A Spectrum Switching Delay-Aware Scheduling Algorithm for Centralized Cognitive Radio Networks

Didem Gözüpek, Seyed Buhari, and Fatih Alagöz

Abstract—We formulate a scheduling problem that takes into account different hardware delays experienced by the secondary users (SUs) in a centralized cognitive radio network (CRN) while switching to different frequency bands. We propose a polynomial-time suboptimal algorithm to address our formulated scheduling problem. We evaluate the impact of varying switching delay, number of frequencies, and number of SUs. Our simulation results indicate that our proposed algorithm is robust to changes in the hardware spectrum switching delay and its performance is very close to its upper bound. We also compare our proposed method with the corresponding constant switching delay-based algorithm and demonstrate that our suggestion of taking into account the different hardware delays while switching to different frequency bands is essential for scheduling in CRNs.

Index Terms—Resource allocation, scheduling, frequency switching delay, MAC, dynamic spectrum access, cognitive radio networks

1 Introduction

When a cognitive radio (CR) device changes its operation frequency, it experiences a hardware switching delay to tune to its new frequency before it can fully utilize it. This delay in general depends on the wideness between the two frequency bands; i.e., switching from central frequency of 800 MHz to 10 GHz conduces larger delay than switching from 800 MHz to 850 MHz due to the hardware capabilities of the frequency synthesizer. For instance, the delay incurred while switching from the central frequency belonging to one GSM operator to the frequency of another GSM operator might be small; however, switching from a GSM band to X-band takes longer time. When the range of frequencies that the cognitive radio network (CRN) operates in is narrow, this delay difference might be negligible. However, the CRNs of the future are envisioned to operate in a wide range of frequency bands [1], [2]. Therefore, spectrum allocation and scheduling algorithms designed for CRNs have to take into account different delays that occur while switching to different frequency bands. Note here that the dependence of the switching delay on the wideness between the old and new central frequencies is unique to the dynamic spectrum access paradigm since other wireless technologies typically operate in a narrower bandwidth.

- *D. Gözüpek is with the Department of Computer Engineering, Gebze Institute of Technology, Kocaeli, Turkey. E-mail: didem.gozupek@gyte.edu.tr.*
- *S. Buhari is with the Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia. E-mail: mesbukary@kau.edu.sa.*
- *F. Alagöz is with the Department of Computer Engineering, Bogazici University, Kuzey Kampus, Bilgisayar Muhendisligi Kat:4, P.K.2 TR-34342 Bebek, Istanbul, Turkey. E-mail: fatih.alagoz@boun.edu.tr.*

Manuscript received 20 Nov. 2011; revised 22 Mar. 2012; accepted 6 Apr. 2012; published online 19 Apr. 2012.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2011-11-0627.
Digital Object Identifier no. 10.1109/TMC.2012.101.

Some works in the literature use the term *channel switching latency* to refer to the delay encountered while searching for an idle channel [3], whereas some other works [4] use the term to refer to the hardware switching delay of the frequency synthesizer given that the CR device has already determined the idle channel to switch to. Our focus in this paper is on the latter definition of the term.

Channel switching delay in CRNs is mostly considered in the realm of routing [4], [5], [6], [7], [8], [9]. The primary goal in most of these works [6], [9] is to minimize the number of channel switchings along the route; hence, they do not differentiate between switching to different frequencies and assume that all of the channel switchings cause a certain delay irrespective of the frequency separation distance. Only a few of these works about routing [4], [5] consider the possibly different delays depending on the wideness between the frequency bands. Besides, the current works about scheduling and channel assignment in CRNs [10], [11], [12], [13], [14], [15] do not take the spectrum switching delay into account. To the best of our knowledge, ours is the first study on scheduling in CRNs that takes into account the hardware switching delay depending on the separation distance between the current and subsequent frequency bands.

The remainder of this paper is organized as follows: In Section 2, we discuss motivation and related work. In Section 3, we formulate our scheduling problem and in Section 4 we describe our proposed polynomial-time suboptimal algorithm. We present the simulation results in Section 5, and we conclude the paper in Section 6.

2 Motivation and Related Work

Switching the frequency of a wireless transceiver requires changing the input voltage of the voltage-controlled oscillator (VCO), which operates in a phase locked loop (PLL), to achieve the desired output frequency. Frequency switching speed is regarded as a critical performance parameter in many modern communication systems [16].

It is an important factor since it reduces the time available for data transmission. The frequency switching delay depends on the relative positions of the two channels on the radio spectrum [4], [5], [9], [16], [17] because as the difference between the reference and the final frequency is high, any small frequency drift in the reference oscillator is significantly magnified in the final synthesized frequency [16]. To alleviate this limitation, devices make this conversion in a step by step manner, which increases the time to switch to far away frequencies.

Hardware switching delay is a device dependent parameter. For instance, the TCI Model 7234 wideband SHF tuner has a tuning speed of 1 ms for each 500 MHz step [18], whereas the tuning speed of TCI 715 is 1 ms for each 10 MHz step [19]. As it was also pointed out by Cesana et al. [20], some device specifications such as [21] do not explicitly report the dependence of the switching delay on the separation distance between the frequencies. This is because the operational frequency ranges of these devices are narrower, and hence the difference depending on the frequency separation distance is negligible. For instance, TCI 735 has two operation modes, one between 20 and 3,000 MHz and the other between 3,000 and 8,000 MHz. In both cases, the typical switching delay is 1 ms and the maximum delay is 5 ms, which still implies that the switching delay is not constant. Both of these operational modes are narrower compared to 0.5-40 GHz range of the TCI 7234. The CRs of the future are envisioned to operate at a wide range of frequencies; therefore, frequency separation distance is inevitably an important factor for the spectrum switching delay of the future CRs. Even if the CR device operates in a narrower frequency band and the spectrum switching delay is assumed to be constant, the entire mathematical analysis in this paper remains valid except for a few modifications about how the switching delay is calculated. In essence, our proposed algorithm can be used with any switching delay model; the only difference is the part where the switching delay is calculated. We explain this fact in detail in Section 4.

Besides the requirement that CRs of the future have to operate in a wide range of frequencies, there are numerous other factors that render our spectrum switching delay-aware scheduler vital:

1. Smaller hardware spectrum switching delay means more expensive CR equipment. Our proposed scheduler obviates the need to have smaller hardware switching delay, and hence plays a role in decreasing the cost of the CR devices. Since cost will be an important factor when CRs appear in the market, having a more intelligent scheduling algorithm such as our proposed algorithm in this paper allows us to utilize less expensive devices. This way, CR equipment manufacturers can increase their profits. Since a cheaper CR device will enable more people to start using CR devices, our proposed algorithm will also have an impact on increasing the market penetration of CRs when they first appear in the market.
2. Smaller spectrum switching delay means heavier CR equipment because more filters are needed in the hardware to achieve smaller delay. For instance, the spectrum processor of TCI 745 [22] is 30 kg, whereas the one of TCI 7234 [18] is 6 kg. These weights are impractical for future CRs since they cannot be, for instance, a mobile handset. Achieving small spectrum switching delay with reasonable equipment weights remains a challenge. Since the performance of our proposed algorithm is robust to changes in the spectrum switching delay, it obviates the need to have small spectrum switching delay. Therefore, by using our proposed algorithm in the software, equipment manufacturers can use less number of filters in the hardware and still achieve high throughput performance. Therefore, our algorithm also helps decrease CR device weights by obviating the need to accomplish smaller spectrum switching delay and, hence, enabling the usage of less number of filters in the hardware.
3. Smaller spectrum switching delay means larger device dimensions because filters occupy space. For instance, the latest TCI 745 [22] has dimensions of $8U$ high, rack mount ($14'' \text{ H} \times 19'' \text{ W} \times 22'' \text{ D}$), which is far from being a handheld CR device. Hence, our proposed spectrum switching delay-aware scheduling algorithm (S^2DASA) in this paper helps achieve more reasonable device sizes for the future CRs by obviating the need to use large number of filters for less switching delay.
4. In wireless networks, scheduling decisions are typically made for a duration where the network conditions are fairly stable. In CRNs, this duration also depends on primary users' (PU) activities since secondary users (SUs) are obliged not to disturb PUs. A swiftly changing PU activity and spectral environment makes the scheduling periods shorter since the CR devices have to adapt their transmission parameters such as rate and power more quickly in order not to disturb PUs. In the time-slotted scheduling framework in this paper, we take a time slot length as 100 ms, which is appropriate for slowly varying spectral environments like the TV bands. For instance, TCI 7234 [18] has a switching delay of 1 ms per 500 MHz. Switching the operation frequency from 0.5 GHz to 40 GHz requires 79 ms delay. Considering that the time slot length is 100 ms, 79 ms of switching delay implies that only 21 percent of the time slot can actually be utilized for data transmission. In other words, the impact of switching delay can be significant even in a slowly varying spectral environment where a 100 ms of time slot length is sufficient. As the spectral environment varies more frequently, time slot lengths have to decrease and hence, the impact of the spectrum switching delay intensifies.
5. Different CR users may opt for CR devices with different spectrum switching delay. The entity that is responsible for executing the scheduling algorithm usually cannot mandate the CR users to use a specific device. For instance, in a centralized CRN where the cognitive base station (CBS) is responsible for the management of the SUs in its service area, the scheduler resides at the CBS. If some CR users have less expensive devices with larger spectrum switching delay and if the CBS operator does not consider switching delay in its scheduling algorithm, it may unwittingly end up with providing coarse QoS to these CR users, which are the customers of the CBS

operator. Some customers may even starve in terms of throughput without the CBS operator having intended to do so.

