This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2009 proceedings

Channel-Assignment and Scheduling in Wireless Mesh Networks considering Switching Overhead

Mira Yun[1], Yu Zhou[1], Amrinder Arora[2], and Hyeong-Ah Choi[1]
[1]Department of Computer Science, George Washington University, Washington, DC
[2]Department of Computer Science, Bowie State University, Bowie, MD
Email: {mirayun, yuzhou, hchoi}@gwu.edu, aarora@bowiestate.edu

***Abstract*—This paper considers the channel-assignment and scheduling in wireless mesh networks that employ multiple radios and multiple channels. In contrast to the various algorithms available in the literature, we explicitly model the delay overhead that is incurred during channel switching, and use that delay in the design of algorithms. We prove that the well known Greedy Maximal Scheduling (GMS) algorithm does not have any provable efficiency ratio when the switching overhead is considered. We present a centralized algorithm (CGSSO), and a dynamic algorithm (DMSSO), both of which consider switching overhead. Simulation results show that the proposed algorithms significantly outperform other algorithms in packet throughput and average packet delay metrics. Results also show that the improvements in performance become more pronounced as the switching delay increases.**

I. Introduction

Wireless Mesh Networks (WMNs), envisioned as one of the key infrastructures in the next generation wireless networks, can provide more reliable data transmissions and significantly improve the throughput with multi-radio multi-channel architecture [1]. In a multi-radio multi-channel environment, a proper assignment of channels to interfaces is a critical factor in optimizing the network interference and throughput. Various channel assignment strategies are reported in the literature (a summary can be found in [2]), which is divided into three main categories: fixed [3], [4], [5], dynamic [6], and hybrid [4], [7], [8]. Fixed strategy imposes static channel assignment to radio interfaces whereas dynamic strategy changes the assignment as needed. Hybrid strategies combine static and dynamic assignment concepts by using static assignment for certain nodes/radios, and using a dynamic assignment approach for the other nodes.

Among the centralized algorithms, an optimum scheduling policy was given by Tassiulas et al. in their seminal paper [9]. This scheduling policy, commonly referred to as Maximum Weighted Scheduling, is computationally prohibitive for general interference models (such as 2-hop interference model), and a simpler but suboptimal strategy called Greedy Maximal Scheduling (GMS) is well established as a scheduling algorithm for single channel multi-hop wireless networks. GMS has been known to have an efficiency ratio of $1/\kappa$ for single-channel networks in [10] and $1/(\kappa + 2)$ in multiple-channel networks. Very recently, insights into the true efficiency ratio of GMS have been presented in [11], where the authors have shown that the efficiency ratio of GMS is equal to a network property (pooling factor), and the authors also show that the worst-case efficiency ratio of GMS in geometric network graphs is between $\frac{1}{6}$ and $\frac{1}{3}$.

In distributed algorithms, Joo [12] proposed a simple distributed scheduling algorithm for single channel wireless networks, which achieves an efficiency ratio no smaller than GMS. In [13], Lin et al. provide a distributed algorithm for multi-channel network with low-complexity that has the same level of efficiency ratio as GMS. Ko et al. [14] also proposed a distributed channel assignment algorithm with channel interference cost function which indicates the spectral overlapping level between channels. In addition, much of the recent research on multi-radio multi-channel WMNs has solved the channel assignment and routing problem jointly as a challenging cross-layer problem [13], [15], [16]. In [13], a distributed channel scheduling algorithm is proposed, which guarantees the same efficiency ratio as the centralized GMS algorithm in multi-channel wireless networks.

It is however interesting to note that none of these algorithms consider the overhead incurred from switching radios dynamically from one channel to another. Some of them do mention delay, but simply assume that the switching delay can be reduced and made negligible by improving hardware technology and refining protocols. Our simulation results, however show that the actual performance of these algorithms can be much lower than expected when actual switching delay is injected into each time slot. In this paper, motivated by this observation, we study the channel assignment and scheduling in multi-radio multi-channel mesh networks and develop algorithms that take the switching delay into account. Simulation results show that the performance of our algorithms significantly improve over the other existing ones.

The rest of the paper is organized as follows. In the next section, we discuss the role of switching overhead and explain why it is necessary to consider the effects. In Section III, we outline the system model and give a problem statement. In Section IV, we first present an extension of the well-known Greedy Maximal Scheduling taking the switching delay into account in the channel assignment and then present a distributed algorithm that considers the switching delay as well. Simulation results are shown in Section V, and our conclusions and some directions for future work are presented in Section VI.

