On Multi-Radio Multi-Channel Scheduling considering Switching Overhead in Wireless Mesh Networks

Mira Yun, Yu Zhou, Amrinder Arora, and Hyeong-Ah Choi

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 show that the well known Greedy Maximal Scheduling (GMS) dose not achieve any provable efficiency ratio when the switching overhead is considered. Moreover, we prove that the problem of finding a schedule that achieves the maximum throughput capacity for tree networks with switching overhead is NP-complete, even though the equivalent problem can be solved in polynomial time if switching overhead is not considered. We extend existing algorithms in the literature, GMS and Distributed Maximal Scheduling (DMS), taking the switching delay into account in the channel assignment. Simulation results show that our proposed algorithms significantly outperform prior known algorithms, when actual switching delays are injected into time slots, in packet throughput, capacity and average packet delay. Results also show that the improvements in performance become more pronounced as the switching delay increases.

Index Terms—Wireless mesh networks, channel assignment, scheduling algorithm, switching overhead

1 Introduction

In Wireless Mesh Networks (WMNs), channel assignment and scheduling methods are widely considered to be crucial in optimizing the network interference and throughput. In order to increase the network capacity, the wireless mesh routers can be equipped with multiple radios operating in multiple non-overlapping channels. By assigning different channels to radio interfaces in the interference range, multiple channel-interface pairs can be served simultaneously, and this leads to higher network capacity. In this multi-radio multi-channel environment, a proper assignment of channels to interfaces is a critical factor of resource allocation problem [1].

Channel assignment schemes can be divided into three categorizes: fixed, dynamic, and hybrid assignment [2]. Fixed schemes assign channels to radio interfaces statically [3],[4],[5] and are adequate for use when the network condition is stable. In order to consider significant changes to traffic load or network topology in fixed schemes, quasi-static channel assignments are proposed in [6] and [7]. Quasi-static schemes allow the channel assignment changes, but infrequently, resulting in negligible traffic measurement overheads and switching delays. Dynamic assignment schemes change the assignment as needed, per packet or time slot [8]. Since radio interfaces can frequently switch from one channel to another, dynamic schemes can accommodate the changes in network conditions. However channel switching delays (typically 802.11 card has a few hundreds of microseconds to a few milliseconds) can be a challenging problem [2],[9],[10]. 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 [4],[11],[12].

As observed in [2],[10],[13], switching overhead (for switching channels *and* switching radios) is a major factor that limits overall network throughput in dynamic schemes. Interestingly however, while some of the dynamic channel assignment algorithms do mention switching delay, none of them consider the overhead incurred from switching radios dynamically from one channel to another into account. The algorithms 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 delays are injected into time slots. In this paper, motivated by these observations, we study multi-radio multi-channel scheduling with switching overhead explicitly considered. Our research contributions presented in this paper are as follows:

- We model the delay overhead that is incurred during channel switching, and use that delay in the design of algorithms.
- We show that the well known Greedy Maximal Scheduling (GMS) dose not achieve any provable

- *Mira Yun, Yu Zhou, and Hyeong-Ah Choi are with the Department of Computer Science, The George Washington University, Washington, DC, 20052. E-mail: {mirayun,yuzhou, hchoi}@gwu.edu.*
- *Amrinder Arora is with NTELX Inc. and the Department of Computer Science, The George Washington University, Washington, DC, 20052 Email: aarora@ntelx.com, amrinder@gwu.edu*

efficiency ratio when the switching overhead is considered.

- We show that when the switching overhead is non-negligible, finding a schedule that achieves the maximum throughput capacity is NP-complete for tree networks even under 2-hop interference model. This problem is, by contrast, solvable in polynomial time in tree networks under the k-hop interference model if switching overhead is not considered [14].
- We extend existing algorithms in the literature, namely GMS and Distributed Maximal Scheduling (DMS) [15], taking the switching delay into account in the channel assignment.
- Through discrete-event simulations we show that our algorithms significantly outperform GMS and DMS, when actual switching delays are injected in time slots, in capacity, packet throughput and average packet delay metrics.

The rest of the paper is organized as follows. We review related work in Section 2 and discuss the role of switching overhead and explain why it is necessary to consider the effects in Section 3. In Section 4, we outline the system model and give a problem statement. We show the NP-completeness of the scheduling problem while considering the switching overhead for tree networks in Section 5. In Section 6, we present our extensions of GMS and DMS taking the switching overhead into account in the channel assignment and scheduling problem. Simulation results are shown in Section 7, and our conclusions are presented in Section 8.

2 Related Work

A vast amount of research has been conducted to exploit multiple channels for performance improvement. Ramachandran et al. [12] proposed a centralized channel assignment algorithm which has a Channel Assignment Server (CAS). CAS allocates channels to radio interfaces while minimizing interference in multi-radio multi-channel WMNs. Among the centralized algorithms, an optimum scheduling policy was given by Tassiulas et al. in their seminal paper [16]. 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 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 $1/\kappa$ in single-channel networks [17] and $1/(\kappa+2)$ in multiple-channel networks [15], where κ is the interference degree of the network [18],[19]. Very recently, insights into the true efficiency ratio of GMS have been presented in [14], where authors showed that the efficiency ratio of GMS is equal to a network property (pooling factor). They also showed 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 [20] proposed a simple distributed scheduling algorithm for single channel wireless networks, which achieves an efficiency ratio no smaller than GMS. In [15], Lin et al. suggested a distributed algorithm for multi-channel network with low-complexity that has the same level of efficiency ratio as GMS. Ko et al. [21] also proposed a distributed channel assignment algorithm with channel interference cost function which indicates the spectral overlapping level between channels. Generally, distributed algorithms can only achieve a fraction of the maximum possible throughput due to the lack of complete information. To provide the higher throughput in distributed schemes, Brzezinski et al. [22] proposed algorithms for pre-partitioning a mesh network into smaller subnetworks in which simple distributed scheduling algorithms can achieve the maximum capacity.