6. Our simulation results indicate that high throughput savings are achieved even when the number of CR users is as small as 10. Furthermore, the throughput savings of our proposed algorithm increase as the number of frequencies increases. The increased impact of switching delay on the throughput performance as the number of frequencies increases indicates that our proposed algorithm will become increasingly useful as the CR paradigm proliferates in real applications.
7. The range of frequencies that a centralized CBS cell operates in is not static. A central entity called "spectrum broker" or "spectrum policy server" coordinates the spectrum usage of several base stations by periodically (usually in a medium term) allocating frequency pools to the base stations [1], [2], [23], [24], [25], [26]. To this end, auction theoretic or game theoretic techniques are usually employed [23], [25], [26]. Therefore, it is not feasible for the hardware providers of the mobile terminals to design the CR hardware customized for the frequency range of each and every CBS cell. Furthermore, CR devices may need to handover from one CBS cell to another. Having a CR device specifically designed for operating in a narrow range of frequencies hinders the handover mechanisms between different CBS cells. Moreover, the ultimate goal of the CR technology is to provide an interoperable "universal wireless device" that can seamlessly handle a wide range of frequencies [27]. To this end, CR devices should be able to operate in a wide range of frequencies.

3 Problem Formulation

In our previous work [13], we formulated a scheduling problem that makes frequency, time slot, and data rate allocation to the SUs in a CRN cell by maximizing the total average throughput of all SUs in the CRN cell, while at the same time ensuring that reliable communication of the SUs with the centralized CBS is maintained, no collisions occur among the SUs, and the PUs are not disturbed. The centralized time-slotted CRN cell that the scheduling framework in [13] focuses on is illustrated in Fig. 1. The work in [13] considers the case where SUs possibly have multiple interfaces for data transmission. The first step of the solution in [13] consists of determining the values for U_{if}, which stands for the maximum number of packets that can be sent by SU i using frequency f in any time slot during the entire scheduling period consisting of T time slots. These values are determined such that the transmission power of the SUs do not violate the maximum tolerable interference power limits of the PUs, and reliable communication of the SUs with the CBS is maintained. The following formula derived in [13] accomplishes these goals: $U_{if} = \lfloor \ln(1 + P_{IF_{max}} \times (\frac{|G_{i0}|}{|G_{if}| \times \sigma})^2) \rfloor$, where $P_{IF_{max}}$ stands for the maximum tolerable interference power of the PUs, which is without loss of generality (w.l.o.g.) assumed to be the same for all PUs and frequencies,

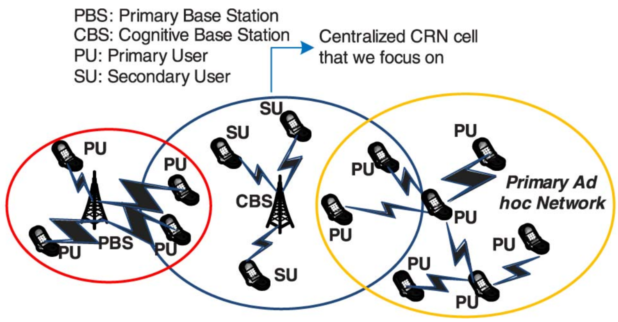


Fig. 1. The considered centralized CRN architecture [13].

and $|G_{i0}|$ is the channel gain between SU i and the CBS. Besides, $|G_{if}|$ is the maximum channel gain among all the channel gains between SU i and PUs that are actively using frequency f in the scheduling period, and σ^2 is the noise variance. Moreover, it is assumed for simplicity and yet w.l.o.g. that $S = B \times T_s$, where S is the packet size, B is the bandwidth, and T_s is the time slot length. The derivation of the formula for U_{if} takes into account spectrum availabilities and interference to PU. Interested readers can refer to [13] for details. For simplicity of notation, let us use in the remainder of this paper the symbol V_{if} in lieu of $\ln(1 + P_{IF_{max}} \times (\frac{|G_{i0}|}{|G_{if}| \times \sigma})^2)$; in other words, $U_{if} = \lfloor V_{if} \rfloor$. That is to say, in the first stage, U_{if} values are determined according to the activities of PUs and other channel conditions.

Assume that the network conditions, i.e., the PU and SU locations, the PU spectrum occupancies, and all the channel fading coefficients, are small enough not to have any impact on the U_{if} values for a duration of T time slots in the considered centralized CRN cell. Note that because of the floor operator in $U_{if} = \lfloor V_{if} \rfloor$, the scheduling period length T does not mandate the PU and SU locations as well as the PU spectrum occupancies to remain constant in that time period, but only requires that the change in their values does not alter U_{if}. The value of T, in general, depends on the characteristics of the spectrum environment. For instance, a slowly varying spectrum environment like the TV broadcast bands utilized by an IEEE 802.22 network allows T to have a fairly large value. In particular, IEEE 802.22 mandates that SUs need to vacate the spectrum bands within 2 seconds of the appearance of the PUs [28]. As in [13], in the simulations part of this work, we have taken $T = 1$ s (consisting of 10 time slots where each time slot is $T_s = 100$ ms), which is sufficient for proper operation of IEEE 802.22 networks.

Given the U_{if} values determined in the first stage, the work in [13] solves in the second stage the following binary integer linear program (ILP) to maximize the total throughput:

$$\max\left(\sum_{i=1}^{N}\sum_{f=1}^{F}\sum_{t=1}^{T}\frac{U_{if}X_{ift}}{T}\right), \tag{1}$$

$$s.t.$$

$$\sum_{f=1}^{F}\sum_{t=1}^{T} X_{ift} \geq 1;\ \forall i \in \mathcal{N}, \tag{2}$$

$$\sum_{i=1}^{N} X_{ift} \leq 1;\ \forall f \in \mathcal{F}, \forall t \in \mathcal{T}, \tag{3}$$

$$\sum_{f=1}^{F} X_{ift} \leq a_i;\ \forall i \in \mathcal{N}, \forall t \in \mathcal{T}, \tag{4}$$

where $\mathcal{N}$ denotes the set of a total number of N SUs in the CRN cell, $\mathcal{F}$ denotes the set of a total number of F frequencies in the CRN cell, and $\mathcal{T}$ denotes the set of a total number of T time slots in a scheduling period; i.e., $\mathcal{T} = \{1, 2, \ldots, T\}$. In addition, X_{ift} is a binary decision variable such that $X_{ift} = 1$ if SU i transmits with frequency f in time slot t and 0 otherwise, and a_i is the number of interfaces of SU i. In this formulation, (2) guarantees that at least one time slot is assigned to every SU, and hence providing a temporal notion of fairness. Without this temporal fairness constraint in the problem formulation, some SUs with bad channel conditions may end up with being unable to send any packets for a long time. Some transport layer protocols such as TCP close the connection if no packets are received for a certain amount of time. Constraint (2) gives each SU the opportunity to send at least something and therefore avoids this undesired disconnection situation caused by transport layer protocols. Besides, (3) ensures that at most one SU can transmit in a particular time slot and frequency, and hence obviating collisions between the SUs. Moreover, (4) represents the fact that an SU i cannot transmit at the same time using frequencies more than the number of its transceivers (interfaces), a_i, because each transceiver can tune to at most one frequency at a time. Besides, the work in [13] assumes w.l.o.g. that the V_{if} and consequently the U_{if} values are the same for all the interfaces of a particular SU i for a particular frequency f. The goal of this assumption is to evaluate the impact of the number of interfaces more effectively. In this paper, we follow the same assumption.

