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

Hua Yu Prasant Mohapatra Xin Liu
Department of Computer Science, UC Davis, CA 95616
Email: {huayu,pmohapatra,xinliu}@ucdavis.edu
Phone: 530-220-5704 530-754-8380 530-752-4767
Fax: 530-752-4767

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 both dynamic and static channel assignments and corresponding link schedules that maximize the network capacity. We focus on determining the highest gain we can achieve from increasing the number of radios and channels under certain traffic demands. We consider two different types of traffic demands. One is expressed in the form of data size vector, and the other is in the form of data rate vector. For the first type of traffic demand, our objective is to minimize the number of time slots to transport all the data. For the second type of traffic demand, our objective is to satisfy the bandwidth requirement as much as possible. We perform 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 high speed 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 interference. The system capacity also degrades due to the multi-hop nature of WMNs [8].

One approach to enhance the capacity is to utilize multiple channels that are available 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 [21, 25, 29]. The capacity limit

A preliminary version of this paper will be presented at Mobiquitous 2007 [30].
The work was in part supported by Intel Inc. through a gift grant and NSF through CAREER Award #0448613.

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 transmission and it is possible for each node to use a different channel in each time slot. To cater to current commercial off-the-shelf (COTS) radios without fast-switching capabilities, static channel assignment and link scheduling are considered based on the link schedule corresponding to the dynamic channel assignment. Here, channel switching can only be done sparingly, at a slow time scale, for example, when the network topology is changed or the traffic demand is varied. In this work, we focus on determining the highest gain we can achieve from increasing the number of channels and radios with certain traffic demands under both dynamic and static channel assignments.

We consider two different types of traffic demands. One is expressed in the form of data size vector, and the other is in the form of data rate vector. Each element corresponds to each link. For the first type of traffic demand, we may imagine typical traffic in the network is from ftp application, where the data size of each connection is easier to measure. Given this type of traffic demand, our objective is to minimize the time to transport all the data. For the second type of traffic demand, we may imagine the traffic in the network is continuous, like video-streaming, where the data rate is easier to measure. With this kind of traffic demand, our objective is to satisfy the bandwidth requirement as much as possible. Then in the later part of our report, we associate the first type of traffic demand with ”ftp-type applications”, and the second type with ”video-type applications” for description. However note that it does not mean that our scheme is only for the ftp or video applications.

Since we consider infrastructure-based mesh networks with little topology change, the aggregate traffic load of each mesh router changes infrequently. With routing strategies that produce fixed routes, the aggregate traffic demand on each link can be estimated. Also some existing software tools, such as COMO [9], provide the ability to measure the average traffic.

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 [28]. Then, we propose algorithms to find a sub-optimal channel assignment and a centralized link schedule for both ftp-type and video-type application models in a multi-channel multi-radio wireless mesh network, wherein all the radios either have fast-switching capabilities or not. We named these algorithms as FDCA (*ftp-type dynamic channel assignment*), VDCA(*video-type dynamic channel assignment*), FSCA(*ftp-type static channel assignment*), and VSCA(*video-type static channel assignment*).

The paper has the following contributions. First, we consider both switching radio case and non-switching radio case. For each case, we propose two different channel assignment and link scheduling algorithms based on different traffic demands. 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, defined as the ratio of the flow rate to bandwidth requirement, among all links. For switching radio case, contrary to most other works [3, 10], the achieved channel assignment and link schedule for each 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. For non-switching radio case, we are achieving high performance by resolving the radio constraint and channel constraint from the link schedule obtained from the dynamic channel assignment.

Second, 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. As the performance of dynamic channel assignment bounds that of static channel assignment, the derived bounds apply to both channel assignments. We find that our algorithms perform well compared to the bounds and the bounds can be reached for some specific traffic patterns.

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, two radios work very well for most topologies. When more channels are available, adding more radios can help ftp-type applications, but may provide 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

Generally,there have been two types of approaches. One approach assumes the radio is capable of fast switching on per-packet level [4, 7, 14, 23, 25, 29] while another approach assigns channels to multiple radios for a relatively long time [6, 15, 19, 20, 24, 26]. Some other works [3, 7, 10, 12, 21, 22, 27] produce the routing strategy while considering the CA in order to maximize the network capacity. Recently, there are also some studies on the mechanism of partially overlapped channels [13, 16–18], which permits sender and receiver to use non-orthogonal channels to communicate.

Dynamic channel assignment. 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 [23], every node is assigned a quiescent channel and listens to it. Sender switches to receiver's channel to transmit. In MMAC [25], nodes meet at a common channel periodically to negotiate the channels to use for transmission in the next phase. In DCA [29], One radio is used for the RTS/CTS control packets and the other is for data packets. The data packets are sent in the channels negotiated through control packets. Das et. al [14] proposes MAC protocol for multiple radios by using an additional busy tone interface and a MAC protocol that only requires a single radio interface. Although there are many protocols designed for dynamic channel assignment in multi-channel networks, it is not apparent in these proposals how many radios are actually required to maximally utilize the available channels. In our work, we are trying to investigate how many radios and channels are adequate and the performance bounds.