In addition, much of the recent research on multi-radio multi-channel WMNs has dealt the channel assignment and routing problem jointly as a challenging cross-layer problem [15],[23],[24]. Raniwala et al. [3] showed significant improvement of the overall network goodput in 802.11 based multi-channel WMN architecture by considering channel assignment and routing jointly. In [15], authors proposed a distributed channel scheduling algorithm that guarantees the efficiency ratio to be same as the centralized GMS algorithm in multi-channel wireless networks.

Recently switching overhead has been acknowledged as a factor for channel assignment and routing problems in multi-radio WMNs. However, none of existing schemes considers the switching overhead in channel assignment algorithm itself. In [10], Feng et al. suggested a hybrid channel assignment protocol (HCAP) to find out a reasonable tradeoff between flexibility and switching overheads. In order to avoid frequent interface switching, HCAP adopts static assignment for nodes that have the heaviest loads. In [13], authors considered switching overhead as a key factor to estimate interference level of a node/link.

3 Switching Overhead

In this section, we present the justification for considering switching overhead in the design of channel assignment algorithms.

3.1 Switching Delay - Negligible or not?

In multi-radio multi-channel environment, many channel assignment algorithms need frequent channel switching to optimize the efficiency of WMNs. However channel switching incurs some non-negligible delay, which leads to accumulation of switching delays between end to end nodes.

In a 802.11 card, the hardware switching delay is typically in the order of a few hundreds of microseconds to a few milliseconds [8],[9]. When a packet of 1024 bytes is transmitted through 802.11a/b network where the typical transmission rate is about 25Mbps/6Mbps, it takes $1024 \times 8/(25 \times 10^6) = 328\mu s$ or $1024 \times 8/(6 \times 10^6) = 1.3ms$, which are in the same range of 802.11a/b switching delay. Furthermore when switching occurs across different frequency bands (e.g., 5GHz for 802.11a and 2.4GHz for 802.11b/g) the impact of switching delay on the overall network performance becomes even more significant. In [4], Kyasanur and Vaidya showed that the switching delay degrades the network capacity as a function of $\frac{S}{S+T}$ (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.

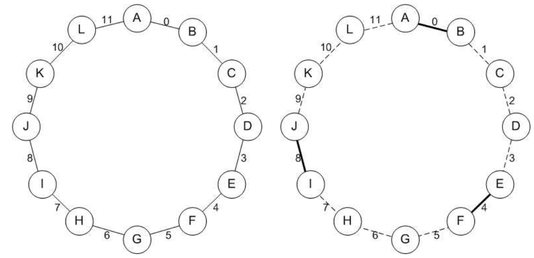
With technology advancements, it is expected that the switching delay will become smaller overtime [25]. However, the switching delay can be expressed in terms of packet duration as $d_t \times L/P$, where d_t is the hardware switching delay, and P and L are the packet size and transmission speed respectively. While the hardware switching delay can 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.

3.2 Shortcomings of GMS with Switching Overhead

It has been shown in [14] that the worst-case efficiency ratio of GMS in geometric network graphs is between $\frac{1}{6}$ and $\frac{1}{3}$. In our earlier work [26], however, we indicated that when switching overhead is considered, GMS algorithm may have no provable efficiency ratio. Next, we present a counterexample which shows that the efficiency ratio of GMS algorithm can be arbitrarily close to 0 when switching overhead δ is considered in a network with 2-hop interference model.

Consider the network topology as shown in Figure 1. We assume that the traffic moves in clockwise direction, all link capacities are 4 and the initial queue size is $\chi+3$ at nodes A, E and I, $\chi+2$ at nodes B, F and J, $\chi+1$ at nodes C, G and K, and χ, at nodes D, H and L, where χ is a large number. Consider that all nodes have a constant arrival rate of $1-\delta+\epsilon$, where ϵ is a small number. GMS initially serves links 0, 4 and 8, as the queues at the origin nodes of those links are the highest and all link capacities are equal. In the next timeslot, GMS serves nodes 1, 5 and 9. In the following timeslot, GMS serves nodes 2, 6 and 10. followed by nodes 3, 7 and 11. Then, the entire cycle repeats. During these 4 timeslots, each node receives service over one timeslot, and is able to send $4(1-\delta)$ bits. During the same 4 timeslots, it receives a total of $4(1-\delta+\epsilon)$ new bits from its arrival process, and thus its queue is not stable under GMS.

However, the following schedule can serve the same system with an arrival rate of $4/3-\delta$:



(a) Topology (b) Initial schedule

Fig. 1. Example topology and schedule with GMS

Consider three link assignments: $\{0,3,6,9\}$, $\{1,4,7,10\}$ or $\{2,5,8,11\}$. We observe that considering the 2-hop interference model, these link assignments are valid. Start with the first one, and switch to the next one after $\chi/3$ timeslots. Thus, over χ timeslots, each node can send $(\chi/3-1)4+(1-\delta)4$ bits. During the same χ timeslots, it receives a total of $\chi(4/3-\delta)$ new bits from its arrival process, and thus its queue is stable under this channel assignment scheme.

Therefore, the efficiency ratio of GMS over this schedule is no better than $\frac{1-\delta+\epsilon}{4/3-\delta}$. Since δ can be vary between 0 and 1, the efficiency ratio can be arbitrarily close to 0.

4 System Model and Problem Statement

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

4.1 System Model

We consider a time slotted multi-hop network modeled by an undirected graph $G=(V,E)$ where V denotes the 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 $\alpha(v)$ 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 the queue is assumed to be corresponding to $b(l)$, that is, the transmitter node of link l. The rate at which link l can transmit on channel c is denoted by $r(l,c)$.