The formulation in (1)-(4) of the work in [13] assumes that no delay occurs when an SU switches from a frequency to another frequency. However, in reality, some portion of the subsequent time slot is inevitably wasted to tune to the new frequency; therefore, only the remaining portion of the next time slot can be used for actual data transmission. It may even be the case that the time it takes to switch to the new frequency is greater than or equal to the time slot length, which means that no packets can actually be sent using the new frequency. Since the scheduling decisions are known in advance by SUs, they should not waste time and energy in vain to switch to the new frequency; they should instead stay in the same frequency. On the other hand, the new frequency band might be more advantageous in terms of throughput by having a higher U_{if} value. The question is whether the delay incurred while switching to the new frequency band offsets this throughput advantage or not. If the throughput advantage of the new frequency band outweighs the disadvantage of throughput losses due to switching delay, then the SU may still prefer switching to the new frequency. Therefore, there is a tradeoff here; i.e., switching to the new frequency band may or may not be advantageous depending on the circumstances (switching delay and the channel conditions (U_{if} values) of the old and new frequencies). Furthermore, we also need to keep track of the information about which interface is assigned to which frequency since each interface experiences different switching delays depending on the frequency that it was assigned to in the previous time slot. In this paper, we extend the work in [13] to account for the spectrum switching delay, which depends on the distance between the used frequencies. Moreover, as in [13], we assume in the simulations part of this work that the buffers of the SUs are continuously backlogged; i.e., there are always enough number of packets to transmit with the data rate determined by the scheduling algorithm. This assumption enables us to effectively evaluate the performance of the scheduling algorithms by avoiding the possible influence of the traffic arrival process. Furthermore, as in [13] and most other works in the literature, we assume that physical layer parameters such as channel gains are known a priori. In practice, channel gains can be estimated by the SUs for instance by employing sensors near all receiving points and can be made available at the central controller, which is the CBS.

Let us denote by C_{iat} the frequency that interface a of SU i is assigned to in time slot t. Then, $C_{iat} = \sum_{f=1}^{F} f\ X_{iaft}$, where X_{iaft} is a binary variable that equals 1 if frequency f is assigned to the interface a of SU i in time slot t, and 0 otherwise. Let us denote by Δ_{iaft} the absolute difference in terms of the number of frequencies that interface a of SU i has to step to use frequency f in time slot t. Note that the interfaces do not have to be assigned some frequency in every time slot; in other words, it is possible for an interface not to be assigned any frequency in some time slot. If interface a of SU i has not been assigned some frequency for time slot t, then we say that t is a *silent time slot* for the interface a of SU i. Otherwise, we say that t is a *busy time slot* for the interface a of SU i. A time slot t may be a silent time slot for some interface but a busy time slot for another interface. Let us denote by m_{iat} the index of the busy time slot before time slot t. If t is the first busy time slot in the scheduling period, then $m_{iat} = 0$. In other words,

$$m_{iat} = \max_{\substack{\exists f s.t. X_{iafj}=1 \\ j<t}} \{j, 0\}.$$

Then the number of silent time slots between the current time slot t and the previous busy time slot for interface a of SU i equals $t - m_{iat} - 1$. If $m_{iat} = 0$, i.e., if t is the first busy time slot for interface a of SU i in the scheduling period, then in this case, as in [29], we assume that $\Delta_{iaft} = 0$. In other words, we assume that the interfaces are pretuned to the first used frequency. In practice, this delay in the first used time slot may depend on various other factors such as MAC protocol. If a single interface is used for both data transmission and control traffic, the interface may have to tune to the frequency band of the common control channel (CCC) during the time between consecutive scheduling periods. How frequently the tuning to the CCC is performed and which frequency the CCC uses depends on the protocol implementation. To isolate us from the possible influence of these factors, as in [29], we assume that the interfaces are pretuned to the first used frequency.

On the other hand, if a frequency f is not the first used frequency for interface a of SU i and there are silent time slots preceding time slot t, i.e., $0 < m_{iat} < t - 1$, then interface a uses these silent time slots to switch to the new frequency f. Scheduling decisions are made by the CBS for the duration of a scheduling period, which consists of T time slots. Scheduling decisions for all T number of time slots are made by the scheduling algorithm but it is not the case that the scheduling algorithm is executed in each time slot. The decisions for all time slots of that particular scheduling period are made once and this is before the actual scheduling period for data transmission of that scheduling period starts. In other words, ours is a "frame-based" scheduling discipline rather than a "slot-based" scheduling discipline. These scheduling decisions (X_{iaft} values) are then sent by the CBS to the SUs through the CCC. Therefore, SUs know the scheduling decisions (which frequencies are assigned to them in which time slots) before the beginning of the first time slot of the scheduling period. Because the scheduling decisions are known by SUs in advance, they can use these silent time slots to switch to the new frequency. If the number of silent time slots are enough to achieve the entire frequency switching, SU becomes ready to use the new frequency in the upcoming busy time slot. In this case, SU does not waste any portion of the busy time slot for frequency switching and hence it can use the entire busy time slot for data transmission. Otherwise, SU utilizes the silent time slots to achieve some portion of the frequency switching. The remaining switching is completed at the beginning of the next busy time slot. If the silent time slots and portions of the busy time slot are still not enough to achieve the frequency switching and no available time remains in the busy time slot for data transmission, then it means that no packets can be sent by the SU using the new frequency in the busy time slot.

Let us denote by β the switching delay in terms of milliseconds for each unit step in the frequency range. The value for β is hardware dependent; for instance, the delay is taken as $1\ \mathrm{ms}/10\ \mathrm{MHz}$ in [19]. In this case, if the frequencies are separated with a 10 MHz difference (for instance $f=1$ corresponds to 800 MHz, $f=2$ corresponds to 810 MHz etc.), then $\beta = 1\ \mathrm{ms}$. If $m_{iat} \neq 0$, then $\left|f - C_{iam_{iat}}\right|$ is the number of frequencies that SU i needs to sweep to tune to the new frequency f in time slot t. Since there are $(t - m_{iat} - 1)$ number of silent time slots, each one of which has a length of T_s milliseconds, then $\frac{(t-m_{iat}-1)T_s}{\beta}$ portion of the frequency band is switched to during the silent time slots. The remaining portion that needs to be switched during the busy time slot t is equal to $(|f - C_{iam_{iat}}| - \frac{(t-m_{iat}-1)T_s}{\beta})^+$, where $(x)^+ = \max(0, x)$. Recall that Δ_{iaft} denotes the absolute difference in terms of the number of frequencies that interface a of SU i has to step to use frequency f in time slot t. Then, $\Delta_{iaft} = 0$ if $m_{iat} = 0$. Otherwise, $\Delta_{iaft} = (|f - C_{iam_{iat}}| - \frac{(t-m_{iat}-1)T_s}{\beta})^+$. Similar to the works in [4] and [5], we use in the calculation of Δ_{iaft} a linear relationship between the switching delay and the wideness of the difference in the assigned frequencies. In essence, any switching delay model is applicable to our proposed algorithm with a slight modification. For instance, if it is assumed that the switching delay is constant, then $\Delta_{iaft} = 0$ if $m_{iat} = 0$ or $f = C_{iam_{iat}}$, and $\Delta_{iaft} = (1 - \frac{(t-m_{iat}-1)T_s}{\beta})^+$ otherwise. The rest of the analysis remains the same. We elaborate on this in detail in Section 4. In the remainder of this work, we consider the case where the spectrum switching delay is not constant and depends on the frequency separation distance. Finally, if we denote by B_{iaft} the maximum number of packets that can be sent by interface a of SU i if it is tuned to frequency f in time slot t, then $B_{iaft} = \lfloor (1 - \frac{\beta \times \Delta_{iaft}}{T_s})^+ V_{if} \rfloor$.