978-1-4244-3435-0/09/$25.00 ©2009 IEEE

II. Switching Overhead - Negligible or Not?

Many algorithms that try to optimize the efficiency of WMNs use a strategy that requires frequent change in channel assignment, which leads to accumulation of switching delays between end to end nodes. This delay happens at multiple levels. Hardware delays occur in the physical layer where as software delays in the upper layer for synchronization and scheduling.

For a typical 802.11 card, the hardware switching delay can vary from a few hundreds of microseconds to a few milliseconds [6], [17], [18]. When a packet of 1024 bytes is transmitted through 802.11a network where the typical transmission rate is about 25Mbps, it takes $1024 * 8/(25 * 10^6) = 328\mu s$ which is in the same range of 802.11a switching delay. In the 802.11b network in which the typical transmission rate is about 6Mbps, it takes $1024 * 8/(6 * 10^6) = 1.3ms$. The impact of switching delay on the overall network performance becomes even more significant when switching occurs across different frequency bands. For example, when a Mesh Router has two radios such as 802.11a card (operating on 5GHz band) and 802.11b/g card (operating on 2.4 GHz band), switching between two bands is possible with the delay much larger than the delay from switching within the same band.

The switching delay not only affects the end to end delay but also degrades network capacity. According to [4], the switching delay degrades the network capacity as a function of $S/S+T$ (where S is switch delay and T is transmission time). As in the example above, the value of S can approach the value of T. This causes a significant degradation in network capacity.

One reason that is sometimes given to ignore the switching delay overhead is that with technological advancements, the switching delay will become smaller overtime. However, the switching delay can be expressed in terms of packet duration as $d_t \times$ transmission speed/packet size, where d_t is the hardware switching delay. Thus, while the hardware switching delay can indeed be expected to progressively get smaller, the transmission speeds can be expected to progressively get larger. Thus, the trend on the overall loss of bandwidth due to the switching delay is difficult to predict due to this "push-pull" effect of technology. This highlights the need to design channel assignment schemes that consider the delay induced due to the switching overhead, and to model their performance as a function of the switching overhead.

III. System Model and Problem Statement

In this section, we first describe a network model considered in this paper. Then, we present a formulation of our problem for channel assignment and scheduling.

A. System Model

We consider a time slotted multi-hop network modeled by an undirected graph $G = (V, E)$ where V denotes a set of nodes and E denotes the set of edges. For a link $l \in E$ when used in a transmission, the transmitter and the receiver nodes are denoted by $b(l)$ and $e(l)$, respectively. For a node $v \in V$, the set $E(v)$ denotes the set of edges incident on node v. Let C denote the set of channels available in the system. Time is slotted into a unit length. Each node is equipped with at least one radio and can dynamically switch radios from one channel to another with additional overhead δ represented as a fraction of the time slot duration, i.e., $\delta = switching\ time/time\ slot\ duration$. It is assumed that each node v is equipped with $\alpha(v) \leq k$ radios such that at any time, v can be involved in up to $\alpha(v)$ transmissions as either transmitters or receivers.

Let I_l denote the set of links that interfere with link l. It is assumed that during the scheduling period (i.e., over a certain period of time), the network topology is fixed; hence, the interfering set I_l of link l is also fixed. We assume that the interference relation is symmetrical. We denote the queue length of link l in time slot t by $q(l,t)$ where $l = (b(l), e(l))$ is treated as a directional link. The rate at which link l can transmit on channel c is denoted by $r(l,c)$.

B. Problem Statement

There are S users in the system. We assume that user s injects packets into the system with a rate λ_s and traffic from s follows a fixed path during the scheduling period. (The routing table is assumed to be fixed during the scheduling period). Let $h(l,s) = 1$ if user s's traffic traverses over link l, and 0 otherwise. The evolution of $q(l,t)$ is then:

$$q(l,t+1) \quad = \quad [q(l,t) + \sum_{s=1}^{S} h(l,s)\lambda_s - D(l,t)]^+ \quad (1)$$

where $D(l,t)$ denotes the number of packets that link l can serve in time t and $[\cdot]^+$ denotes the projection to $[0,\infty)$. We say the system is stable if the queue length of each link in any time slot remains finite.

Let $\vec{\lambda} = [\lambda_1, \cdots, \lambda_S]$ denote the traffic injected by the S users into the system. The *capacity region* under a particular channel assignment and scheduling algorithm is the set of $\vec{\lambda}$ such that the system remains stable. The *optimal capacity region* Ω is defined to be the union of capacity regions of all algorithms. An algorithm is called *throughput-optimal* if it can achieve the optimal capacity region Ω. The *efficiency ratio* of an algorithm is the largest number γ such that for any load $\vec{\lambda} \in \Omega$, $\gamma\vec{\lambda}$ is in capacity region of the algorithm.