Static channel assignment Das et al. [6] present a couple of optimization models for multi-radio multi-channel WMNs, but without any practical algorithm. The authors [19] propose a linear optimization model channel allocation and interface assignment model. Ramachandran et al. [20] proposed a centralized channel assignment algorithm where one radio at each node is tuned to a common channel to preserve the original topology. Kim et al. [24] presented a distributed randomized channel assignment scheme SAFE to distribute edges sharing a particular channel evenly while maintaining network connectivity. Compared to their approach [19, 20, 24] to assign channels to the radio interfaces, we think that our approach of assigning channels to links is better since assignment of channels to radios still leaves the problem of which channel to use for a transmission/link. Das et al. [26] addressed the problem of assigning channels to communication links in the network with the objective of minimizing overall network interference. They designed a centralized tabu-based and a distributed greedy algorithm to solve this problem with a graph-coloring approach. Marina and Das [15] posed the CA problem as a coloring problem with the objective of minimizing the maximum size of interfering edge set over all edges. Compared to [15, 26], our work specifically considered the traffic demand when maximizing the network capacity.

Joint CA and routing Raniwala et al. proposed such work in [22] and a distributed version in [21]. However, they do not quantify the performance of their solutions with respect to the optimal, and their assignment algorithm works only for mesh nodes whose connectivity graph is a tree. Gong et. al [7] proposed a combined proactive routing and multi-channel medium access control (CA-OLSR and MC-MAC) protocol to improve the capacity of wireless ad hoc networks. 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]. Kodialam et. al. focused on the capacity planning problem of feasibility of a given end-to-end demand vector in multi-radio multi-channel wireless mesh networks [10]. Lin and Rasool [12] developed a fully online distributed algorithm that jointly solves the channel-assignment, scheduling and routing problem. The work in [3, 21, 22] only considers static channel assignment while [7, 12] considers the dynamic channel assignment.The work in [10] considers both. Different from this group of work, our work decouples the routing problem with the channel assignment problem. In addition, because our work does not involve routing, the complexity to find a numerical solution is much less significant. For switching radio case, contrary to [10], the achieved channel assignment and link schedule for each time slot are feasible because they satisfy both radio and channel constraints.

III. System Model and Problem Formulation

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

A. System Model

We consider a relatively small wireless mesh network $G = (V, E)$ with M nodes and L links, where $V = \{v|v$ is a mesh router $\}$, and $E = \{l|l$ is 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.

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 of 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 protocol model in [8]. 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. Due to board crosstalk or radio leakage [1, 11], commodity radios on a node may actually interfere even if they are tuned to different channels. However, this phenomena can be addressed by providing some amount of shielding or antenna separation [11], or increased channel separation (as is the case in 802.11a) [21].

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. For dynamic channel assignment, channels for the activated links are allocated at the beginning of each time slot. For static assignment, the channel assigned for each link is fixed for all the time slots during a long period. In each time slot, only non-interfering links are 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. First, 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 $l = (u, v)$. To simplify explanation, we associate the solution for this type of traffic demand with the solution for ftp-type applications later on. 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 $d_l^{opp} = d_l^1/(B\,\tau)$.

Second, 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 $l = (u, v)$. To simplify explanation, we associate the solution for this type of traffic demand with the solution for video-type applications later in this report. 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 matrices. The first 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 above is for dynamic channel assignment. With static channel assignment, all the CM^ts are the same, which result in only one static channel assignment matrix $CM_s = \{\delta_{cl}\}$ where

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

The second 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

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

As there is one static channel assignment matrix to indicate all the possible channels assigned for a link over the time, a link assigned with one of the channels may not be scheduled at a particular time slot due to channel constraint. So in our implementation, for each element θ_l^t, we specifically indicate which portion of the channels are assigned at a particular time slot. However, for consistence of notation, we still denote the element as the number of activations instead of on what channels the link is activated.

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 for dynamic channel assignment 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.

However, equation 1 does not hold for static case. Actually $\theta_l^t \leq \sum_{c=1}^{K} \delta_{cl}$. Because CM_s indicates all the possible different channels assigned for each link, a link assigned with some channel may not be scheduled at one of the time slots due to channel constraint. Note that the maximum number of all the possible channels assigned for a link does not exceed the number of radios over the time, whereas it is possible for dynamic channel assignment due to switching capabilities of radios.

C. MAC layer problem formulation

We first describe the problem formulations for ftp-type and video-type applications with dynamic channel assignment.