We then extend the throughput maximizing scheduler formulation in [13] to account for the spectrum switching delay and formulate the following binary integer program:

$$\max\left(\sum_{i=1}^{N}\sum_{a=1}^{a_i}\sum_{f=1}^{F}\sum_{t=1}^{T}\frac{B_{iaft}X_{iaft}}{T}\right), \tag{5}$$

s.t.

$$\sum_{a=1}^{a_i}\sum_{f=1}^{F}\sum_{t=1}^{T} X_{iaft} \geq 1;\ \forall i \in \mathcal{N}, \tag{6}$$

$$\sum_{i=1}^{N}\sum_{a=1}^{a_i} X_{iaft} \leq 1; \forall f \in \mathcal{F}, \forall t \in \mathcal{T}, \tag{7}$$

$$\sum_{f=1}^{F} X_{iaft} \leq 1; \forall a \in \mathcal{A}_i, \forall i \in \mathcal{N}, \forall t \in \mathcal{T}, \tag{8}$$

$$C_{iat} = \sum_{f=1}^{F} f\ X_{iaft}; \forall i \in \mathcal{N}, \forall a \in \mathcal{A}_i, \forall t \in \mathcal{T}, \tag{9}$$

$$m_{iat} = \max_{\substack{\exists f s.t. X_{iafj}=1 \\ j<t}} \{j, 0\}, \tag{10}$$

$$\Delta_{iaft} = \begin{cases} 0, & \text{if } m_{iat} = 0 \\ \left(|f - C_{iam_{iat}}| - \frac{(t-m_{iat}-1)T_s}{\beta}\right)^+, & \text{o.w.,} \end{cases} \tag{11}$$

$$B_{iaft} = \left\lfloor \left(1 - \frac{\beta \times \Delta_{iaft}}{T_s}\right)^+ V_{if} \right\rfloor, \tag{12}$$

where $\mathcal{N}$, $\mathcal{A}_i$, a_i, $\mathcal{F}$, and $\mathcal{T}$ are as defined previously. The objective function in (5) maximizes the total average throughput of all the SUs in the CRN cell, whereas constraint (6) guarantees that each SU is assigned at least one time slot. Moreover, (7) represents the constraint that at most one interface can use a frequency in a specific time slot, and (8) denotes that each interface can tune to at most one frequency in a time slot. In other words, purposes of the constraints (6), (7), and (8) are the same as purposes of the constraints (2), (3), and (4), respectively. The focus of this paper is on spectrum switching delay rather than providing temporal fairness. The reason for having the temporal fairness constraint in (6) is to ensure comparability of the results with the problem formulation outlined in (1)-(4) from our previous work in [13]. In other words, the reason for having the same constraints in (6), (7), and (8) is because our goal is to evaluate the impact of the spectrum switching delay awareness more effectively by comparing our spectrum switching delay-aware scheduler with the scheduler in [13, (1)-(4)]. To be able to make an effective comparison, all other features of both schedulers except switching delay awareness have to be the same. Therefore, we have constraints (6), (7), and (8). Furthermore, the constraints (9), (10), (11), and (12) are as explained previously. Note here that because B_{iaft} values depend on the frequency assignments in the previous time slots, the objective function in (5) is nonlinear. Nonlinear binary integer programming is in general known to be computationally hard in the literature [30].

4 Proposed Algorithm

In this section, we propose a polynomial time heuristic algorithm to address the problem in (5)-(12). In the remainder of this paper, we refer to our proposed algorithm by $\mathrm{S^2DASA}$.

We outline the main steps of $\mathrm{S^2DASA}$ in Algorithm 1. The set $\Phi \subseteq \mathcal{N}$ symbolizes the set of SUs which have not yet been assigned any time slot during the execution of the algorithm. B'_{iaf} represents the benefit (in terms of the maximum number of packets that can be transmitted) that interface a of SU i receives for using frequency f in that particular time slot. In Step 1, $\mathrm{S^2DASA}$ initializes the set Φ to $\mathcal{N}$ since none of the SUs have been assigned a time slot at the beginning of the algorithm. Moreover, Step 1 initializes the benefit values B'_{iaf} for the first time slot to $\lfloor V_{if} \rfloor$ since no switching delay occurs in the first time slot. $\mathrm{S^2DASA}$ makes the frequency assignment sequentially for each time slot. At the beginning of each time slot, the algorithm chooses the set Ψ, which is the set of SUs that have to be assigned at least one frequency in that particular time slot. $B^{'}_{iaf}$ indicates the maximum number of packets that can be sent by interface a of SU i if it is tuned to frequency f in that particular time slot. To determine the set Ψ, we introduce a metric called Γ_i to select the SUs with relatively good $B^{'}_{iaf}$ values averaged over all of their interfaces. Step 3 of $\mathrm{S^2DASA}$ assigns the Γ_i value by determining the average benefit value per interface for each SU in the set Φ. In each time slot, the set $\Psi \subseteq \Phi$ with relatively high Γ_i values is selected, and every SU in this set Ψ is guaranteed to be assigned with a frequency in that time slot. Steps 4 and 5 of $\mathrm{S^2DASA}$ assign all the SUs in the set Φ to the set Ψ if the number of SUs in the set Φ is less than or equal to $\lceil \frac{N}{T} \rceil$. Otherwise, in Step 7, $\lceil \frac{N}{T} \rceil$ number of SUs that have the largest Γ_i values are selected and added to the set Ψ. In Step 9, $\mathrm{S^2DASA}$ runs the following ILP:

$$\max\left(\sum_{i=1}^{N}\sum_{a=1}^{a_i}\sum_{f=1}^{F} B'_{iaf} X'_{iaf}\right), \tag{13}$$

$$s.t. \quad \sum_{a=1}^{a_i}\sum_{f=1}^{F} X'_{iaf} \geq 1; \forall i \in \Psi, \tag{14}$$

$$\sum_{i=1}^{N}\sum_{a=1}^{a_i} X'_{iaf} \leq 1; \forall f \in \mathcal{F}, \tag{15}$$

$$\sum_{f=1}^{F} X'_{iaf} \leq 1; \forall a \in \mathcal{A}_i, \forall i \in \mathcal{N}, \tag{16}$$

where X'_{iaf} is a binary decision variable that equals 1 if interface a of SU i transmits using frequency f, and 0 otherwise. At the end of the decision for every time slot, B'_{iaf} values are calculated and updated (in Steps 15-17) for the subsequent time slot. Notice here that B'_{iaf} values are input variables rather than decision variables in the optimization problem in (13)-(16). Constraint (14) ensures that the SUs in the set $\Psi \subseteq \Phi$ are assigned at least one frequency, which meets the constraint in (6) for the SUs in Ψ. As in (7) and (8), constraints (15) and (16) ensure that at most one interface can use a frequency, and each interface can tune to at most one frequency (in that time slot).

Algorithm 1. $\mathrm{S^2DASA}$ (Algorithm for the problem in (1)-(4))
Require: $\mathcal{N}$, $\mathcal{F}$, $\mathcal{T}$, $\mathcal{A}_i$, V_{if}.
Ensure: X_{iaft} values $\forall i \in \mathcal{N}, \forall a \in \mathcal{A}_i, \forall f \in \mathcal{F}, \forall t \in \mathcal{T}$.
1: $\Phi \leftarrow \mathcal{N}$, $B^{'}_{iaf} \leftarrow \lfloor V_{if} \rfloor, \forall i \in \mathcal{N}, \forall a \in \mathcal{A}_i, \forall f \in \mathcal{F}$
2: **for** $t = 1$ to T **do**
3: $\quad \Gamma_i \leftarrow \sum_{a=1}^{a_i}\sum_{f=1}^{F} B^{'}_{iaf}/a_i, \forall i \in \Phi$
4: $\quad$ **if** $|\Phi| \leq \lceil \frac{N}{T} \rceil$ **then**
5: $\quad\quad \Psi \leftarrow \Phi$
6: $\quad$ **else**
7: $\quad\quad$ Sort all $i \in \Phi$ wrt. Γ_i. Add $\lceil \frac{N}{T} \rceil$ number of i's that have the largest Γ_i values to Ψ; i.e., $|\Psi| = \lceil \frac{N}{T} \rceil$
8: $\quad$ **end if**
9: $\quad$ Run ILP in equations (13)-(16) with $\Psi, B^{'}_{iaf}, \mathcal{N}, \mathcal{F}, \mathcal{A}_i$, and obtain $X'_{iaf}, \forall i \in \mathcal{N}, \forall a \in \mathcal{A}_i, \forall f \in \mathcal{F}$
10: $\quad$ **for all** $i \in \Phi$ **do**
11: $\quad\quad$ **if** $\sum_{a=1}^{a_i}\sum_{f=1}^{F} X'_{iaf} \geq 1$ **then**
12: $\quad\quad\quad \Phi \leftarrow \Phi - \{i\}$
13: $\quad\quad$ **end if**
14: $\quad$ **end for**
15: $\quad X_{iaft} \leftarrow X'_{iaf}, \forall i \in \mathcal{N}, \forall a \in \mathcal{A}_i, \forall f \in \mathcal{F}$
16: $\quad$ Calculate $B_{iaf(t+1)}$ by using X_{iaft}
17: $\quad B'_{iaf} \leftarrow B_{iaf(t+1)}, \Psi \leftarrow \emptyset$
18: **end for**
19: **return** $X_{iaft}, \forall i \in \mathcal{N}, \forall a \in \mathcal{A}_i, \forall f \in \mathcal{F}, \forall t \in \mathcal{T}$

Lemma 1. *If the problem in (5)-(12) has a feasible solution, then* $\mathrm{S^2DASA}$ *gives a feasible solution.*