A major component of any throughput-optimal scheduling problem is to solve an optimization problem in each time slot t that maximizes $\sum_{l\in E}\sum_{c\in C} q(l,c,t)\, r(l,c,t)$ satisfying the given constraints. With the switching delay δ as an additional constraint, we formulate the scheduling problem as follows where $z(l,c,t) \in \{0,1\}$ is a decision variable such that $z(l,c,t) = 1$ means that channel $c \in C$ is assigned to link $l \in E$ in time slot t.

Scheduling with δ:
Input: $Z(t-1) = [\ z(l,c,t-1)\ ]$ for all $l \in E$ and $c \in C$; $q(l,t)$ for all $l \in E$; and $r(l,c,t)$ for all $l \in E$ and $c \in C$.
Output: $Z(t) = [\ z(l,c,t)\ ]$ where (i) $z(i,c,t) \in \{0,1\}$, (ii) for any l, $l' \in E$ such that $l' \in I_l$ and $c \in C$,

$z(l,c,t) + z(l',c,t) \leq 1$, (iii) for any $l \in E$, $\sum_{c \in C} z(l,c,t) \leq \min\{\alpha(b(l)), \alpha(e(l))\}$, and satisfying (i-iii), the objective is to maximize

$$\sum_{l \in E, c \in C} \{z(l,c,t)\, q(l,t)\, r(l,c,t) \mid z(l,c,t-1) = 1\}$$
$$+ \sum_{l \in E, c \in C} \{z(l,c,t)\, (1-\delta)\, q(l,t)\, r(l,c,t) \mid z(l,c,t-1) = 0\}$$

Note that if $z(l,c,t-1) = 1$ and $z(l,c,t) = 1$, the channel c can be fully utilized on link l during the time slot t. But if $z(l,c,t-1) = 0$, the channel c when assigned to l in time t can be utilized for only a fraction $1-\delta$ of the time slot.

C. Limitations of GMS when Considering Switching Overhead

As mentioned in the introduction, there are numerous results on the efficiency ratio of the GMS algorithm. Most recently, it has been shown in [11] that the worst-case efficiency ratio of GMS in geometric network graphs is between $\frac{1}{6}$ and $\frac{1}{3}$. However, we show that when switching overhead is considered, GMS algorithm may have no provable efficiency ratio, even when considering single hop interference model in geometric network graphs.

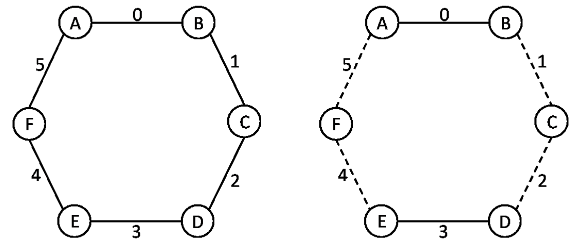
Theorem 1. *Considering the switching overhead, the efficiency ratio of GMS algorithm can be made arbitrarily close to* 0.

Proof: We prove this by using a counter example. Consider the graph as shown in Figure 1(a). We assume that the traffic moves in clockwise direction, all link capacities are 3 and the initial queue size is $S+2$ at nodes A and D, $S+1$ at nodes B and E, and S at nodes C and F, where S is a large number. We show that if all nodes have a constant arrival rate of $1-\delta+\epsilon$, then the network is not stable under GMS.

We observe, that during the first timeslot, GMS activates links 0 and 3 as shown in Figure 1(b). As $1-\delta$ portion of the timeslot can be used for data transmission, at the end of the first timeslot, the queue sizes are: $S+2\delta+\epsilon$ at nodes A and D, $S+2-\delta+\epsilon$ at nodes B and E, and $S+1-\delta+\epsilon$ at nodes C and F. GMS activates links 1 and 4 for the timeslot 2, and at the end of timeslot 2, the queue sizes are: $S+1+\delta+2\epsilon$ at nodes A and D, $S+\delta+2\epsilon$ at nodes B and E, and $S+2-2\delta+2\epsilon$ at nodes C and F. For timeslot 3, GMS activates links 2 and 5, and at the end of timeslot 3, the queue sizes are: $S+2+3\epsilon$ at nodes A and D, $S+1+3\epsilon$ at nodes B and E, and $S+3\epsilon$ at nodes C and F. So, after 3 timeslots, queues have increased by $3\,\epsilon$. Thus, with this offered load, the system is unstable under GMS.