4.2 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 $\gamma \leq 1$ 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.

5 NP-Completeness with Switching Overhead

In this section, we address the algorithmic aspects of the scheduling problem on the maximum throughput capacity while considering the switching overhead. Recently, Joo, Lin, and Shroff [14] have shown that the GMS achieves the full capacity region in single-channel tree networks under the k-hop interference model without considering the switching overhead. In contrast, we show that when the switching overhead is non-negligible, finding a schedule that achieves the maximum throughput capacity is NP-complete for single-channel tree networks even under 2-hop interference model.

5.1 Definitions and Preliminaries

In an undirected graph G, each link $l\in E$ is assigned a weight $w(l)$ denoting the amount of data to be transferred over the link l. We assume that the system operates synchronously in a time slotted mode. Each time slot is assumed to be one unit (of very small length), and the duration δ of switching overhead is a multiple of time slots. The value of $w(l)$ for each $l\in E$ is also assumed to be an integer. In order to transmit between two nodes u and v in time slot t using the same channel c, a switching delay δ is required before transmission if u and v were not engaged in transmission in time slot $t-1$ using channel c, which is called the *switching overhead constraint*. We assume 2-hop interference model. The problem is to find a schedule for transmitting all data $\sum_{l\in E} w(l)$ in the network that minimizes the overall completion time under the switching overhead constraint.

Note that a solution to this formulation of scheduling problem with provable upper bound on the schedule length can be used to understand the maximum throughput capacity. Let ς be a schedule for an arbitrary instance G with schedule length $T_\varsigma(G)\leq \tau\cdot T_{opt}(G)$ where $T_{opt}(G)$ denotes the optimal schedule length for G and τ is a constant. Then, by setting the rate at which data can be transferred over the link l to be $w(l)/\tau$ for each $l\in E$, the worst-case efficiency ratio of ς over the maximum throughput capacity can be at least $1/\tau$.

5.2 NP-Completeness for Tree Networks

Theorem 1. *Finding an optimal schedule under 2-hop interference model with switching overhead $\delta>0$ is NP-complete even if the network topology is a tree and only one channel is available in the network.*

Proof: To show the NP-completeness, we present a polynomial time reduction from a well-known NP-complete problem, namely the 3-Partition Problem.

3-Partition Problem (3-PART): Given a set of $3m$ positive integers $A=\{a_1,\cdots,a_{3m}\}$ such that $B/4<a_i<B/2$ for $1\leq i\leq 3m$, where $B=\frac{1}{m}\sum_{i=1}^{3m} a_i$, does there exist a partition of A into m disjoint sets $A_1,\cdots,A_m$ such that $\sum\{a_i \mid a_i\in A_j\}=B$ for $1\leq j\leq m$? (Note that by