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. $arg \min_{N_t} f_l^1/d_l^1 \geq 1,\ \forall l \in E$. 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.

$$Objective : \min_{CM^t} N_t \tag{2}$$

subject to

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

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

$$\sum_{l \in CG(G)} \delta_{cl}^t <= 1, \forall t, \tag{5}$$

$$s_l^{N_t}/d_l^{opp} \geq 1, \, \forall l \in E. \tag{6}$$

The first constraint is node-radio constraint. Any successful transmission on a link prevents from the transmissions from all the other links in the same contention region. 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 $\theta_l^t \leq R$. 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: $\sum_{c=1}^{K} \delta_{cl}^t \leq K, \forall l \in E, \forall t$. Then by Eqn. 1, the constraint in equation 4 is always satisfiable in our formulation. The third one is interference-channel constraint. At any time slot, at most one link in the same contention region can be activated with the same channel, where the links in the same contention region interfere with each other. The last is the flow constraint. The scheduled flow size on each link should be no less than the required one.

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

$$Objective: \max \, \min \, \frac{1}{d_l^{util}} \frac{s_l^{N_t}}{N_t}, \, \forall l \in E \tag{7}$$

and that the last constraint is not needed.

We move to describe the problem formulations with non-switching radio case next.

For the ftp-type applications, the objective is similar but we have one more decision variable LM. This is because LM is not fully determined by CM_s. As the radio is not fast-switching capable, we need to add one more constraint, i.e. non-switching constraint. It means that the number of different channels assigned on each radio over the time is smaller than or equal to the number of radios. Thus the modifications of the formulation from switching-radio case is as follows.

$$Objective: \min_{CM_s, LM} N_t \tag{8}$$

subject to

$$\sum_{c=1}^{K} \delta_{cl} \leq R, \forall l \in E, \forall t \tag{9}$$

$$\sum_{l \in CG(G)} (\bigcap \theta_l^t) = \phi, \forall t \tag{10}$$

For real-time video-type applications, the formulation is similar to the switching radio case with the exception of adding the constraint in equation 9.

IV. Dynamic Channel assignment and link scheduling algorithms

Our link scheduling and channel assignment algorithm (FDCA and VDCA) 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*). The channel assignment in the last step can be integrated into the second step whenever a OTSS has been achieved. However, to make explanation clear, we describe the last step separately. The dynamic channel assignment algorithm for ftp-type applications is listed in algorithm *1* while VDCA is similar to FDCA by replacing function *ScheduleDym1* with function *ScheduleDym2* and using parameter $D_{util} = D./B$ as input.

Algorithm 1: FDCA

Input: network topology graph G, traffic demand D
Output: link scheduling matrix LM_d; channel assignment matrix $CM = \{CT^t\}$;
begin
  $MG \leftarrow$ `GenerateFramework(`G`);`
  $D_{opp}(l) \leftarrow \lceil D(l)/(B\tau) \rceil$;
  $LM_d \leftarrow$ ScheduleDym1(MG, D_{opp}) ;
  $CM =$ ChannelAssignmentDym(LM_d);
end

A. Framework Generation

Based on our prior work [28], 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.

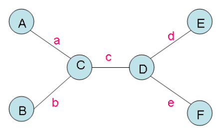


Fig. 1. Topology 1

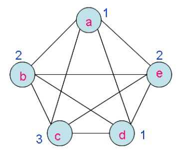


Fig. 2. Flow interference graph of topology 1

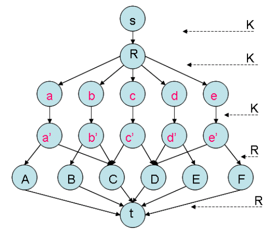


Fig. 3. Framework of topology 1

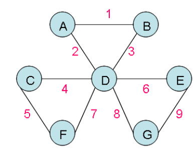


Fig. 4. Topology 2

For example, we may achieve a maximum capacity of $3B$ with links $\{a, b, d\}$ activated for the topology in Fig. 1 under 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 [28] 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.

The ILP problem is formulated as shown below.

$$Objective : maximize \sum_{l=1}^{L} (w_l * f_l) \tag{11}$$

subject to

$$f_{ie} = f_{ej}, \forall i, j \in N(e), \forall e \in E \cup E', \tag{12}$$

$$\sum_{i \in N(v)} f_{iv} = \sum_{j \in N(v)} f_{vj}, \forall v \notin E \cup E' \cup \{s, t\}, \tag{13}$$

$$f_l \diamondsuit edgeCap(l), \forall l \in E, \diamondsuit \in \{\leq, \geq\}. \tag{14}$$

The first constraint (Eqn. 12) 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. 13) is the flow conservation constraint for all other nodes. The edge capacity constraint (Eqn. 14) 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, which is simple to implement and has close approximation for small networks. 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 1 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 Eqn. 11. 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 [28] by setting all weights to one and skipping edge capacity constraint we imposed here. 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 VI.

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

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

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

`Initialize` ()
while $\exists S(l) < D_{opp}(l)$ **do**
  $F \leftarrow$ `GetOTSLS(`*MG,W,'le',edgeCap*`)`
  $S \leftarrow S + F$
  $edgeCap \leftarrow edgeCap - F$
  $LM \leftarrow LM \cup F$

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

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 $1/|LM_d|$. However, a link schedule that maximizes the minimal link satisfaction ratio may not have the minimal number of time slots. 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 VI.