However, it is easy to observe that the following schedule can serve the same system with an arrival rate of $3/2-\delta$: Consider two link assignments: $\{0,2,4\}$ or $\{1,3,5\}$. Start with either one, and switch between the two every $S/2$ timeslots.

Therefore, the efficiency ratio of GMS is no better than $\frac{1-\delta}{3/2-\delta}$. Since δ can be made as close to 1 as necessary, this efficiency ratio can be made arbitrarily close to 0. □



(a) Topology (b) Maximal schedule

Fig. 1. Example topology and schedule with GMS

IV. ALGORITHMS

In this section, we propose two channel scheduling algorithms with considering switching overhead as centralized and distributed control algorithms respectively. In order to show how we extend the existing algorithms with switching overhead, we describe our algorithm beginning with the existing algorithm.

A. Centralized Greedy Scheduling with Switching Overhead (CGSSO) Algorithm

Among the centralized channel assignment algorithms, the Greedy Maximal Scheduling (GMS) algorithm has been considered as one of the most well-known sub-optimal scheduling algorithms [11]. In this subsection, we describe the GMS algorithm and our extended algorithm (CGSSO) that considers switching overhead for multi-channel multi-radio wireless networks.

In Greedy Maximal Scheduling, a queue-weighted rate $w(l,c,t)$ is defined as $w(l,c,t) = q(l,t)\, r(l,c,t)$. GMS starts with the link-channel pair (l,c) with the largest $w(l,c,t)$. After finding the largest weight $w(l,c,t)$, it removes all link-channel pairs that cannot be scheduled due to (l,c) being scheduled. In other words, it removes all link-channel pairs (k,c) with $k \in I_l$. If $\alpha(l) = 0$, which means link l already uses up all available radio interfaces, it removes all link-channel pairs (k,c^*) with $k \in E(b(l)) \cup E(e(l))$. With the remaining pairs, it continues to find the largest weight until no link-channel pairs are left.

Considering switching overhead, our extended algorithm CGSSO is summarized in Algorithm 1. In order to consider the switching overhead δ, we provide two different weight functions depending on whether or not the switching is needed. At each time t, we know the set $\mathcal{Z}$ of all scheduled link-channel pairs (l,c) at time $t-1$. For each link-channel pair in set $\mathcal{Z}$, we define $w(l,c,t) = q(l,t)\, r(l,c)$. For each link-channel pair not in $\mathcal{Z}$, we define $w(l,c,t) = (1-\delta)\, q(l,t)\, r(l,c)$. Using this modified weight function, we follow the GMS algorithm as defined above to get the schedule of links and channels at each timeslot.

B. Distributed Maximal Scheduling with Switching Overhead (DMSSO) Algorithm

In [13], a distributed joint channel-assignment, scheduling, and routing algorithm (referred here as "DASP") is proposed.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2009 proceedings

Algorithm 1 Centralized Greedy Scheduling with Switching Overhead (CGSSO)

For each time-slot t:
Let $\mathcal{F} = \{(l,c) | l \in E, c \in C\}$, $\mathcal{Z} = \{(l,c) | z(l,c,t-1) = 1\}$
Initialize $\beta(v) \leftarrow \alpha(v)$ for all nodes v
For $\mathcal{Z}$, define $w(l,c,t) = q(l,t) r(l,c)$.
For $\{\mathcal{F} - \mathcal{Z}\}$, define $w(l,c,t) = (1-\delta) q(l,t) r(l,c)$
while $size(\mathcal{F}) > 0$ **do**
  In $\mathcal{F}$, find (l,c) with the largest weight $w(l,c,t)$
  $z(l,c,t) \leftarrow 1$
  $\beta(b(l)) \leftarrow \beta(b(l)) - 1$
  $\beta(e(l)) \leftarrow \beta(e(l)) - 1$
  for $k \in I_l$ **do**
    remove (k,c) from $\mathcal{F}$
  end for
  if $\beta(b(l)) = 0$ **then**
    for $k \in E(b(l))$ **do**
      Remove (k,c') from $\mathcal{F}$ for all channels c'
    end for
  end if
  if $\beta(e(l)) = 0$ **then**
    for $k \in E(e(l))$ **do**
      Remove (k,c') from $\mathcal{F}$ for all channels c'
    end for
  end if
end while

In order to show the impact of switching overhead on the WMN throughput we extend their algorithm by considering switching overhead. Without the switching overhead, DASP can be summarized as follows. For each time t,