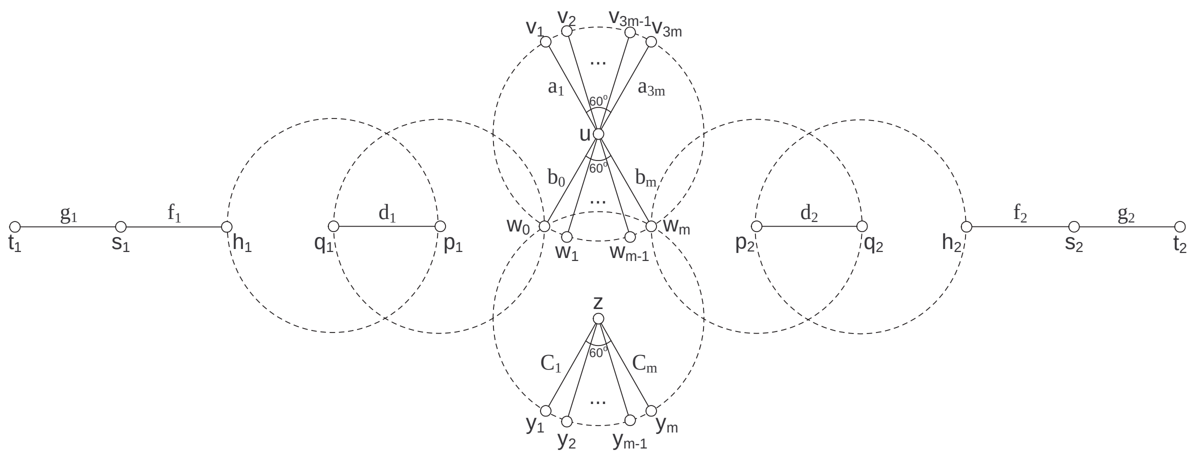


Fig. 2. Edge-weighted tree T

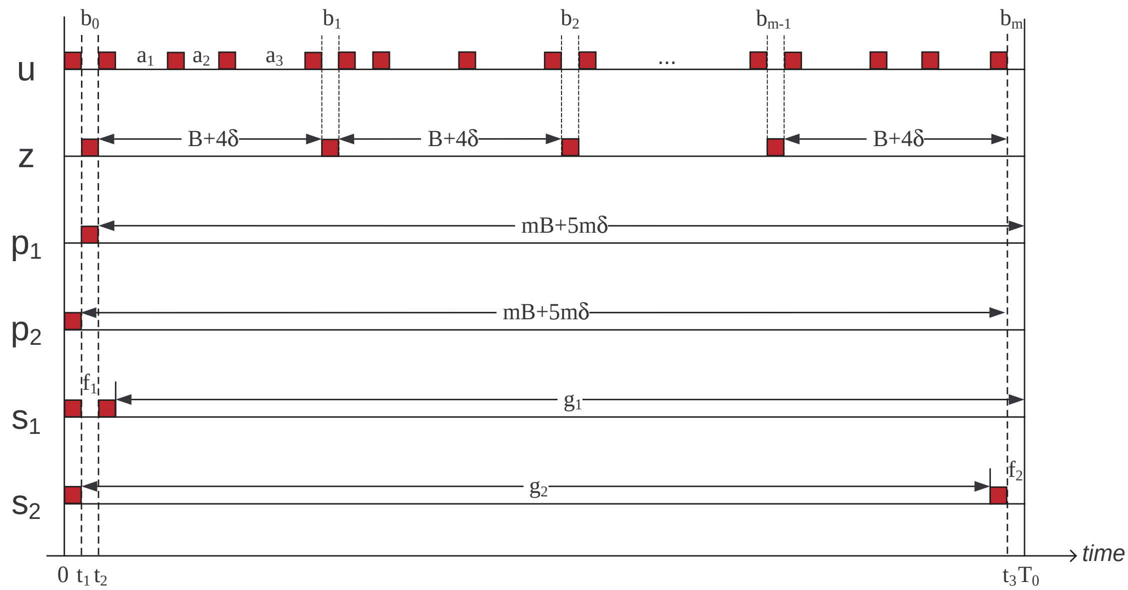


Fig. 3. Optimum Schedule

the bounds on a_i's, each A_j must contain exactly three members.)

We begin with defining an edge-weighted tree T from an arbitrary instance A as an input to the 3-PART. The node set $V(T)$ is defined as: $V(T) = \{v_1, \cdots, v_{3m}\} \cup \{u\} \cup \{w_0, \cdots, w_m\} \cup \{z\} \cup \{y_1, \cdots, y_m\} \cup \{p_1, q_1, h_1, s_1, t_1\} \cup \{p_2, q_2, h_2, s_2, t_2\}$. The link set $E(T)$ is defined as: $E(T) = \{(v_i, u) \mid 1 \leq i \leq 3m\} \cup \{(u, w_i) \mid 0 \leq i \leq m\} \cup \{(z, y_i) \mid 1 \leq i \leq m\} \cup \{(p_i, q_i), (h_i, s_i), (s_i, t_i) \mid 1 \leq i \leq 2\}$. The weight $w(l)$ for each link l is defined as $w(u, v_i) = a_i$ for $1 \leq i \leq 3m$, $w(u, w_i) = b_i = \delta$ for $0 \leq i \leq m$, $w(z, y_i) = C_i = B + 4\delta$ for $1 \leq i \leq m$, $d_1 = d_2 = mB + 5m\delta$, $f_1 = f_2 = \delta$, and $g_1 = g_2 = mB + (5m-1)\delta$. (Note that in the remaining discussion of the proof, we will also use weight $w(l)$ to refer to the corresponding link l.)

The distance between two nodes of any link is exactly ι where ι is the transmission range. Distances between w_0 and p_1 and between w_m and p_2 are also ι. Hence, transmission of link d_1 interferes with b_0 and f_1 but not with any other link. Similarly, transmission of link d_2 interferes with b_m and f_2 but not with any other link. It is also noted that link C_i, for any i, cannot be simultaneously engaged in transmission with link b_j for any j, but can be with any other link in T. Figure 2 shows such a tree T in which weight $w(l)$ associated with each link l denotes the amount of traffic to be transferred between the two end points of l.

From the construction of T, we observe that any optimum schedule has schedule length $T_{opt} \geq mB + (5m+2)\delta \stackrel{\text{def}}{=} T_0$. This due to the fact that node u requires transmission time at least $mB + (m+1)\delta$, and u has to communicate with $4m+1$ different receivers requiring at least $(4m+1)\delta$ switching delays. Here, we note that switching delay δ is also required for initial transmission at any node. In the following, we proceed to show that the 3-PART has a solution if and only if there exists a schedule with schedule length $T_{sch} = T_0$.

Suppose the 3-PART has a solution, and assume without loss of any generality that $(a_1 + a_2 + a_3) = (a_4 + a_5 + a_6) = \cdots = (a_{3m-2} + a_{3m-1} + a_{3m}) = B$. We then find an optimum schedule as shown in Figure 3.

In Figure 3, the schedule of links performed versus time are shown at nodes u, z, p_1, p_2, s_1, and s_2, where the intervals indicate the time required to transmit the corresponding data or time for switching delay. Using this schedule, all transmissions are clearly completed in T_0 time.

Conversely, assume that there exists a schedule ς with schedule length $T_\varsigma = T_0$. We then note that node u must be busy from time 0 to time T_0 by transmitting or switching (i.e., no idle period from time 0 till time T_0). In the following, we proceed to prove that at node u, either b_0 must be scheduled first (i.e., from time t_1 to t_2 as shown in Figure 3) and b_m is scheduled last (i.e., from time t_3 to T_0 as shown in Figure 3), or b_m first and b_0 last. Since tree T is symmetric, we will only prove the former case.

Suppose b_0 is not scheduled first at node u and scheduled from time t_4, where $t_2 < t_4 < t_3 - \delta$. Then, at node p_1, d_1 must be scheduled in two different intervals. Then, f_1 must be scheduled between time t_4 and $t_4 + \delta$ in order not to create an idle period at node u as otherwise transmissions of f_1 and d_1 will interfere each other. This will leave two non-consecutive intervals, say β_1 and β_2, that can be used for transmission of g_1 at node s_1. However, $\beta_1 + \beta_2 = mB + (5m-2)\delta$ but $g_1 = mB + (5m-1)\delta$. See Figure 4. Therefore, b_0 must be scheduled from time t_1 as shown in Figure 3. Now, assume that b_m is scheduled from time t_5, where $t_5 < t_3$. Then, at node p_2, d_2 must be scheduled in two different intervals, and at node s_2, f_2 must be scheduled between time t_5 and $t_5 + \delta$ in order not to create an idle period at s_2 as otherwise transmissions of f_2 and d_2 will interfere each other. Again, this will leave two non-consecutive intervals, say β_3 and β_4, for transmission of g_2 at node s_2. However, $\beta_3 + \beta_4 = mB + (5m-2)\delta$ but $g_2 = mB + (5m-1)\delta$. See Figure 5. Therefore, b_m must be scheduled last as in Figure 3.