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

Function ScheduleDym2(MG,D_{util})

```
Input: max flow graph MG, Traffic demand D_util
Output: link scheduling matrix LM_d
8  Initialize ()
9  F ← GetOTSLS(MG,W,'ge',edgeCap)
10 while getMore = 1 do
       LM ← LM ∪ F, TSize ← TSize + 1
       S ← S + F , W ← W − F
11     if min(W) < 0 then
           W ← (abs(min(W)) + 1) ∗ Dutil + W
       F_util ← S/TSize, sat ← F_util/D_util
       preMinSat ← minSat, minSat ← min(sat)
12     edgeCap ← ⌊minSat ∗ (TSize + 1) ∗ D_util − S⌋ + 1
13     F ← GetOTSLS(MG,W,'ge',edgeCap)
       if ∄ optimal solution F then
           if minSat = 0 then
14             edgeCap ← minSat ∗ (TSize + 1) ∗ D_util − S
15             F ← GetOTSLS(MG,W,'ge',edgeCap)
           else
16             getMore = 0
17 foreach Result ∈ LM do
       edgeCap ← Result
       F ← GetOTSLS(MG,W,'ge',edgeCap)
18     LM_d ← F ∪ LM_d
```

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.

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 VI.

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 are checked (line 19).

V. Static Channel assignment and link scheduling algorithms

Our static channel assignment and link scheduling algorithms (FSCA and VSCA) are based on the results obtained from algorithm FDCA and VDCA, respectively. The dynamic link channel assignment gives the highest flexibility to maximize the achievable performance for any link channel assignment scheme. Our objective of static channel assignment is to minimize the gap between them by resolving the constraint imposed by radios without fast-switching capabilities.

A. Link Scheduling and static channel assignment 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 complete all the flows. At the same time, the radio and channel constraints must be satisfied.

As mentioned before, the problem of obtaining all the possible OTSLSes, that is, finding minimum time slot schedules to satisfy all the flows, is NP-hard. We still use the greedy set-covering strategy to find a sub-optimal solution to schedule all the flows. For each link, the number of covering is the minimum of the required opportunity and total scheduled opportunity.

Therefore, we make each OTSLS cover as many opportunities as possible until the whole schedule covers the total opportunities. Before doing this, we need to consider the non-switching radio constraint first. The OTSLS in xDCA is independently solved because of the flexibility of dynamic channel assignment. However, we need to check the dependence by scanning all the OTSLSes while meeting all the constraints.

The algorithm works as follows (algorithm *5*). We use U to denote the set of available channels in the channel pool at each time slot. It is initialized to the available channel set C in the system. Whenever a channel is assigned to some link, it is removed from U to avoid link interference. We use NC to denote the channel set assigned to each radio.

The algorithm has three stages. First, we make sure that the links with non-zero flow requirement are activated at least once in the static link schedule by periodically scanning each OTSLS in the dynamic schedule. This is depicted in Lines 2 to 9. If there exists an activated link l in the dynamic schedule that has non-zero flow requirement and has not been activated in the static schedule, we check whether this link can be scheduled while meeting the non-switching radio, node-radio, and interference-channel constraints (Line 4). If there are no such available channels, we will check which constraint is violated and decide to resolve it or not (Line 5). If there exists available channels to be assigned to this link, we select one channel (Line 6) and schedule the link and those in the same maximum independent set (Line 7).

In the *Check_avail_chan* function at Line 4, we consider the two end nodes u and v associated with link l. Let the currently available channels on node u be NU and that on node v be NV. If the size of assigned channels $NC(u)$ to the node u is equal to the number of radios R, NU is set to $NC(u)$ since we can only choose them to meet the non-switching radio constraint; otherwise if smaller, NU is set to the available channels U at this time slot. Note that $|NC(u)|$ or $|NC(v)|$ cannot be greater than R in static channel assignment. Similar setting applies to NV. Thus the channels available to schedule link l are the intersection of NU, NV and U.

If the intersection is empty, we will see what constraint is violated and decide to resolve it or not (Line 5). As each OTSLS in dynamic schedule meets the last two constraints and we only choose the links activated in the dynamic schedule, the node-radio constraint is still satisfied if it is activated in static schedule. However, it is possible that the channel assigned on a link in dynamic schedule cannot be used by this link in static schedule due to non-switching radio constraint, and that the channel that satisfy the non-switching radio constraint may be used by some other interfering links at the same time slot in static schedule. Thus, we need to check whether non-switching radio constraint or the interference-channel constraint is violated.