1) Define $x(l,c,t)$ to be the number of packets that link l can assign to channel c at time t.
For each link l, $x(l,c,t)$ can be assigned as follows.

$$x(l,c,t) = \begin{cases} r(l,c), & \text{if } \frac{q(l)}{\zeta_l} \geq \frac{1}{r(l,c)} \Big[\sum_{k \in I_l} \frac{\eta(k,c,t)}{r(k,c,t)} \\ & + \frac{1}{\alpha(b(l))} \sum_{k \in E(b(l))} \sum_{d=1}^{C} \frac{\eta(k,d,t)}{r(k,d,t)} \\ & + \frac{1}{\alpha(e(l))} \sum_{k \in E(e(l))} \sum_{d=1}^{C} \frac{\eta(k,d,t)}{r(k,d,t)} \Big] \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

ζ_l is an arbitrary positive constant chosen for link l. The per-channel queue $\eta(l,c,t)$ represents the backlog of packets assigned to channel c by link l. From $q(l)$, the number of packets assigned to each channel queue η_l^c is $y(l,c,t) \in [0, x(l,c,t)]$, where $\sum_{c=1}^{C} y(l,c,t) = min\{q(l,t), \sum_{c=1}^{C} x(l,c,t)\}$.

2) Based on the channel queues $(\eta(l,c,t) + y(l,c,t))$, Multi-channel Maximal Scheduling (with LubyMIS algorithm) is carried out. We define $Z^c(t)$ as the set of non-interfering links that are chosen to transmit data at channel c at time t, i.e., $Z(t) = [Z^c(t)]$. For each channel c, $Z^c(t)$ consists of links l that are backloged in channel c, i.e., $\eta(l,c,t) + y(l,c,t) \geq r(l,c)$. Further, for any backloged link-channel pairs (l,c), at least one of the following is true.
  a) Either link l is scheduled in channel c, i.e., $l \in Z^c(t)$, or
  b) Either link k is scheduled in channel c, i.e., $k \in Z^c(t)$ for some backloged $k \in I_l$, or
  c) Either the transmitter or the receiver of link l has used up all the radios.

Considering switching overhead, the proposed algorithm (DMSSO) is summarized in Algorithm 2.

Algorithm 2 Distribute Maximal Scheduling with Switching Overhead (DMSSO)

For each time-slot t,
let $\mathcal{F} = \{(l,c) | l \in E, c \in C\}$, $\mathcal{Z} = \{(l,c) | z(l,c,t-1) = 1\}$
For $\mathcal{Z}$, $x(l,c,t) = r(l,c)$ or 0 based on (3)
For $\{\mathcal{F} - \mathcal{Z}\}$, $x(l,c,t) = (1-\delta)\, r(l,c)$ or 0 based on (4)
for each link l and channel c **do**
  Assign $y(l,c,t) \in [0, x(l,c,t)]$
  where $\sum_{c=1}^{C} y(l,c,t) = min\{q(l,t), \sum_{c=1}^{C} x(l,c,t)\}$
end for
for each channel c **do**
  find $Z^c(t)$ by calling LubyMIS(G, c);
end for

At each time t, we know the set $\mathcal{Z}$ of all scheduled link-channel pairs (l,c) scheduled at time $t-1$

1) Define $x(l,c,t)$ to be the number of packets that link l can assign to channel c at time t.
For the set $\mathcal{Z}$, $x(l,c,t)$ can be assigned as follows.

$$x(l,c,t) = \begin{cases} r(l,c), \text{ if:} \\ \quad \frac{q(l)}{\zeta_l} \geq \frac{1}{r(l,c)} \Big[\sum_{k \in I_l} \frac{\eta(k,c,t)}{r(k,c)} \\ \quad + \frac{1}{\alpha(b(l))} \sum_{k \in E(b(l))} \sum_{d=1}^{C} \frac{\eta(k,d,t)}{r(k,d)} \\ \quad + \frac{1}{\alpha(e(l))} \sum_{k \in E(e(l))} \sum_{d=1}^{C} \frac{\eta(k,d,t)}{r(k,d)} \Big] \\ 0 \quad \text{otherwise} \end{cases} \quad (3)$$

For the set $\{\mathcal{F} - \mathcal{Z}\}$, $x(l,c,t)$ can be assigned as follows.