Proof. For all the SUs in the set Φ, Step 11 of $\mathrm{S^2DASA}$ checks if the ILP execution in Step 9 has assigned at least one frequency to that SU. If yes, then Step 12 removes this particular SU i from the set Φ since it is no longer an SU that is not assigned any time slot. Note here that not only the SUs in the set Ψ, but also the other SUs that are assigned some frequency by the ILP execution in Step 9 are removed from the set Φ. In the worst case, it happens that $\mathrm{S^2DASA}$ assigns some frequency to only the SUs in Ψ in every time slot. If N is a multiple of T, then N/T number of SUs are assigned a frequency in every time slot, and clearly all the N number of SUs are guaranteed to satisfy constraint (2) at the end of the algorithm. If N is not a multiple of T, then $|\Psi| = \lceil \frac{N}{T} \rceil$ in the first $\lfloor \frac{N}{\lceil \frac{N}{T} \rceil} \rfloor$ time slots, and condition in line 4 of the algorithm holds true in the next time slot. Therefore, constraint (2) is met for all SUs after $\lfloor \frac{N}{\lceil \frac{N}{T} \rceil} \rfloor + 1 = \lceil \frac{N}{\lceil \frac{N}{T} \rceil} \rceil$ number of time slots. Since $\lceil \frac{N}{\lceil \frac{N}{T} \rceil} \rceil \leq T$, S^{2}DASA returns a feasible solution to the problem in (5)-(12) as long as the original problem has a feasible solution. Notice here that the problem in (5)-(12) has a feasible solution as long as $F \times T \geq N$ because otherwise it is impossible to assign every SU at least one time slot. Hence, S^{2}DASA returns a feasible solution as long as $F \times T \geq N$. In other words, S^{2}DASA guarantees that each SU is assigned at least one time slot. □

Steps 15-17 serve the purpose of calculation and update of the B'_{iaf} values for the next time slot. More precisely, Step 15 of S^{2}DASA sets the X_{iaft} values for that particular time slot t according to the values of X'_{iaf} as a result of the ILP execution in Step 9. Step 16 calculates $B_{iaf(t+1)}$ values for the subsequent time slot $(t+1)$ according to the X_{iaft} values of the current time slot t and possibly the previous time slots by using the formulas specified in (9)-(12). Step 17 updates the values for B'_{iaf} to be used in the iteration of the subsequent time slot.

Every switching delay model will produce a switching delay value, which may depend on the frequency assignments in the previous time slot. If a switching delay model different from the linear model (either constant switching delay or some other model) needs to be used, the only part that needs to be modified in our algorithm is Step 16. In other words, when another switching delay model is used, only the resulting switching delay value used by the calculation of $B_{iaf(t+1)}$ in Step 16 changes; the rest of the algorithm remains the same. In the remainder of this paper, when we mention S^{2}DASA, we implicitly refer to the varying switching delay case with linear model. If a constant delay or some other delay model is used, we explicitly state the name of the switching delay model.

Moreover, Step 17 sets the value of Ψ to the empty set since the SUs in the current Ψ have already been assigned at least one time slot and the Ψ values need to be reconstructed in the next time slot. The algorithm terminates after assignments are made for all time slots.

Theorem 1. *ILP in (13)-(16) is solvable in polynomial time.*

Proof. Build an edge weighted bipartite (multi)graph $G = (\mathcal{N}, \mathcal{F}, \mathcal{E})$ as follows: Add a vertex v_i to $\mathcal{N}$, $\forall i \in \mathcal{N}$. Add a vertex v_f to $\mathcal{F}$, $\forall f \in \mathcal{F}$. For each $B'_{iaf} \neq 0$, add an edge $\{v_i, v_f\}$ to $\mathcal{E}$ with weight $B'_{iaf}, \forall i \in \mathcal{N}, \forall a \in \mathcal{A}_i, \forall f \in \mathcal{F}$. Let $\mathbf{I}$ be the following function associating an interval of natural numbers with each vertex in G: $\mathbf{I}(v_i) = [1, a_i]$ $\forall i \in \Psi$, $\mathbf{I}(v_i) = [0, a_i]$ $\forall i \notin \Psi$, $\mathbf{I}(v_f) = [0, 1]$ $\forall f \in \mathcal{F}$. The problem of finding a sub(multi)graph that maximizes the total edge weights while respecting the constraints about the interval of allowed degrees for each vertex is known to be solvable for any (multi)graph in polynomial time [31], [32]. In particular, if the (multi)graph is bipartite, then the solution for the ILP representing this problem is equal to the solution of its linear program (LP) because the incidence matrix of a bipartite graph is totally unimodular [31]. If an edge with weight B'_{iaf} is selected in the resulting sub(multi)graph, then X'_{iaf} is set to 1; otherwise, it is set to 0. Hence, the theorem holds. □

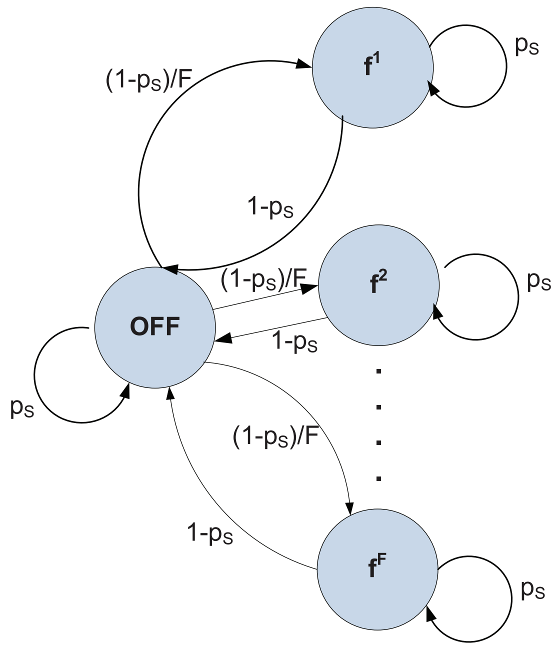


Fig. 2. PU spectrum occupancy model [13].

Since the other steps of S^{2}DASA are clearly solvable in polynomial time, Theorem 1 implies that our proposed heuristic (S^{2}DASA) is solvable in polynomial time. Furthermore, S^{2}DASA is based on running an LP for each time slot. Ellipsoid algorithm [33] can be used to solve LP in polynomial time. In practice, simplex algorithm [33] is widely known to be an efficient algorithm to solve LPs in spite of its exponential complexity.

5 Simulation Results

As in [13], we simulate a centralized CRN cell with 600 meters of radius, $\sigma^2 = 10^{-6}$, $P_{IF_{max}} = 10$ milliwatts, $T = 10$ slots, and $T_s = 100$ ms. Besides, additive white gaussian noise (AWGN) channels are assumed. The dynamicity of the spectral environment stems from two factors: Physical mobility of the SUs and PUs and the changing spectrum occupancy behavior of the PUs. U_{if} values for each scheduling period are possibly different due to the changes in the physical mobility and PU spectrum occupancies. If an SU becomes closer to another PU due to physical mobility, it may need to change its operation frequency in order not to disturb the PU. Likewise, if a PU in the vicinity of the SU starts using a frequency that SU was using, the SU needs to switch to another band.

Both the PUs and the SUs move within the CRN cell according to the random waypoint mobility model with constant velocities of V_p and V_s, respectively, and a staying duration of 10 s between the movement periods. Spectrum usage behavior of the PUs is also as in [13], where a finite state model is used. Fig. 2 illustrates the finite state model used to simulate the PU spectrum occupancy behavior. Each PU is either in the ON state or OFF state. The ON state encompasses one of the F substates, each corresponding to being active using a frequency among a total of F frequencies. The probability of staying in the ON or OFF states is p_S. At the end of each scheduling period, each PU either stays in the same state with probability p_S or changes

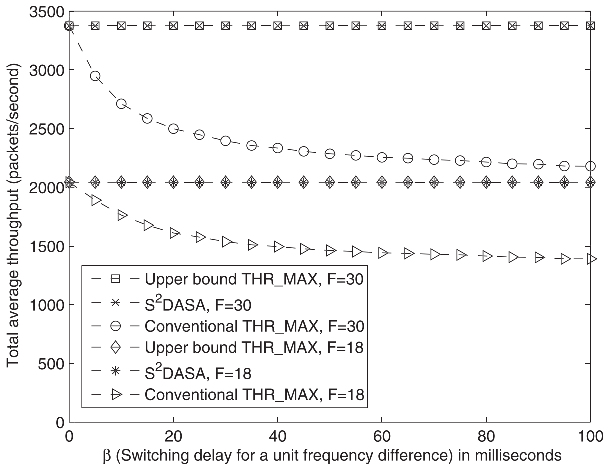


Fig. 3. Average total throughput of all schedulers for varying β.