If there exists a common channel in $NU \bigcap NV$, or at least one radio hasn't been assigned channels ($|NC(u)| < R$, or $|NC(v) < R|$), it means that the channels that satisfy the non-switching radio constraint are used by some other interfering links. When such channel constraint violation occurs, we cannot resolve it by grabbing the channel from other links at this time slot. The reasons are that this constraint may be resolved by adding more time slots, and that abusively grabbing the channel of other links will result in loops since some of the links has been scheduled only once so far and will be scheduled anyway. Otherwise, there are not any common channels in $NU \bigcap NV$ and all the associated radios have been assigned channels ($|NC(u)| = R$, and $|NC(v) = R|$), we should resolve this conflict otherwise this link cannot be scheduled at least once. In order to do this, we choose one of the assigned channels that is least used, and remove the assignment of the channel on the radio. This means that any link activated on such channel in the static schedule will be de-activated. So we will update CM_s, NC, and LM correspondingly. Note here the interference-channel constraint may still exist after resolving the non-switching radio constraint.

If there exists available channels to be assigned to this link, we select one channel (Line 6) and schedule the link and those in the same maximum independent set with non-zero flow requirement and fulfillment of non-switching radio constraint (Line 7). In selecting channels from available channels, we choose the most used channel by the end-point radios; if no such available, then the most used one by all the radios.

The above process stops when we couldn't find a not-scheduled link that has non-zero flow requirement (line 8). At the second stage (Lines 10 to 12), the objective is to maximize the coverings of each OTSLS if the channels and radios allow for more scheduling of the links based on the activation in the dynamic schedule. It is kind of continuation of the first stage, but the difference is that there is only interference-channel constraint violation. We do not need resolve the violations here.

The OTSLS sets LM now contains a schedule which meets all the constraint and maximize the covering up to the activations in the dynamic schedule. Our last stage (Lines 13 to 15) is to choose some of the OTSLSes to satisfy the flow requirement. It is similar to the dynamic channel assignment, we just select the OTSLS that maximizes the effective remaining coverings under flow requirement. For each link, the number of remaining covering is the difference between the required opportunity and the minimum of the required opportunity and total scheduled opportunity.

Algorithm 5: FSCA(MG,D_{opp})

Input: dynamic link scheduling matrix LM_d, Required opportunity D_{opp}
Output: static link scheduling matrix LM_s, channel assignment matrix CM_s

```
Initialize ()
while ∃D_opp(l) > 0 and S(l) == 0 do
    foreach dOTSS ∈ LM_d do
        if ∃l, dOTSS(l) > 0 and ∃D_opp(l) > 0 and S(l) == 0 then
            avail_chans ← Check_avail_chan(l, NC, U)
            if avail_chans = φ then
                [avail_chans, U, dOTSS, LM, CM_s, NC, S] ←
                Resolve(l, AL, S, CM_s, LM, dOTSS, U, NC, sTSsize)
            if avail_chans ≠ φ then
                which_chan ← SelectChannel(CM_s, avail_chans, NC, l)
                [U, dOTSS, LM, CM_s, NC, S] =
                assign_link_IS(mIS, l, NC, U, dOTSS, LM, CM_s, which_chan, S, D_opp)
        if ∃D_opp(l) > 0 and S(l) == 0 then
            break

foreach Result ∈ LM and the correspondent dOTSS ∈ LM_d do
    U ← whole − used_chan, dOTSS ← dOTSS − Result
    while U ≠ φ and ∃dOTSS(l) > 0 do
        l ← the link with maximum un-scheduled flow
        avail_chans ← Check_avail_chan(l, NC, U)
        if avail_chans = φ then
            dOTSS(l) ← dOTSS(l) − 1
        else
        which_chan ← SelectChannel(CM_s, avail_chans, NC, l)
        [U, dOTSS, LM, CM_s, NC, S] =
        assign_link_IS(mIS, l, NC, U, dOTSS, LM, CM_s, which_chan, S, D_opp)

LM_s = LM
while ∃l, S(l) < D_opp(l) do
    F ← set with most unscheduled links in LM
    LM_s ← LM_s ∪ F, S ← S + F
```

B. Link Scheduling for Video-type Application

For video-type applications with bandwidth requirements, 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. At the same time, the radio and channel constraints are satisfied.

Similarly, note that the problem of obtaining all the possible OTSLS is NP-hard. Our approach is based on the

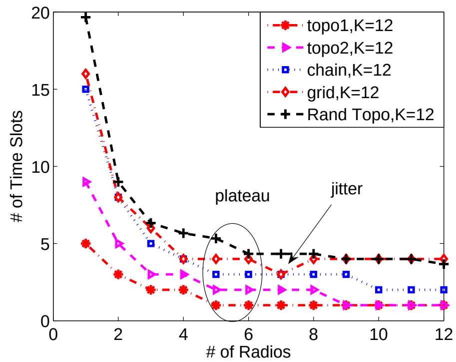


Fig. 5. Time for different topologies with different number of radios

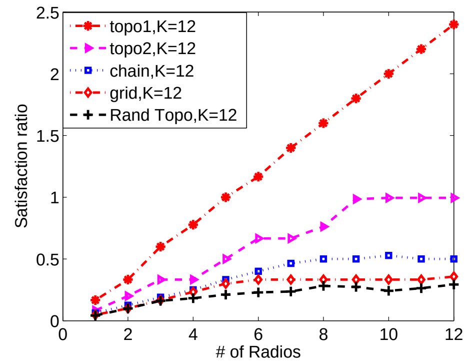


Fig. 6. Link utilization satisfaction ratio for different topologies with different number of radios