$$x(l,c,t) = \begin{cases} (1-\delta)\, r(l,c), \text{ if:} \\ \quad \frac{q(l)}{\xi_l} \geq \frac{1}{r(l,c)} \Big[\sum_{k \in I_l} \frac{\eta(k,c,t)}{r(k,c)} \\ \quad + \frac{1}{\alpha(b(l))} \sum_{k \in E(b(l))} \sum_{d=1}^{C} \frac{\eta(k,d,t)}{r(k,d)} \\ \quad + \frac{1}{\alpha(e(l))} \sum_{k \in E(e(l))} \sum_{d=1}^{C} \frac{\eta(k,d,t)}{r(k,d)} \Big] \\ 0 \quad \text{otherwise} \end{cases} \quad (4)$$

ζ_l and ξ_l are the arbitrary positive constants chosen for link l. From $q(l)$, the number of packets assigned to each channel queue η_l^c is $y(l,c,t) \in [0, x(l,c,t)]$, where $\sum_{c=1}^{C} y(l,c,t) = min\{q(l,t), \sum_{c=1}^{C} x(l,c,t)\}$.

2) Based on the channel queues $(\eta(l,c,t) + y(l,c,t))$, Multi-channel Maximal Scheduling is carried out. We define $Z^c(t)$ as the set of non-interfering links that are chosen to transmit data at channel c at time t, i.e., $Z(t) = [Z^c(t)]$. For each channel c, $Z^c(t)$ consists of links l that are backloged in channel c, i.e.,

$\eta(l,c,t) + y(l,c,t) \geq r(l,c)$. And we give higher priority to the set $\mathcal{Z}$ backloged again. For any remaining backloged link-channel pairs (l,c), at least one of the following is true.

a) Either link l is scheduled in channel c, i.e., $l \in Z^c(t)$, or
b) Either link k is scheduled in channel c, i.e., $k \in Z^c(t)$ for some backloged $k \in I_l$, or
c) Either the transmitter or the receiver of link l has used up all the radios.

In order to implement Multi-channel Maximal Scheduling Algorithm, we use the Luby Maximal Independent Set (LubyMIS) algorithm for each channel c, as described in [19]. The algorithm consists of three rounds. In the first round, all links update their weight $w(l,c,t)$ and send to their interference neighbors. If $(l,c) \in \mathcal{Z}$, $w(l,c,t) = (\eta(l,c,t) + y(l,c,t))\, r(l,c)$. Otherwise $w(l,c,t) = (1-\delta)\,(\eta(l,c,t) + y(l,c,t))\, r(l,c)$. By the end of the first round, links with highest weights (among their respective interference sets) are marked as the winners. In the second round, all winners notify their interference neighbors the fact that they have won. Thus at the end of second round, the interference neighbors know that they are the losers. In the third round, all losers notify their neighbors. Then all the winners, the losers, and the losers' neighbors remove the appropriate nodes and links from the graph G. After the third round, the algorithm repeats from the first round to find the winners, the losers, and the losers' neighbors with remaining nodes and links. This process is repeated until no links are left in G. Finally, LubyMIS returns the set $Z^c(t)$, consisting of the winners.

C. Stability Analysis

We prove in this section that the efficiency ratio of the proposed DMSSO algorithm is $(1-\delta)/(\kappa+2)$, where κ is the interference degree of the network.

Proof: We show that for any $\overrightarrow{\lambda}$, such that $\overrightarrow{\lambda}(\kappa+2)/(1-\delta)$ can be served by a scheduling algorithm, then $\overrightarrow{\lambda}$ can be served by DMSSO.

As outlined in [13], one key to observing this is first note that there must exist some $\tilde{x}(l,c) \in [0, r(l,c)]$ such that:

$$\frac{(1+\epsilon)^2(\kappa+2)}{1-\delta}\sum_{s=1}^{S} H_s^l \lambda_s \leq \sum_{c=1}^{C} \tilde{x}(l,c),\ \forall \text{ links } l \quad (5)$$

$$\sum_{k \in I_l} \frac{\tilde{x}(k,c)}{r(k,c)} \leq \kappa \quad (6)$$

$$\sum_{k \in E(i)} \sum_{c=1}^{C} \frac{\tilde{x}(k,c)}{r(k,c)} \leq \alpha(i) \quad (7)$$

These 3 equations come from the long term average service $\tilde{x}(l,c)$ that a link l can receive on channel c under the stability requirement (5), interference constraint (6) and constraint on the number of radios (7). Using the same Lyapunov function and the techniques outlined in [13], we observe that the results follow as in [13]. □