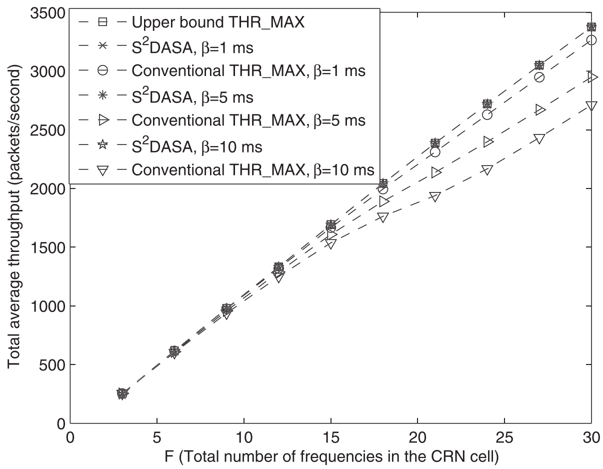


Fig. 4. Average total throughput of all schedulers for varying number of frequencies (F).

its state with probability $1 - p_S$. While switching from the OFF state to the ON state, the probability of selecting each frequency is equally likely; therefore, probability of transition from OFF state to any frequency is $(1 - p_S)/F$. In a slowly varying spectral environment, p_S value is usually high; hence, we selected the p_S value as 0.9 in our simulations. Hence, our simulation results demonstrate that switching delay is an important factor even in a slowly varying spectral environment.

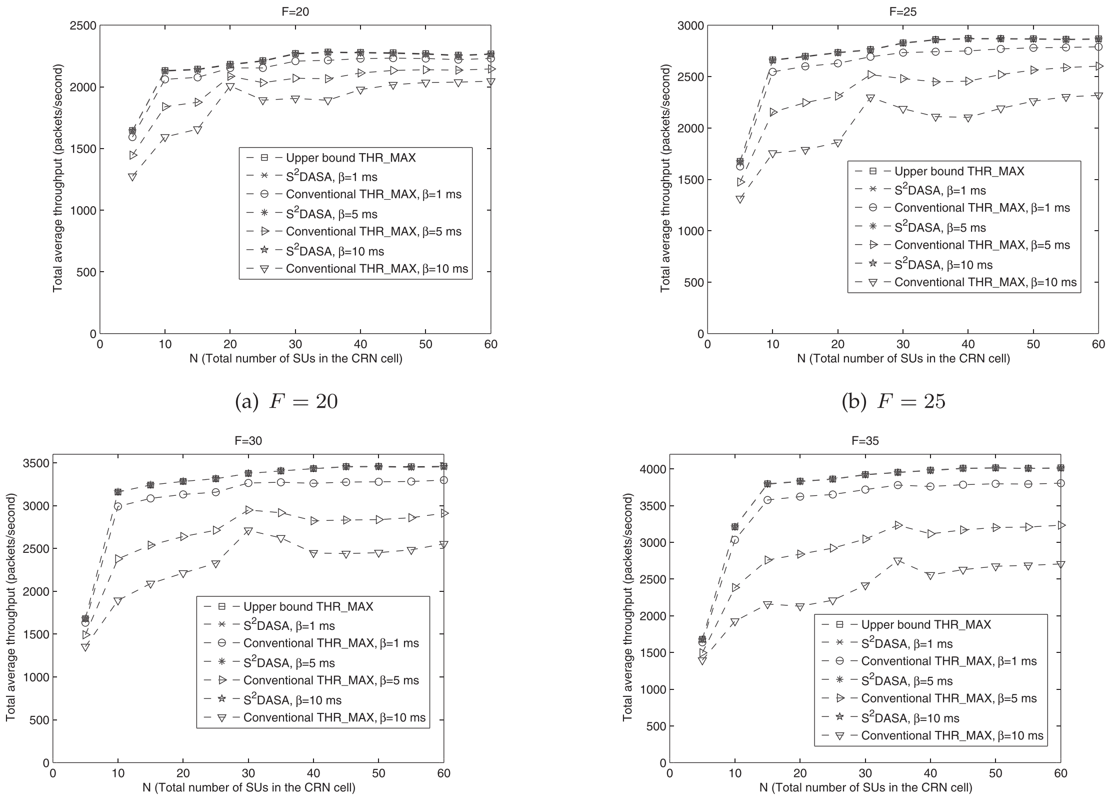
The original simulation results in [13] for the throughput maximizing scheduler, which we refer to here by *upper bound THR_MAX*, do not consider the spectrum switching delay. If we calculate the throughput values of the resulting frequency and time slot assignments in [13] according to the spectrum switching delay model outlined in Section 3, which depends on the difference between the current and the previously assigned frequency, we obtain the actual throughput values of the results in [13], which we refer to here by *conventional THR_MAX*. Since upper bound THR_MAX values are calculated by assuming that switching delay has no impact on the throughput performance, upper bound THR_MAX is an upper bound for the performance of the optimal solution to our formulated problem in (5)-(12), which is in turn an upper bound for our proposed heuristic algorithm S^{2}DASA. Therefore, the optimal solution of our formulated problem in (5)-(12) has to lie between the values for upper bound THR_MAX and S^{2}DASA.

As in [13], we assume that all the PUs and SUs move with a constant velocity of V_p and V_s, respectively. We denote the number of PUs in the cell by M. In all experiments in this paper, we take $M = 20$, $V_p = V_s = 13\,\text{m/s}$, and $a_i = 3\,\forall i \in \mathcal{N}$. In [13], a statistical method is used to calculate the number of samples to be taken so that the sample mean of all the samples are ± 0.5 of the actual mean with a 95 percent confidence level. The number of samples we take for S^{2}DASA in all the experiments is the same as the corresponding ones in upper bound THR_MAX, as calculated in [13].

We evaluate the impact of F and β in two different sets of experiments. In the first one, we vary the value of β and plot the results for $F = 18$ and $F = 30$. We set $N = 30$ and compare the results of upper bound THR_MAX, conventional THR_MAX, and S^{2}DASA in Fig. 3. The results show that for both $F = 18$ and $F = 30$, S^{2}DASA yields very close performance to upper bound THR_MAX, which can be regarded as an upper bound to the optimization problem in (5)-(12). Since the results are very close, they look like almost the same in the figure; however, there are subtle differences in reality. For instance, for $F = 30$, the throughput that upper bound THR_MAX yields is 3376.6 packets/time slot, whereas the throughput that S^{2}DASA yields for $\beta = 50$ is 3375.6 packets/time slot. In all of the simulation results presented in this paper, the throughput performance of S^{2}DASA is only slightly less than the one of upper bound THR_MAX; therefore, they cannot be visually differentiated in the figures. We also observe in Fig. 3 that the decrease in conventional THR_MAX as β increases for $F = 30$ is larger than the decrease for $F = 18$. This implies that S^{2}DASA results in higher savings from throughput as F and β increase. Furthermore, we can also observe that the performance difference between S^{2}DASA and upper bound THR_MAX remains very little as β increases. These results demonstrate that the performance of our proposed algorithm S^{2}DASA is not only very close to its upper bound, but also robust to changes in the hardware switching delay.

In the second set of experiments, we set $N = 30$ and vary the value of F. We then plot the results in Fig. 4 for $\beta = 1$, $\beta = 5$, and $\beta = 10$ ms. We again observe that the performance of S^{2}DASA is very close to upper bound THR_MAX, in addition to being robust to changes in the switching delay. As in Fig. 3, the throughput difference between S^{2}DASA and upper bound THR_MAX is so small that the two results cannot be visually differentiated in Fig. 4. We see in Fig. 4 that throughput increases linearly with the number of frequencies in all cases. Furthermore, the performance difference between S^{2}DASA and conventional THR_MAX widens as the number of frequencies increases; hence, throughput savings achieved by our proposed algorithm S^{2}DASA increase as the number of frequencies increases. This result demonstrates that switching delay becomes an even more important factor as the CRNs proliferate by operating in a wider range of frequencies. Therefore, our proposed algorithm S^{2}DASA is promising to be even more useful in the CRNs of the future.

In the third set of experiments, we evaluate the impact of varying number of SUs (N). Fig. 5 shows the total average throughput for different values of F. Figs. 5a, 5b, 5c, and 5d show the results for $F = 20$, $F = 25$, $F = 30$, and



(a) $F = 20$

(b) $F = 25$

(c) $F = 30$

(d) $F = 35$

Fig. 5. Average total throughput of all schedulers for varying number of SUs (N).