achieved result from algorithm VDCA and use the first and second stage of FSCA. We resolve the non-switching radio constraint and have all the links activated in the dynamic schedule being assigned a channel. Then we seek to maximize the capacity of each OTSLS in the second stage. We calculate the satisfaction ratio based on the same TDMA frame size as the dynamic schedule.

VI. Performance Evaluation

In this section, we evaluate the impact of the numbers of channels and radios as well as topology using the channel assignment and link scheduling algorithm for both application models. The 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 evenly distributed on a line. The grid topology is a 4*4 grid. For random topologies, we uniformly and randomly placed 20 nodes in $1000m * 1000m$ 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

In the following two sections, we describe the results from the dynamic channel assignment first.

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 observe that the number of times slots required to schedule all the flows decreases

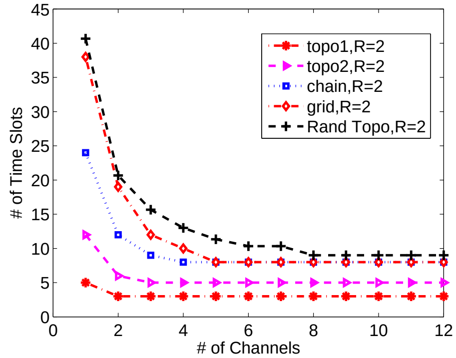


Fig. 7. Time for different topologies with different number of channels

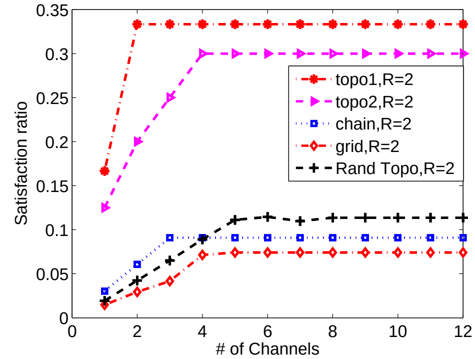


Fig. 8. Link utilization satisfaction ratio for different topologies with different number of channels

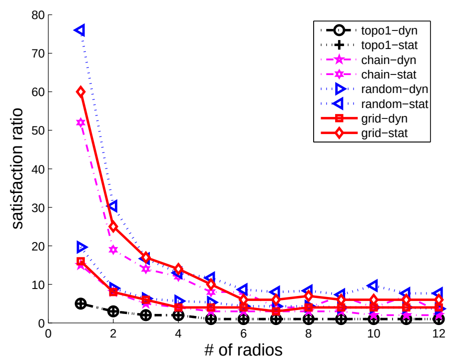


Fig. 9. Time for different topologies with different number of radios

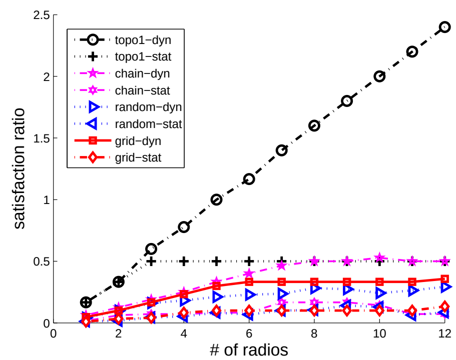


Fig. 10. Link utilization satisfaction ratio for different topologies with different number of radios

and the minimal link satisfaction ratio among all links increases as the number of radios increases. Second, 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 is due to 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.

We found similar trend in static channel assignment as in dynamic channel assignment from Fig. 9 and Fig. 10. We noticed that with our static channel assignment, it is possible to achieve the same minimum number of time slots with simple network topology and small flows as seen from the curve for topology 1 in Fig. 9. With static channel assignment, the advantage of using 2 radios is more obvious because the system degrades to use the same channel with 1 radio if all links have flows, which is seen from the steeper slope in Fig. 9. With more than 6 radios, static channel assignment achieves almost the same number of time slots as the dynamic one. This is because the number of channels 12 limits the increase of the performance of dynamic channel assignment. Even without switching, a relatively large number of radios will present good performance. However, as seen from Fig. 10, the gap does not