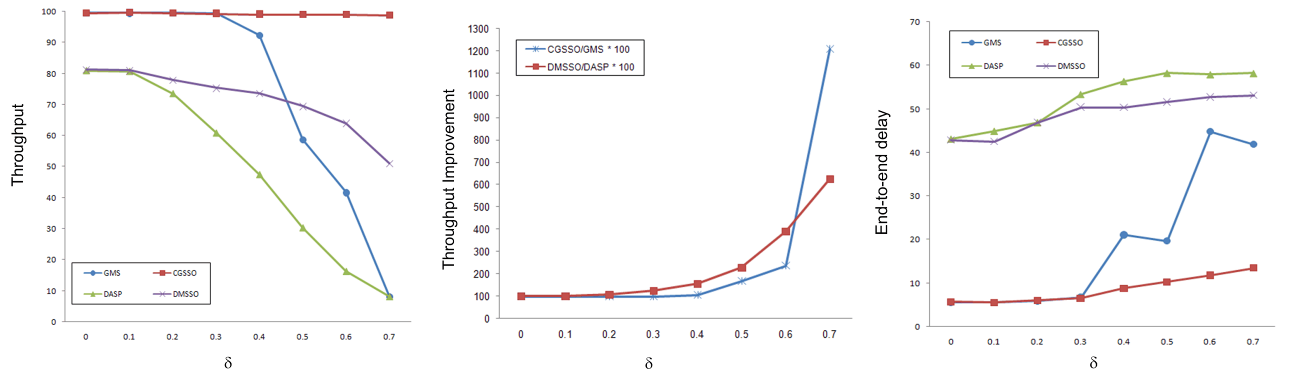
We also observe that the proven efficiency ratio of the proposed DMSSO algorithm is by definition less than the frequency ratio of the DASP algorithm, that is due to the fact that the switching delay has not been taken into account in case of the optimal algorithm. In fact, when we do compare the simulation results of DMSSO and DASP algorithms, it is clear that DMSSO algorithm outperforms the other algorithms significantly.

V. Simulation Results

We evaluate the performance of the channel assignment algorithms. We compare their system throughput and end-to-end delay under 2-hop interference models. Our simulation runs on a 5×5 grid topology where each node can potentially communicate with up to four neighbors. The nodes present in the simulation are assumed to be equipped with multiple radios, and a dynamic range of capacity can be considered. We randomly select capacity for each channel for each link from the range [10, 14]. Then we randomly select ten source-destination pairs for each simulation time (1000 time-slots). Scheduling occurs at every time-slot. To consider the switching delay, each time-slot in the simulation time is divided into ten mini-time-slots. Each source node generates Poisson traffic with average inter-arrival time of one mini-time-slot. The number of radios on each node varies from 2 to 4, which includes one default radio to maintain topology. We assume each radio has 13 non-overlapping channels, and one of those channels is used as the default for the default radio. During simulation time, the routing table is fixed and any underlying routing algorithm can be used independently of scheduling algorithm.

Fig. 2(a) compares the throughput for different algorithms varying switching overhead δ. We define the throughput as the ratio between the total received packets and the total sent packets. As shown in Fig. 2(a), the throughput of GMS and DASP algorithms (both implemented without considering switching overhead) decreases dramatically when δ is larger than 0.3 and 0.1 respectively. However our centralized algorithm CGSSO has almost the same performance with varying δ. And even though the throughput of DMSSO shows a decrease, compared to DASP, it shows dramatic improvement in stability. In order to show the improvement of throughput, we provide Fig. 2(b). As shown in Fig. 2(b), our algorithms can achieve seven to twelve times of throughput when δ is 0.7.

Fig. 2(c) presents the end-to-end delay (time-slots) with varying δ switching overhead. Because the throughput of DMSSO is significantly higher, even though the end-to-end delay appears to be similar between DASP and DMSSO, DMSSO is a significant improvement over DASP. Likewise, CGSSO shows a vast improvement over GMS. One interesting observation is the drop in delay in GMS as switching delay δ changes from 0.4 to 0.5 and 0.6 to 0.7. However, when considering the fact that the throughput dropped considerably at those intervals, those drops in delay do not signify improvement in end-to-end delay.



(a) The average throughput of scheduling algorithms (b) The throughput improvement of scheduling algorithms (c) The average end-to-end delay of scheduling algorithms

Fig. 2. Simulation Results

VI. Conclusion

In this paper, we considered the channel-assignment and scheduling in wireless mesh networks that employ multiple radios and multiple channels. In contrast to the various algorithms that had been available before this work, we explicitly modeled the delay overhead that is incurred during channel switching, and used that delay in the design of algorithms. We presented both centralized and dynamic algorithms for channel assignment and scheduling problem, and presented detailed simulation results that show that the proposed algorithms significantly outperform the prior known algorithms in packet throughput and average packet delay metrics. Results also showed that the improvements in performance become more pronounced as the switching delay increases.