$F = 35$, respectively. We observe in these figures that the throughput savings achieved by our algorithm are significant even when $N = 10$ and they remain significant as N increases. We also observe that there is a ripple effect in conventional THR_MAX at the point where $N = F$; i.e., the real throughput at $N = F$ is higher than the real throughput at $N = F - 5$ and $N = F + 5$. The ripple is more evident for the $\beta = 10$ case. Recall that in all the simulations, $a_i = 3\ \forall i \in \mathcal{N}$ and there is a constraint that each SU has to be assigned at least one time slot. Until the point where $N = F$, it gets more difficult to assign each SU at least one time slot as the number of SUs increases and the scheduler mostly uses more than one interface. It happens most of the time that an SU is assigned a frequency for a particular time slot but not assigned any frequency in the subsequent time slot because the other SUs need to be assigned some frequency to satisfy the constraint that each SU is assigned at least one time slot. Notice here that the scheduler does not mandate each SU to be assigned a frequency in each time slot, but only mandates that each SU is assigned at least one time slot during the course of the entire scheduling period. When an SU is not assigned a frequency in a particular time slot but assigned a frequency in the subsequent time slot, the hardware switching delay has less impact on the throughput performance since there is more time available to achieve the frequency switching, which is represented by our formulation for Δ_{iaft} in (11). Since this situation occurs more frequently as N value approaches F, we see an increase in the throughput values of conventional THR_MAX. When $N > F$, less number of interfaces are used and the impact of switching delay increases. Let us call an interface *active* if it is used at least once for data transmission during the entire scheduling period. Recall that time slot t is called a silent time slot for an active interface a of SU i if it is not assigned any frequency in time slot t. Recall also that time slot t is otherwise called a busy time slot for interface a of SU i. Note here that this time slot might be in use by some other interface of this SU or of another SU. When we divide the total number of silent time slots in a scheduling period by the number of active interfaces, we find the *"average number of silent time slots per active interface."* Then we find the sample mean of this *"average number of silent time slots per active interface"* over all iterations (all samples). The average number of silent time slots for $F = 20$ are 4.51, 10.39, 11.62, 10.5, and 10.28 percent and the average number of active interfaces are 2.48, 2.34, 1.6, 1.5, and 10.28 for $N = 10$, $N = 15$, $N = 20$, $N = 25$, and $N = 30$, respectively. The ripple effect at $F = N$ can also be observed for the average number of silent time slots per active interface. We have observed the same situation for $F = 25$, $F = 30$, and $F = 35$. The similarity of this behavior to the throughput performance corroborates our explanation for the ripple effect observed at the real throughput performance at $F = N$.

In the fourth set of experiments in Fig. 6, we compare our algorithm S^{2}DASA with upper bound THR_MAX, conventional THR_MAX and S^{2}DASA with the constant delay scenario. Figs. 6a, 6b, and 6c show the results for $\beta = 1$, $\beta = 5$, and $\beta = 10$ cases, respectively. We can see that S^{2}DASA (varying delay) performs better than S^{2}DASA with constant delay case in all scenarios. Furthermore, its performance is close to upper bound THR_MAX. The results corroborate that the assumption of

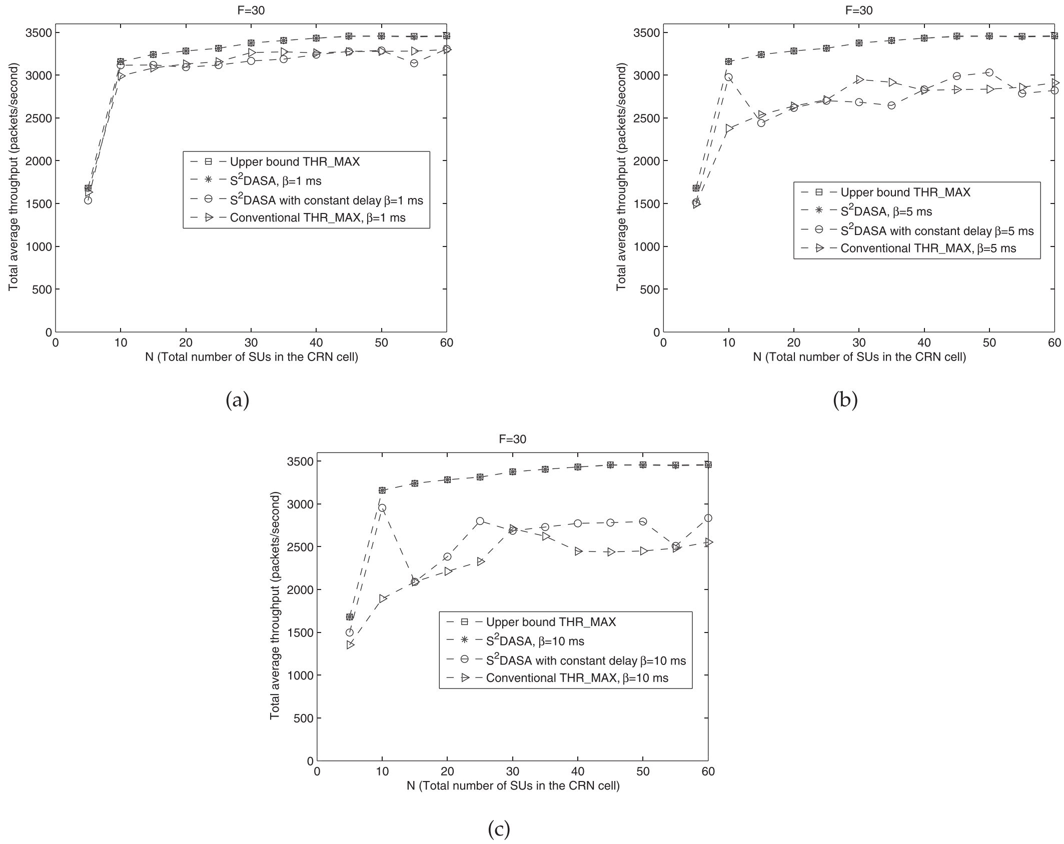


Fig. 6. Average total throughput comparison with constant switching delay case.

constant switching delay in a CRN setting can lead to poor throughput performance. In some cases, assuming that the switching delay is constant can lead to even lower throughput than conventional THR_MAX. The superiority of $\mathrm{S^2DASA}$ with varying switching delay over $\mathrm{S^2DASA}$ with constant switching delay becomes more evident as β (hardware switching delay) increases. To sum up, Fig. 6 demonstrates that the gist of our paper, which is to take into account in the scheduling algorithm for CRNs the increasing spectrum switching delay due to switching to further away frequencies, is vital in CRNs.

Recall that the optimal solution of our formulated problem in (5)-(12) has to lie between the values for upper bound THR_MAX and $\mathrm{S^2DASA}$. In all the simulation results in this paper, the performance of $\mathrm{S^2DASA}$ is already very close to upper bound THR_MAX. Therefore, for all practical purposes, our proposed algorithm $\mathrm{S^2DASA}$ has remarkable performance.

6 Conclusion

We formulate in this paper a scheduling problem that considers different hardware delays that occur during switching to different frequency bands. We propose a polynomial time heuristic algorithm called $\mathrm{S^2DASA}$ to solve our formulated problem. The simulation results show that the throughput that $\mathrm{S^2DASA}$ yields is very close to its upper bound. Moreover, $\mathrm{S^2DASA}$ is robust to changes in the hardware spectrum switching delay. Furthermore, throughput savings it achieves increase as the number of frequencies in the CRN cell (F) and the hardware switching delay for a unit frequency difference (β) increases. Furthermore, the throughput savings of our algorithm are significant even when there are a small number of SUs, and the savings remain significant as the number of SUs increases. Simulation results demonstrate that our idea of taking into account different hardware delays that occur during switching to different frequency bands is essential for CRNs since the assumption of constant switching delay can lead to low throughput performance.

The superior throughput performance of $\mathrm{S^2DASA}$ calls for techniques that quantify its worst case performance analytically. As a future work, we plan to derive an analytical lower bound for the throughput performance of our proposed algorithm $\mathrm{S^2DASA}$. To this end, we plan to utilize algorithmic graph theory and approximation algorithms.

Acknowledgments

This work was supported by the State Planning Organization of Turkey (DPT) under grant number DPT-2007K 120610 and the Scientific and Technological Research Council of Turkey (TUBITAK) under grant number 109E256. This work was conducted when Didem Gözüpek was a PhD candidate at Boğaziçi University and in part when Seyed Buhari was a visiting researcher at Boğaziçi University.

References

[1] I. Akyildiz, W. Lee, M. Vuran, and S. Mohanty, "NeXt Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: A Survey," *Computer Networks*, vol. 50, no. 13, pp. 2127-2159, 2006.

[2] B. Wang and K. Liu, "Advances in Cognitive Radio Networks: A Survey," *IEEE J. Selected Topics in Signal Processing,* vol. 5, no. 1, pp. 5-23, Feb. 2011.

[3] H. Kim and K. Shin, "Efficient Discovery of Spectrum Opportunities with MAC-Layer Sensing in Cognitive Radio Networks," *IEEE Trans. Mobile Computing,* vol. 7, no. 5, pp. 533-545, May 2008.