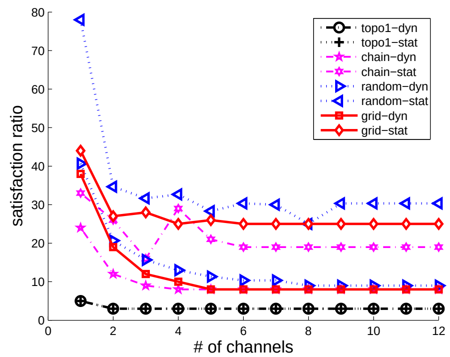


Fig. 11. Time for different topologies with different number of channels

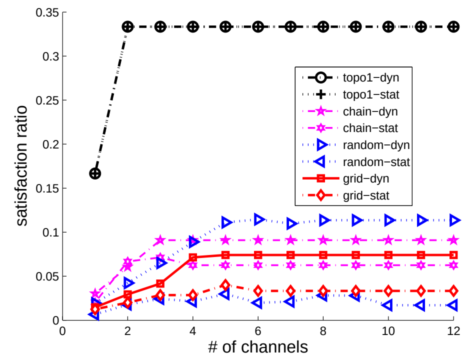


Fig. 12. Link utilization satisfaction ratio for different topologies with different number of channels

diminish with an increase of the number of radios. So we infer that dynamic channel assignment is more suitable for video-type applications.

B. Impact of Number of Channels and Topology

From the previous section, we observe that the improvements of adding a second radio are equal to or more than those of adding one more on 2 radios or more 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 from 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 cannot 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.

We also found similar trend in static channel assignment as in dynamic channel assignment from Fig. 11 and Fig. 12. With our static channel assignment, it is possible to achieve the same minimum number of time slots with simple network topology and small flows as seen from the curve for topology 1 in Fig. 10. With static channel assignment, the advantage of using 2 channels is also very obvious because too much interference will exist if using the same channel. However, with an increase in the number of channels, the gap between dynamic channel assignment and static channel assignment almost maintains as seen from Fig. 11 and Fig. 12. This is because 2 radios cannot consume too many channel whether dynamic or static channel assignment is used.

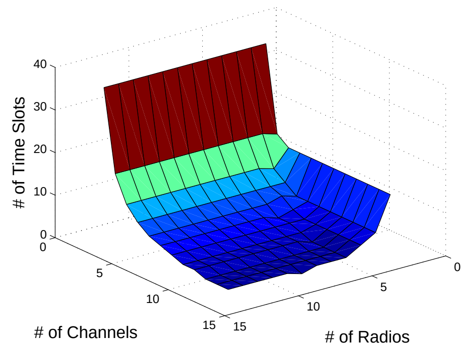


Fig. 13. Time for grid topology with various number of radios and channels

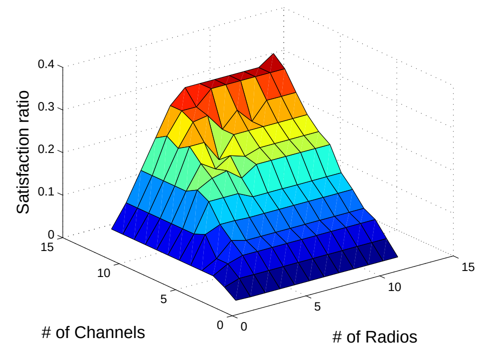


Fig. 14. Link utilization satisfaction ratio for grid topology with various number of radios and channels

C. Relationship between Number of Radios and Channels

In this section, we study the relationship between the number of radios and channels. Due to the space limit, we only present the result of the dynamic channel assignment here. We vary the number of radios and channels from 1 to 12 to get various combinations of number of radios and channels. Fig. 13 and Fig. 14 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. 13 and Fig. 14 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 can help considerably for video-type applications, but to a less extent for ftp-type applications.

D. Performance Comparison with Bounds

We have seen the performance difference between dynamic and static channel assignment, and the performance of dynamic channel assignment bounds the static one. In this section, we compare the performance of our algorithm of the dynamic channel assignment with the bounds we derived in Sections IV-B and IV-C. As observed from Fig. 15, 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. 16, 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

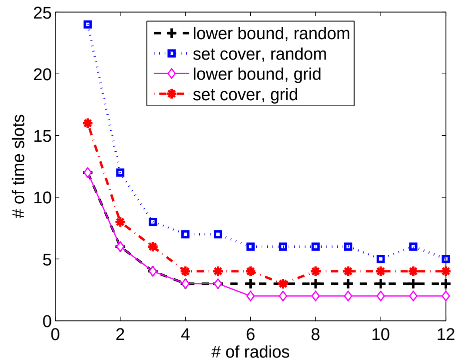


Fig. 15. Lower Bound for random and grid topology with different number of radios

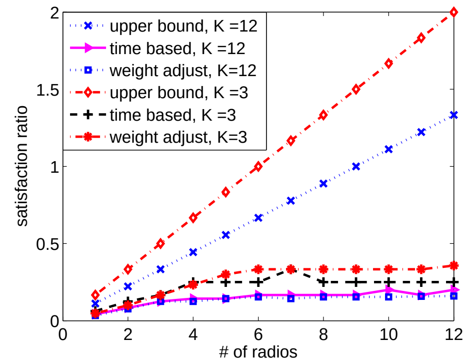


Fig. 16. Upper Bound Link utilization satisfaction ratio for random and grid topology with different number of radios

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.