We finally argue that at node u, scheduling of remaining b_i's for $1 \le i \le m-1$, must be done same as in Figure 3, which allows scheduling of C_j's for $1 \le j \le m$ again same as in Figure 3. This is due to the fact that at node z, a period of δ during the interval from time 0 to t_2 can be used only for switching (not for transmission), leaving the remaining δ as an idle period, and interval from t_3 to T_0 cannot be used at all for transmission. Therefore, each C_j must be scheduled non-preemptively in order not to create additional switching delay (i.e., only δ period can be used for switching delay for each C_j); hence, the schedule shown in Figure 3 is the only feasible one. It is then easy to see that exactly three a_i's must be scheduled to fit in three intervals of total length B, which completes a proof of the theorem. □

6 Scheduling considering Switching Overhead

In this section, we extend the existing algorithms taking the switching delay into account in the channel assignment. The basic idea is to define different weight functions depending on the necessity of switching. In the following subsections, we present our centralized and distributed control algorithms beginning with the existing algorithms.

6.1 Centralized Scheduling with Switching Overhead

Greedy Maximal Scheduling (GMS) has considered as the efficient and low-complexity scheduling algorithm for both single-channel and multi-channel wireless networks [14]. In this subsection, we will show the extension of GMS algorithm that considers switching overhead for multi-channel multi-radio wireless networks. In multi-channel multi-radio environment, GMS schedules link-channel pairs in decreasing order of the queue-weighted rate conforming to interference constraints. Let $\mathcal{F}$ denote the set of all link-channel pairs in a network graph G, i.e., $\mathcal{F} = \{(l,c) | l \in E, c \in C\}$. For all link-channel pairs (l,c), $w(l,c,t)$ is defined as the queue-weighted rate $q(l,t)r(l,c)$. 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, remove from $\mathcal{F}$ all ink-channel pairs (k,c) with $k \in I_l$. And if $\alpha(l) = 0$, which means link l already uses up all available radio interfaces, remove from $\mathcal{F}$ all link-channel pairs (k,c') with $k \in E(b(l)) \cup E(e(l))$. With the remaining pairs in $\mathcal{F}$, continue to find the largest weight until no link-channel pairs are left in $\mathcal{F}$. The detailed GMS algorithm is shown in Algorithm 1.

Our centralized algorithm considering switching overhead defines two different weight functions depending on whether or not the switching is needed. For a set of all scheduled link-channel pairs in time slot $t-1$, i.e. $\mathcal{Z} = \{(l,c) | z(l,c,t-1) = 1\}$, we define $w(l,c,t) = q(l,t)r(l,c)$. For a set $\{\mathcal{F} - \mathcal{Z}\}$, define $w(l,c,t) = (1-\delta)q(l,t)r(l,c)$. In other words, we consider the switching delay factor δ as an additional factor of the weight function if the channel switching is needed when channel $c \in C$ is assigned to link $l \in E$ in time slot t. With different weight functions we use above GMS algorithm to get $Z(t)$.

6.2 Distributed Scheduling with Switching Overhead

In [15], a distributed joint channel-assignment, scheduling, and routing algorithm (referred here as Distributed Maximal Scheduling (DMS) is proposed. They developed a distribute scheduling algorithm for multi-channel network that can guarantee the same efficiency ratio as the centralized GMS. The main idea is to use two

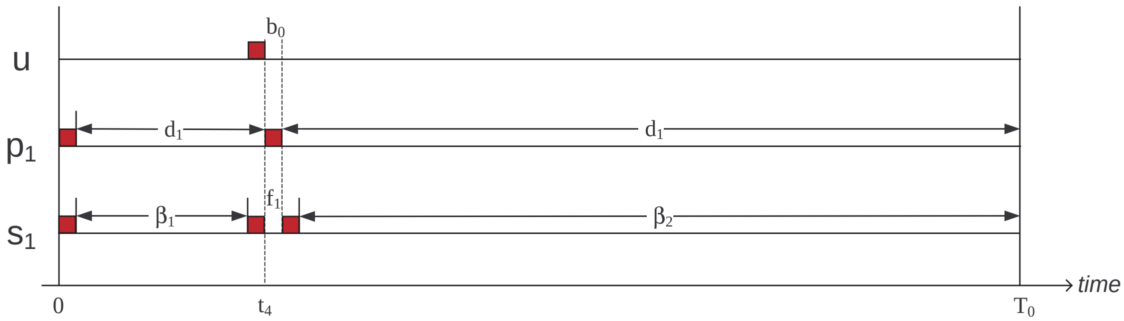


Fig. 4. Schedule for b_0

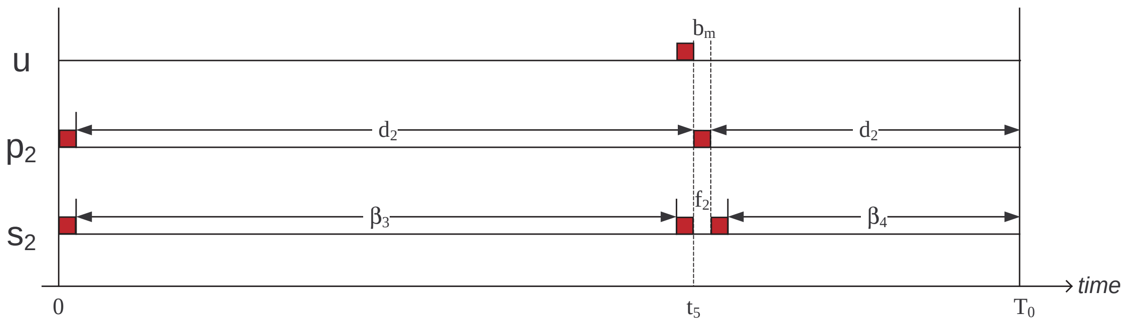


Fig. 5. Schedule for b_m

Algorithm 1 Greedy Maximal Scheduling (GMS)

$\beta(v) \leftarrow \alpha(v)$ for all nodes v
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

queueing steps to handle channel diversity. In the first step, packets arriving to each link l are assigned to each channel queue (logically) to prevent links from using "weak" channels. By using the queue length information DMS logically define the number of packets that link l can assign to channel c. In the second step, actual channels are assigned to radios according to multi-channel maximal scheduling algorithm.