[4] G. Cheng, W. Liu, Y. Li, and W. Cheng, "Spectrum Aware On-Demand Routing in Cognitive Radio Networks," *Proc. IEEE Second Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN),* 2007.

[5] G. Cheng, W. Liu, Y. Li, and W. Cheng, "Joint On-Demand Routing and Spectrum Assignment in Cognitive Radio Networks," *Proc. IEEE Comm. Int'l Conf. (ICC),* 2007.

[6] I. Filippini, E. Ekici, and M. Cesana, "Minimum Maintenance Cost Routing in Cognitive Radio Networks," *Proc. IEEE Sixth Int'l Conf. Mobile Adhoc and Sensor Systems (MASS),* 2009.

[7] K. Chowdhury and M. Felice, "SEARCH: A Routing Protocol for Mobile Cognitive Radio Ad-Hoc Networks," *Computer Comm.,* vol. 32, no. 18, pp. 1983-1997, 2009.

[8] J. Chen, H. Li, J. Wu, and R. Zhang, "STARP: A Novel Routing Protocol for Multi-Hop Dynamic Spectrum Access Networks," *Proc. ACM Workshop Mobile Internet through Cellular Networks,* 2009.

[9] S. Krishnamurthy, M. Thoppian, S. Venkatesan, and R. Prakash, "Control Channel Based MAC-Layer Configuration, Routing and Situation Awareness for Cognitive Radio Networks," *Proc. IEEE Military Comm. Conf. (MILCOM),* 2005.

[10] M. Thoppian, S. Venkatesan, R. Prakash, and R. Chandrasekaran, "MAC-Layer Scheduling in Cognitive Radio Based Multi-Hop Wireless Networks," *Proc. Int'l Symp. World of Wireless, Mobile and Multimedia Networks (WoWMoM),* 2006.

[11] Y. Yuan, P. Bahl, R. Chandra, T. Moscibroda, and Y. Wu, "Allocating Dynamic Time-Spectrum Blocks in Cognitive Radio Networks," *Proc. ACM MobiHoc,* 2007.

[12] R. Urgaonkar and M. Neely, "Opportunistic Scheduling with Reliability Guarantees in Cognitive Radio Networks," *Proc. IEEE INFOCOM,* 2008.

[13] D. Gözüpek and F. Alagöz, "An Interference Aware Throughput Maximizing Scheduler for Centralized Cognitive Radio Networks," *Proc. IEEE 21st Int'l Symp. Personal Indoor and Mobile Radio Comm. (PIMRC),* 2010.

[14] P. Mitran, "Interference Reduction in Cognitive Networks via Scheduling," *IEEE Trans. Wireless Comm.,* vol. 8, no. 7, pp. 3430-3434, July 2009.

[15] Z. Zhao, Z. Peng, S. Zheng, and J. Shang, "Cognitive Radio Spectrum Allocation Using Evolutionary Algorithms," *IEEE Trans. Wireless Comm.,* vol. 8, no. 9, pp. 4421-4425, Sept. 2009.

[16] S. Maroug, "Frequency-Switching Speed and Post-Tuning Drift Measurement of Fast-Switching Microwave-Frequency Synthesisers," *IET Science, Measurement and Technology,* vol. 1, no. 2, pp. 82-86, 2007.

[17] H. Ma, L. Zheng, and X. Ma, "Spectrum Aware Routing for Multi-Hop Cognitive Radio Networks with a Single Transceiver," *Proc. Third Int'l Conf. Cognitive Radio Oriented Wireless Networks and Comm. (CrownCom),* 2008.

[18] "TCI 7234 Wideband SHF Tuner Data Specification," http://www.tcibr.com/ufiles/Library/7234_webp.pdf, 2010.

[19] "TCI 715 Data Specification," http://www.tcibr.com/ufiles/Library/715_webp.pdf, 2008.

[20] M. Cesana, F. Cuomo, and E. Ekici, "Routing in Cognitive Radio Networks: Challenges and Solutions," *Ad Hoc Networks,* vol. 9, pp. 228-248, 2011.

[21] "TCI 735 Data Specification," http://www.tcibr.com/ufiles/Library/735_webp.pdf, 2010.

[22] "TCI 745 Data Specification," http://www.tcibr.com/ufiles/Library/745_webs.pdf, 2010.

[23] O. Ileri, D. Samardzija, and N. Mandayam, "Demand Responsive Pricing and Competitive Spectrum Allocation via a Spectrum Server," *Proc. IEEE First Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN),* pp. 194-202, 2005.

[24] W. Lee and I. Akyildiz, "Joint Spectrum and Power Allocation for Inter-Cell Spectrum Sharing in Cognitive Radio Networks," *Proc. IEEE Third Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN),* pp. 1-12, 2008.

[25] H. Kamal, M. Coupechoux, and P. Godlewski, "Inter-Operator Spectrum Sharing for Cellular Networks Using Game Theory," *Proc. IEEE 20th Int'l Symp. Personal Indoor and Mobile Radio Comm. (PIMRC),* pp. 425-429, 2009.

[26] T. Yamada, D. Burgkhardt, I. Cosovic, and F. Jondral, "Resource Distribution Approaches in Spectrum Sharing Systems," *EURASIP J. Wireless Comm. and Networking,* vol. 2008, p. 8, 2008.

[27] P. Koch and R. Prasad, "The Universal Handset," *IEEE Spectrum,* vol. 46, no. 4, pp. 36-41, Apr. 2009.

[28] *IEEE 802.22,* http://www.ieee802.org/22/, 2012.

[29] M. Yun, Y. Zhou, A. Arora, and H. Choi, "Channel-Assignment and Scheduling in Wireless Mesh Networks Considering Switching Overhead," *Proc. IEEE Int'l Conf. Comm. (ICC '09),* pp. 1-6, 2009.

[30] R. Hemmecke, M. Koppe, J. Lee, and R. Weismantel, "Nonlinear Integer Programming," *50 Years of Integer Programming 1958-2008,* pp. 561-618, 2010.

[31] R. Pulleyblank, "Faces of Matching Polyhedra," PhD thesis, 1973.

[32] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency.* Springer, 2003.

[33] A. Schrijver, *Theory of Linear and Integer Programming.* Wiley-Interscience, 1999.

Didem Gözüpek received the BS degree (high honors) in telecommunications engineering from Sabanci University, Istanbul, Turkey, in 2004, the MS degree in electrical engineering from the New Jersey Institute of Technology in 2005, and the PhD degree in computer engineering from Boğaziçi University, Istanbul, Turkey, in 2012. She is currently an assistant professor in the Computer Engineering Department, Gebze Institute of Technology, Kocaeli, Turkey. From 2005-2008, she worked as an R&D engineer for a telecommunications company in Istanbul. She received the Boğaziçi University Best PhD Thesis Award in 2012 and the Dr. Serhat Özyar Honorary Award in 2013. She was a finalist for the Google Anita Borg Memorial Scholarship in 2009. Her main research interests include scheduling and resource allocation in communication networks, algorithmic graph theory, and approximation algorithms.

Seyed Buhari received the BE degree in computer engineering from Mepco Schlenk Engineering College (Madurai Kamaraj University), India, in 1996, the ME degree in computer science and engineering from the PSG College of Technology (Bharathiar University), India, in 1998, and the PhD degree in information technology from Multimedia University, Malaysia. He is currently an assistant professor in the Department of Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia. From 2006-2012, he worked as a senior lecturer in the Faculty of Science, University Brunei Darussalam. From 2000-2006, he was with the Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Saudi Arabia. His current research interests include the areas of cognitive radio networks, grid computing, IPv6 performance testing, and high performance computing.

Fatih Alagöz received the BSc degree in electrical engineering from Middle East Technical University, Turkey, in 1992, and the MSc and DSc degrees in electrical engineering from George Washington University in 1995 and 2000, respectively. He is currently a professor in the Department of Computer Engineering, Boğaziçi University, Istanbul, Turkey. From 2001-2003, he was with the Department of Electrical Engineering, United Arab Emirates University. In 1993, he was a research engineer in a missile manufacturing company, Muhimmatsan AS, Turkey. His current research interests include satellite networks, wireless networks, sensor networks, and UWB communications. He has contributed to 10 research projects for organizations including the US Army of Intelligence Center, US Naval Research Laboratory, UAE Research Fund, Turkish Scientific Research Council, State Planning Organization of Turkey, and BAP. He has edited five books and published more than 100 scholarly papers. He is the satellite systems advisor to the Kandilli Earthquake Research Institute, Istanbul, Turkey. He has served on many technical committees and organized many technical sessions at international conferences. He has numerous professional awards. He is a member of the IEEE Satellite and Space Communications Technical Committee.