While we have presented simulation analysis and the efficiency ratio of our algorithm, more work needs to be done to find algorithms that have a better efficiency ratio, possibly using other functions of switching delay.

References

[1] S. Roy, A. Das, R. Vijaykumar, H. Alazemi, E. Alotaibi, and H. Ma, “Capacity scaling with multiple radios and multiple channels in wireless mesh networks,” in *IEEE Wireless MESH Network Workshop*, September 2005.

[2] H. Skalli, S. Ghosh, S.K. Das, L. Lenzini, and M. Conti, “Channel assignment strategies for multiradio wireless mesh networks: Issues and solutions,” *Communications Magazine, IEEE*, vol. 45, no. 11, pp. 86–95, November 2007.

[3] Ashish Raniwala, Kartik Gopalan, and Tzi cker Chiueh, “Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 8, no. 2, pp. 50–65, 2004.

[4] P. Kyasanur and N. H. Vaidya, “Capacity of multi-channel wireless networks: Impact of number of channels and interfaces,” in *Proceedings of the 11th annual ACM/IEEE international conference on Mobile computing and networking*, August 2005.

[5] M. Marina and S. Das, “A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks,” in *Broadband Networks, 2005 2nd International Conference on*, October 2005, vol. 1, pp. 381–390.

[6] Ashish Raniwala and T.-C. Chiueh, “Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network,” in *IEEE INFOCOM*, 2005.

[7] Pradeep Kyasanur and Nitin H. Vaidya, “Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 10, no. 1, pp. 31–43, 2006.

[8] Krishna N. Ramachandran, Elizabeth M. Belding, Kevin C. Almeroth, and Milind M. Buddhikot, “Interference-aware channel assignment in multi-radio wireless mesh networks,” .

[9] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” *Automatic Control, IEEE Transactions on*, vol. 37, no. 12, pp. 1936–1948, Dec 1992.

[10] Xiaojun Lin and Ness B. Shroff, “The impact of imperfect scheduling on cross-layer congestion control in wireless networks,” *IEEE/ACM Trans. Netw.*, vol. 14, no. 2, pp. 302–315, 2006.

[11] Changhee Joo, Xiaojun Lin, and N.B. Shroff, “Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks,” in *IEEE INFOCOM*, April 2008, pp. 1103–1111.

[12] Changhee Joo, “A local greedy scheduling scheme with provable performance guarantee,” in *MobiHoc ’08: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, New York, NY, USA, 2008, pp. 111–120, ACM.

[13] Xiaojun Lin and Shahzada Rasool, “A distributed joint channel-assignment, scheduling and routing algorithm for multi-channel ad hoc wireless networks,” in *Proceedings of the IEEE INFOCOM*, 2007.

[14] Bong-Jun Ko, V. Misra, J. Padhye, and D. Rubenstein, “Distributed channel assignment in multi-radio 802.11 mesh networks,” in *Wireless Communications and Networking Conference, IEEE*, 2007, pp. 3978–3983.

[15] M. Nekoui, A. Ghiamatyoun, S.N. Esfahani, and M. Soltan, “Iterative cross layer schemes for throughput maximization in multi-channel wireless mesh networks,” *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*, pp. 1088–1092, Aug. 2007.

[16] Wai-Hong Tarn and Yu-Chee Tseng, “Joint multi-channel link layer and multi-path routing design for wireless mesh networks,” *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 2081–2089, May 2007.

[17] Paramvir Bahl, Ranveer Chandra, and John Dunagan, “SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks,” in *MobiCom ’04: Proceedings of the 10th annual international conference on Mobile computing and networking*, New York, NY, USA, 2004, pp. 216–230, ACM.

[18] Ramanuja Vedantham, Sandeep Kakumanu, Sriram Lakshmanan, and Raghupathy Sivakumar, “Component based channel assignment in single radio, multi-channel ad hoc networks,” in *MobiCom ’06: Proceedings of the 12th annual international conference on Mobile computing and networking*, New York, NY, USA, 2006, pp. 378–389, ACM.

[19] M Luby, “A simple parallel algorithm for the maximal independent set problem,” in *STOC ’85: Proceedings of the seventeenth annual ACM symposium on Theory of computing*, New York, NY, USA, 1985, pp. 1–10, ACM.