In order to show the impact of switching overhead on the WMN throughput we extend their algorithm by considering switching overhead. We describe the single-path case (SP) only, but our switching overhead concept can be extended to the multi-path case.

Without the switching overhead, DMS 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 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 (We use LubyMIS algorithm [27]) 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.

Algorithm 2 Distribute Scheduling with Switching Overhead (DSSO) Algorithm

```
1: For 𝒵, x(l,c,t) = r(l,c) or 0
2: For {ℱ − 𝒵}, x(l,c,t) = (1 − δ)r(l,c) or 0
3: for each link l and channel c do
4:    Assign y(l,c,t) ∈ [0, x(l,c,t)]
      where Σ_{c=1}^{C} y(l,c,t) = min{q(l,t), Σ_{c=1}^{C} x(l,c,t)}
5: end for
6: for each channel c do
7:    find Z^c(t) by calling LubyMIS(G, c);
8: end for
```

Considering switching overhead, the proposed algorithm (DSSO) is as follows. For each time t, we have known the set $\mathcal{Z}$ of all scheduled link-channel pairs (l,c) at time $t-1$.

In the first step we define $x(l,c,t)$ for the set $\mathcal{Z}$ and the set $\{\mathcal{F}-\mathcal{Z}\}$ separately. For the set $\mathcal{Z}$ we follow the above DMS definition. For the set $\{\mathcal{F}-\mathcal{Z}\}$, $x(l,c,t)$ can be assigned as follows.

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

ξ_l is an arbitrary positive constant chosen for link l.

In the second step, based on the channel queues $(\eta(l,c,t) + y(l,c,t))$, Multi-channel Maximal Scheduling (LubyMIS) is carried out. And we give higher priority to the set $\mathcal{Z}$ backloged again.

In order to implement Multi-channel Maximal Scheduling algorithm, we use the Luby Maximal Independent Set (LubyMIS) algorithm for each channel c [27]. The algorithm consists of three rounds. In the first round, each link updates their weight $w(l,c,t)$ and send to 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 weight are marked as the winner. In the second round, each winner notify their interference neighbors the fact that they have won. Thus at the end of second round, the interference neighbors knows that they are the losers. In the third round, each loser notifies its neighbors. Then all the winners, the losers, and the loser's neighbors remove the appropriate link-channel pairs from the network. After the third round, the algorithm repeats from the first round to find the winners, the losers, and the loser's neighbors with remaining nodes and links. This process is repeated until no more link-channel pairs are left. Finally, LubyMIS provides $Z^c(t)$, consisting of the winners.

6.3 Stability Analysis

We prove in this section that the efficiency ratio of the proposed DSSO 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 DSSO.

As outlined in [15], one key to observing this is first note that there must exist some $\tilde{x}(l,c) \in [0, r(l,c)]$ such that:

$$\frac{(1+\epsilon)^2(\kappa+2)}{1-\delta}\sum_{s=1}^{S} H_s^l \lambda_s \leq \sum_{c=1}^{C}\tilde{x}(l,c), \forall \text{ links } l \tag{4}$$

$$\sum_{k\in I_l}\frac{\tilde{x}(k,c)}{r(k,c)} \leq \kappa \tag{5}$$

$$\sum_{k\in E(i)}\sum_{c=1}^{C}\frac{\tilde{x}(k,c)}{r(k,c)} \leq \alpha(i) \tag{6}$$

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 (4), interference constraint (5) and constraint on the number of radios (6). Using the same Lyapunov function and the techniques outlined in [15], we observe that the results follow as in [15].

We also observe that the proven efficiency ratio of the proposed DSSO algorithm is by definition less than the frequency ratio of the DMS 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 DSSO and DMS algorithms, it is clear that DSSO algorithm outperforms DMS significantly.

7 Simulation Results

In this section, we use simulation to evaluate the performance of the channel assignment algorithms. We first compare their system throughput and end-to-end delay with varying switching overhead δ. Then we show the average backlog under different packet arrival rate.

7.1 Simulation Scenarios

We consider two different network scenarios under 2-hop interference models. In the first scenario, we consider an 8×8 grid topology where each node could potentially communicate with up to four neighbors.

