to 0.0667, an increase of 3.6 times.

In general, with a small number of channels, 2 radios work very well for most topologies, which is also within reasonable costs. When more channels are available, adding more radios

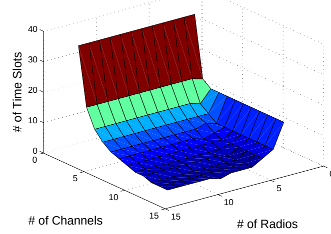

Fig. 9. Time for grid topology with various number of radios and channels

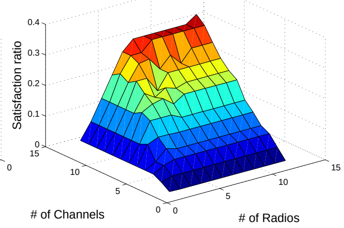

Fig. 10. Link utilization satisfaction ratio for grid topology with various number of radios and channels

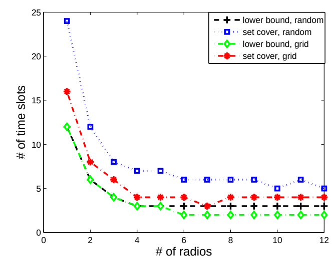

Fig. 11. Lower Bound for random and grid topology with different number of radios

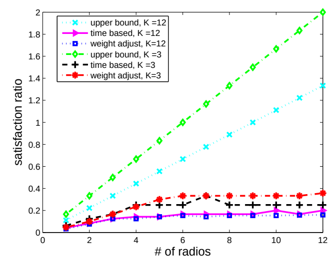

Fig. 12. Upper Bound Link utilization satisfaction ratio for random and grid topology with different number of radios

can help considerably for video-type applications, but to the less extent for ftp-type applications.

D. Performance Comparison with Bounds

We compare the performance of our algorithm with the bounds we derived in Sections IV-B and IV-C. As observed from Fig. 11, our algorithm is between 1.7 and 2.3 times worse than the lower bound for random topology in achieving the minimal number of time slots and between 1.3 and 2.0 times worse than the lower bound for grid topology.

As seen from Fig. 12, our algorithm performs within 12% to 38% of the upper bound for random topology in achieving the maximal minimal link satisfaction ratio and within 10% to 25% of the upper bound for grid topology. Our upper bound is not tight because we assume that 1) at any time slot all the radios and channels can be utilized by the link with the highest traffic demand, and 2) this corresponding link satisfaction ratio is minimal among all the links, which cannot be easily achieved in practice. We also observe that our algorithm performs equally well as the algorithm using the heuristic of minimum time slots for random topology, but performs better for grid topology.

VI. Conclusion

In this work, we propose two application oriented dynamic channel assignment and link scheduling algorithms for a given topology with multiple channels and radios capable of packet level channel switching. For any given traffic pattern, we provide the bounds for both algorithms. We then analyze the impact of the number of radios and channels as well as the topology on system performance. From the results, we observe that increasing the number of radios and channels provides diminishing returns in the amount of time slots minimized and the capacity increased. In general, a small number of channels and radios work very well for most topologies, which is reasonable in cost. When more channels are available, adding more radios can help video-type applications considerably, but to less extent for ftp-type applications. For future work, we will consider the problem of how to non-uniformly distribute the radios to fully utilize the available channels or to satisfy the traffic pattern requirement.

References

[1] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou. A multi-radio unification protocol for IEEE 802.11 wireless networks. In *BROADNETS*, 2004.

[2] I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Computer Networks*, 47(4):445–487, March 2005.

[3] M. Alicherry, R. Bhatia, and L. E. Li. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In *MobiCom*, 2005.

[4] 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.

[5] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2001.

[6] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *MobiCom*, 2004.

[7] P. Gupta and P. Kumar. Capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2), 2000.

[8] M. Kodialam and T. Nandagopal. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks. In *MobiCom*, 2005.

[9] P. Kyasanur and N. H. Vaidya. Capacity of multi-channel wireless networks: impact of number of channels and interfaces. In *MobiCom*, 2005.

[10] P. Kyasanur and N. H. Vaidya. Routing and interface assignment in multi-channel multi-interface wireless networks. In *IEEE WCNC*, 2005.

[11] H. Liu, H. Yu, X. Liu, C.-N. Chuah, and P. Mohapatra. Scheduling multiple partially overlapped channels in wireless mesh networks. In *IEEE ICC*, 2007.

[12] A. Mishra, E. Rozner, S. Banerjee, and W. Arbaugh. Exploiting partially overlapping channels in wireless networks: Turning a peril into an advantage. In *ACM/USENIX Internet Measurement Conference*, 2005.

[13] A. Mishra, E. Rozner, S. Banerjee, and W. Arbaugh. Using partially overlapped channels in wireless meshes. In *Wimesh*, 2005.

[14] A. Mishra, V. Shrivastava, S. Banerjee, and W. Arbaugh. Partially overlapped channels not considered harmful. *SIGMETRICS Perform. Eval. Rev.*, 34(1):63–74, 2006.

[15] A. Raniwala and T. cker Chiueh. Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In *IEEE Infocom*, 2005.

[16] A. Raniwala, K. Gopalan, and T. cker Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 8(2):50–65, 2004.

[17] N. Shacham and P. J. King. Architectures and performance of multi-channel multi-hop packet radio networks. *IEEE Journal on Selected Area in Communication*, 5(6), 1987.

[18] J. So and N. H. Vaidya. Multi-channel mac for ad hoc networks: handling multi-channel hidden terminals using a single transceiver. In *MobiHoc*, 2004.

[19] W. Wang and X. Liu. A framework for maximum capacity in multi-channel multi-radio wireless networks. In *IEEE Consumer Communications and Networking Conference*, 2006.

[20] 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.