VII. Conclusion

In this work, we propose both dynamic and static channel assignment and link scheduling algorithms for a given topology with multiple channels and radios under two different traffic demands. For any given traffic pattern, we provide the bounds for the algorithms. We then analyze the impact of the number of radios and channels as well as the topology on system performance. 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 a less extent for ftp-type applications. For future work, we will consider the problem of how to non-uniformly distribute radios to fully utilize the available channels or to satisfy the traffic demand 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] A. K. Das, H. M. K. Alazemi, R. Vijayakumar, and S. Roy. Optimization models for fixed channel assignment in wireless mesh networks with multiple radios. In *IEEE SECON*, 2005.

[7] M. Gong, S. Midkiff, and S. B. Mao. A combined proactive routing and multi-channel mac protocol for wireless ad hoc networks. In *Broadnet*, 2005.

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

[9] Intel. The CoMo project. Available: http://como.intel-research.net/.

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

[11] S. Liese, D. Wu, and P. Mohapatra. Experimental characterization of an 802.11b wireless mesh network. Technical report, univeristy of california at davis, 2005.

[12] X. Lin and S. Rasool. A distributed joint channel-assignment, scheduling and routing algorithm for multi-channel ad hoc wireless networks. In *Infocom*, 2007.

[13] 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.

[14] R. Maheshwari, H. Gupta, and S. Das. Multichannel mac protocols for wireless networks. In *SECON*, 2006.

[15] M. Marina and S. Das. A topology control approach to channel assignment in multi-radio wireless mesh networks. In *Broadnets*, 2005.

[16] 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.

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

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

[19] H. M. Rad and V. Wong. Joint channel allocation, interface assignment and mac design for multi-channel wireless mesh networks. In *InfoCom*, 2007.

[20] K. Ramachandran, E. Belding, K. Almeroth, and M. Buddhikot. Interference-aware channel assignment in multi-radio wireless mesh networks. In *IEEE Infocom*, 2006.

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

[22] 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.

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

[24] M. Shin, S. Lee, and Y. Kim. Distributed channel assignment in multi-radio 802.11 mesh networks. In *MASS*, 2006.

[25] 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.

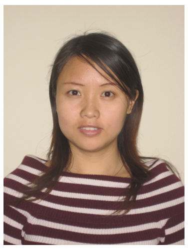
[26] A. Subramanian, R. Krishnan, S. R. Das, and H. Gupta. Minimum interference channel assignment in multi-radio wireless mesh networks. In *Secon*, 2007.

[27] J. Tang, G. Xue, and W. Zhang. Interference-aware topology control and qos routing in multi-channel wireless mesh networks. In *MobiHoc*, 2005.

[28] 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.

[29] 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.

[30] H. YU, P. Mohapatra, and X. Liu. Dynamic channel assignment and interference-free link scheduling in multi-radio multi-channel wireless mesh networks. In *Mobiquitous*, 2007.



Hua Yu is a Ph.d. student in the Department of Computer Science at the university of California, davis. She received B.E. and M.S. degrees in Computer Science from the University of Science and Technology and Fudan University in 2000 and 2003, respectively. Her research interests are in the area of wireless mesh networks, with the focus on resource management and protocol designs. She is a student member of IEEE.

Dr. Prasant Mohapatra is currently a Professor in the Department of Computer Science at the University of California, Davis. In the past, he was on the faculty at Iowa State University and Michigan State University. Dr. Mohapatra received his Ph.D. in Computer Engineering from the Pennsylvania State University in 1993. Dr. Mohapatra's research interests are in the areas of wireless networks, sensor networks, Internet protocols and QoS. He was/is on the editorial board of the IEEE Transactions on computers, IEEE Transactions on Parallel and Distributed Systems, ACM WINET, and Ad Hoc Networks. He has been on the program/organizational committees of several international conferences. He was the Program Vice-Chair of INFOCOM 2004, Program Co-Chair of QSHINE 2006, and the Program Co-Chair of the First IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON 2004). He has been a Guest Editor for IEEE Network, IEEE Transactions on Mobile Computing, IEEE Communications, and the IEEE Computer.

Dr. Xin Liu received her Ph.D. degree in electrical engineering from Purdue University in 2002. She is currently an assistant professor in the Computer Science Department at the University of California, Davis. Before joining UC Davis, she was a postdoctoral research associate in the Coordinated Science Laboratory at UIUC. Her research is on wireless communication networks, with a focus on resource allocation and dynamic spectrum management. She received the Best Paper of year award of the Computer Networks Journal in 2003 for her work on opportunistic scheduling. She received NSF CAREER award in 2005 for her research on "Smart-Radio-Technology-Enabled Opportunistic Spectrum Utilization." She received the Outstanding Engineering Junior Faculty Award from the College of Engineering, University of California, Davis in 2005.