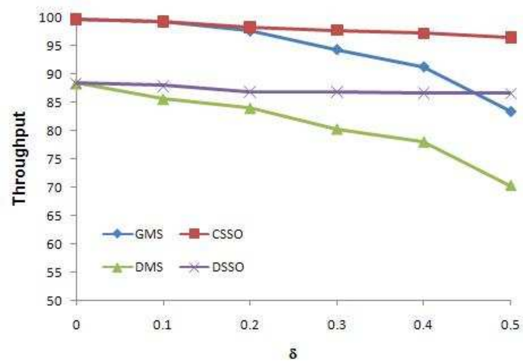


Fig. 6. The average throughput of scheduling algorithms

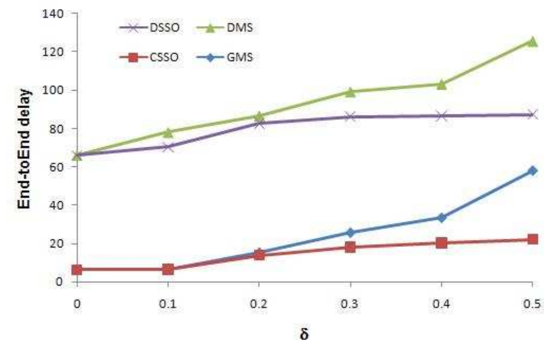


Fig. 7. The average end-to-end delay of scheduling algorithms

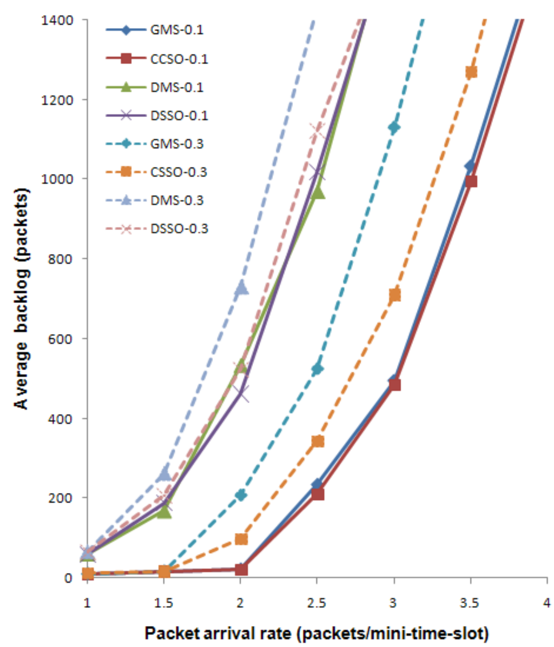


Fig. 8. The average backlog with $\delta = 0.1$and 0.3

Schedule occurs at every time-slot during simulation time (1000 time-slots). To consider the switching delay, each time-slot is divided into ten mini-time-slots (total of 10000 mini-time-slots). The number of radios on each node varies from 2 to 4, which includes one default radio to maintain the topology. We assume each radio has 7 non-overlapping channels, of which one of those channels is used as the default for the default radio. We randomly selected capacity for each channel for each link from [10, 14] (uniformly distributed), which means the number of unit packet per mini-time-slot. Then we randomly pick fifteen source-destination pairs for each of the packet arrival rate λ which follows the Poisson distribution.

For the second scenario, we consider 5 randomly generated mesh networks with 25 nodes in a square of 300×300 meters. Two nodes are connected by a link if they are within transmission range (100 meters). Each node has 4 radios and each radio has 7 channels which has a capacity between [10, 14] (uniformly distributed). Then we randomly pick ten source-destination pairs having 5 hops each. We assume a Poisson process with packet generation rate $\lambda = 3$ for packet arrivals. During simulation time (1000 time-slots), the routing table is fixed. Any routing algorithm can be used to create the routing table. Our work focuses on the channel assignment and scheduling aspect only.

7.2 Simulation Results

In the first scenario, we show the average backlog packets under different scheduling algorithm and switching overhead pairs. And we also show the system throughput and end-to-end delay with varying switching overhead with $\lambda = 2$. Fig. 6 compares the throughput for different algorithms varying switching overhead δ. We define the throughput as the ratio of the total received packets to the total sent packets. As shown in Fig. 6, the throughput of GMS and DMS which implemented without considering switching overhead has decreased dramatically when δ is larger than 0.2 and 0.1 respectively. However proposed algorithms (CSSO and DSSO) have almost the same performance with varying δ. Fig. 7 presents the end-to-end delay (time-slots) with varying δ switching overhead. As expected proposed algorithms show a vast improvement over GMS and DMS.

Fig. 8 plots the average backlog (queue length) versus the packet arrival rate under four different scheduling algorithms with $\delta = 0.1$ and 0.3. When the packet arrival rate (packets/mini-time-slot) approaches a certain limit, the average backlog increase dramatically. When $\delta = 0.1$, newly proposed algorithms have a slightly improved performace than others. However, when $\delta = 0.3$, the throughput performance of proposed algorithms is significantly improved. For example, GMS has more than double of the average backlog than our CSSO when the packet arrival rate is 2. Thus, by considering switching overhead, we can significantly improve network throughput and capacity.

In the second scenario, we considered 5 different random topologies. Table 1 shows average back-

TABLE 1
Average Backlog Improvement

Centralized Algorithm			
Network	$\delta = 0.2$	$\delta = 0.3$	$\delta = 0.4$
1	23.71	63.12	76.17
2	19.26	54.63	73.44
3	23.02	40.55	53.17
4	20.04	59.76	74.99
5	29.25	49.68	63.23
Distributed Algorithm			
Network	$\delta = 0.2$	$\delta = 0.3$	$\delta = 0.4$
1	8.68	30.88	46.78
2	4.19	26.31	41.87
3	7.84	24.68	41.24
4	11.16	28.16	48.64
5	7.29	20.15	33.98

TABLE 2
Throughput Improvement

Centralized Algorithm			
Network	$\delta = 0.2$	$\delta = 0.3$	$\delta = 0.4$
1	100.94	110.76	126.97
2	100.78	110.70	133.50
3	107.52	124.75	154.34
4	101.45	111.83	134.61
5	105.05	119.09	146.88
Distributed Algorithm			
Network	$\delta = 0.2$	$\delta = 0.3$	$\delta = 0.4$
1	102.70	115.40	132.98
2	101.88	114.82	135.56
3	106.74	130.72	175.19
4	104.30	117.92	145.79
5	104.31	119.51	146.32

log improvement of different δ values in each network. The average backlog improvement is defined as $\frac{ExistingAlgo\_backlog - ProposedAlgo\_backlog}{ExistingAlgo\_backlog} \times 100$, where $ExistingAlgo\_backlog$ is the average backlog of GMS and DMS and $ProposedAlgo\_backlog$ is the average backlog of the proposed algorithms (CSSO and DSSO). Overall, the proposed algorithms outperform in every case consistently with improvements from 7% to 70%.

Table 2 shows the improvement of throughput with varying δ in each network. The proposed algorithms can achieve up to 146% of throughput improvement. By presenting the results in 5 different random topologies, we showed that our proposed algorithms always outperform the existing algorithms and the improvements become more pronounced as the switching overhead increase.

8 Conclusion

In this paper, we considered the impact of the channel switching in multi-radio multi-channel WMNs. In contrast to currently available algorithms in many literatures, we explicitly modeled the delay overhead that is incurred during channel switching and used that delay in the design of algorithms. As a strong motivation, we presented that finding a schedule that achieves the maximum throughput capacity is NP-complete even for tree networks under 2-hop interference model, even though the equivalent problem is solvable in polynomial time if switching overhead is not considered. We extended two well known algorithms, centralized and distributed, taking the switching overhead into account in the channel assignment. Through discrete-event simulations we showed that the proposed algorithms significantly outperform the prior known algorithms,when actual switching delays are injected in time slots, 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 proposed algorithms, 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. 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] 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.*, 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*, vol. 1, October 2005, pp. 381–390.

[6] R. Juan and Q. Zhengding, "Centralized quasi-static channel assignment in multi-radio wireless mesh networks," in *Communication Systems, 2008. ICCS 2008. 11th IEEE Singapore International Conference on*, Nov. 2008, pp. 1149–1154.

[7] A. Subramanian, H. Gupta, S. Das, and J. Cao, "Minimum interference channel assignment in multiradio wireless mesh networks," *Mobile Computing, IEEE Transactions on*, vol. 7, no. 12, pp. 1459–1473, Dec. 2008.

[8] A. Raniwala and T.-C. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," in *IEEE INFOCOM*, 2005.

[9] R. Vedantham, S. Kakumanu, S. Lakshmanan, and R. 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: ACM, 2006, pp. 378–389.

[10] Y. Feng, M. Li, and M.-Y. Wu, "Improving capacity and flexibility of wireless mesh networks by interface switching," in *Communications, 2008. ICC '08. IEEE International Conference on*, May 2008, pp. 4729–4733.

[11] P. Kyasanur and N. 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.

[12] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, April 2006, pp. 1–12.

[13] Y. Feng, M. Li, and M.-Y. Wu, "A weighted interference estimation scheme for interface-switching wireless mesh networks," *Wirel. Commun. Mob. Comput.*, vol. 9, no. 6, pp. 773–784, 2009.

[14] C. Joo, X. Lin, and N. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multihop wireless networks," *Networking, IEEE/ACM Transactions on*, vol. 17, no. 4, pp. 1132 –1145, aug. 2009.

[15] X. Lin and S. Rasool, "A distributed joint channel-assignment, scheduling and routing algorithm for multi-channel ad hoc wireless networks," in *Proceedings of the IEEE INFOCOM*, 2007.

[16] 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.

[17] X. Lin and N. 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.

[18] P. Chaporkar, K. Kar, and S. Sarkar, "Throughput guarantees through maximal scheduling in wireless networks," in *In Proceedings of 43d Annual Allerton Conference on Communication, Control and Computing*, 2005, pp. 28–30.

[19] P. Chaporkar, "Achieving queue length stability through maximal scheduling in wireless networks," in *University of California*, 2006.

[20] C. 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: ACM, 2008, pp. 111–120.

[21] B.-J. 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. [Online]. Available: http://dx.doi.org/10.1109/WCNC.2007.727

[22] A. Brzezinski, G. Zussman, and E. Modiano, "Distributed throughput maximization in wireless mesh networks via prepartitioning," *Networking, IEEE/ACM Transactions on*, vol. 16, no. 6, pp. 1406–1419, Dec. 2008.

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

[24] W.-H. Tarn and Y.-C. 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.

[25] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 adhoc wireless networks," in *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2004, pp. 216–230.

[26] M. Yun, Y. Zhou, A. Arora, and H.-A. Choi, "Channel-assignment and scheduling in wireless mesh networks considering switching overhead," in *Communications, 2009. ICC '09. IEEE International Conference on*, June 2009.

[27] 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: ACM, 1985, pp. 1–10.

Mira Yun received the B.S. and M.S. degrees in Telecommunication Engineering from Pukyong National University, Busan, Korea, in 2002 and 2004, respectively. Currently, she is a Ph.D. candidate in Computer Science at the George Washington University, Washington, D.C. Her research interests include wireless networks, scheduling algorithms, and resource management.

Zhou Yu received the B.S degrees in Computer Science from Peking University, Beijing, China, in 1998. He is currently pursuing the Ph.D. degree in Computer Science at the George Washington University, Washington, D.C. His research interests include wireless networks and scheduling algorithms.

Amrinder Arora leads the product development division at NTELX Inc., and is also an instructor in the Department of Computer Science at the George Washington University. His research interests are in wireless networks and online algorithms. He received his M.S. and D.Sc. degrees in Computer Science from the George Washington University in 2001 and 2006 respectively.

Hyeong-Ah Choi received the M.S. degree from Seoul National University, Seoul, Korea, in 1982, and the Ph.D. degree from Northwestern University, Evanston, IL, in 1986, both in computer science. From 1986 to 1988, she was an Assistant Professor at Michigan State University, East Lansing, MI. Since 1988, she has been a Member of the Computer Science Department faculty at the George Washington University, Washington, D.C., where she is currently a Professor. Her research interests include design and analysis algorithms, wireless and optical networks, parallel processing, and graph theory.