Allocating Dynamic Time-Spectrum Blocks In Cognitive Radio Networks

Yuan Yuan, Paramvir Bahl, Ranveer Chandra, Thomas Moscibroda, Yunnan Wu
Microsoft Research, Redmond, WA 98052

ABSTRACT

A number of studies have shown the abundance of unused spectrum in the TV bands. This is in stark contrast to the overcrowding of wireless devices in the ISM bands. A recent trend to alleviate this disparity is the design of Cognitive Radios, which constantly sense the spectrum and opportunistically utilize unused frequencies in the TV bands. In this paper, we introduce the concept of a *time-spectrum block* to model spectrum reservation, and use it to present a theoretical formalization of the spectrum allocation problem in cognitive radio networks. We present a centralized and a distributed protocol for spectrum allocation and show that these protocols are close to optimal in most scenarios. We have implemented the distributed protocol in QualNet and show that our analysis closely matches the simulation results.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless Communication*

General Terms

Algorithms, Theory

Keywords

Cognitive radio networks, spectrum allocation, interval coloring, adaptive channel-width

1. INTRODUCTION

The number of wireless devices in the ISM bands have proliferated significantly over the last decade. There are a large number of Wi-Fi and Bluetooth devices, and WiMax and city wide mesh networks are expected to add to this congestion. These technologies interfere with each other and hurt the overall performance. In contrast, other portions of the spectrum are extremely underutilized. Recent measurements have shown that only 5% of the spectrum from 30 MHz to 30 GHz is used in the US [5]. In a recent proposal, the Federal Communications Commission (FCC) is exploring ways to reduce this imbalance by permitting the use of unlicensed devices in the licensed bands when there is no licensed (primary) user [1] operating on it. Parts of the spectrum not in active use by the primary users are often referred to as *white spaces*.

Cognitive radios are a promising technology for enabling unlicensed devices to efficiently use the white spaces. These radios dynamically identify portions of the spectrum that are not in use by primary users, and configure the radio to operate in the appropriate white space. Prior work has shown that white spaces are fragmented and of different sizes. The availability of white spaces is temporal and depends on the geographic location of the radio [5, 30]. Thus, a key challenge in the design of cognitive radios is the dynamic allocation of white spaces to different radios in the network. The efficiency of the spectrum allocation determines both the network's throughput as well as the overall spectrum utilization.

The problem of allocating spectrum in cognitive radio networks poses new challenges that do not arise in several wireless technologies, including Wi-Fi. For example, cognitive radios can dynamically adjust the center frequency and the bandwidth (channel-width) for each transmission [30]. In contrast, traditional wireless networks use channels of fixed, pre-determined width. For example, each channel in IEEE 802.11a [2] is defined by the standard to be 20 MHz wide. Due to the lack of pre-defined channel widths, cognitive radios pose a new spectrum allocation problem: *Which node should use how wide a spectrum-band at what center-frequency and for how long?* We note that the formal analysis of this spectrum allocation problem is more challenging. So far, formal treatment of scheduling and channel assignment problems in multi-hop wireless networks has typically boiled down to *coloring* or *matching* problems, or variants thereof [25]. However, while the notion of *colors* may be suitable to model channels of predefined bandwidth, modeling variable communication networks with variable channel-widths is more complicated.

In this paper, we introduce and study the dynamic spectrum allocation problem in cognitive radio networks. Our goal is two-fold. First, we aim to formalize the problem with respect to metrics that define a good solution. Based on these theoretical underpinnings, we derive guidelines that govern the design of an efficient practical spectrum allocation protocol. Second, we propose and analyze b-SMART, a first-cut practical solution that can be implemented as part

MobiHoc'07, September 9–14, 2007, Montreal, Quebec, Canada.

of our cognitive radio platform [30]. Specifically, we make the following contributions:

- We develop a theoretical framework for dynamic spectrum allocation in cognitive radio networks. The framework captures the essential features of cognitive radios, such as frequency agility and adaptive bandwidth, and it introduces the concept of a *time-spectrum block*, which represents the time for which a cognitive radio uses a portion of the spectrum. We use this concept to define the spectrum allocation problem as the packing of time-spectrum blocks in a two dimensional time-frequency space, such that the demands of all nodes are satisfied best possible. Our model represents a fundamental departure from the conventional analysis framework that explicitly or implicitly uses variants of graph-coloring problems for maximizing spectrum usage (scheduling, frequency-allocation, etc.).
- We prove the problem to be NP Hard and present an approximation algorithm that assumes full knowledge of user demands and performs within a small constant factor of the optimum, regardless of network topology.
- We propose b-SMART (distributed spectrum allocation over white spaces), a practical, distributed protocol to solve the spectrum allocation problem in real cognitive networks. b-SMART enables each node to dynamically decide on a time-spectrum block based only on local information. Using both analysis and extensive simulations in QualNet, we show that b-SMART achieves high throughput and fairness under various scenarios.

2. SPECTRUM ALLOCATION PROBLEM

Our model for a cognitive radio is based on the KNOWS prototype we are currently developing at Microsoft [30]. The prototype consists of a reconfigurable transceiver and a scanner. The parameters of the transceiver, such as the center frequency and bandwidth, can be adjusted within 10s of microseconds. The transceiver can only tune to a *contiguous* segment of spectrum. Due to hardware limitations, the possible bandwidth values are a discrete set in the range of $[b_{min}, b_{max}]$, where b_{min} and b_{max} denote the lower and upper bounds of the supported bandwidth, respectively. The largest usable bandwidth is typically below 40 MHz. We are also including an extra receiver radio in the prototype to enable a common control channel for exchanging spectrum usage information. In our previous work [30], we have shown how the extra receiver can be implemented using the scanner radio.

2.1 Formal Problem Formulation

While our model and problem formulation is specifically tuned to capture the physical capabilities of our KNOWS prototype, it is general enough to allow for a wide range of analytical and algorithmic studies of spectrum allocation problems in cognitive radio networks. The model is primarily intended to capture the novel algorithmic challenges arising from the nodes's capability to adaptively change their bandwidth (in addition to their center-frequency).

Definitions and Notation: We model the cognitive radio network as a set of n nodes $V = \{v_1, \ldots, v_n\}$ located in the two-dimensional Euclidean plane. Let $d(v_i, v_j)$ denote the Euclidean distance between v_i and v_j. Let f_{bot} and f_{top} denote the lower and upper end of the accessible target spectrum, e.g., $f_{bot} = 470MHz$ and $f_{top} = 698MHz$ in the TV spectrum [1]. Each node $v_i \in V$ is equipped with a radio transceiver that is capable of dynamically accessing any *contiguous* frequency band $[f, f+\Delta f]$ for all $f_{bot} \leq f \leq f + \Delta f \leq f_{top}$, as long as $b_{min} \leq \Delta f \leq b_{max}$.

For each pair of nodes $(v_i, v_j) \in V$ within mutual communication range, $D_{ij}(t, \Delta t)$ denotes the *demand* in bit/s that v_i would like to transmit to v_j during time interval $[t, t + \Delta t]$. This link-based demand subsumes the traffic of all flows that are routed over this particular link[1] and our definition captures the fact that demands may vary both between different links and also on a single link over time.

The crucial difference between the spectrum access using predefined channels of fixed channel-width and the dynamic access in cognitive radio networks is that (i) the *bandwidth* (channel-width) of the spectrum allocated to different links becomes an additional variable, and (ii) the radio parameters can be adjusted in a fine timescale. We say that a *time-spectrum block* $B_{ij}^k = (t_k, \Delta t_k, f_k, \Delta f_k)$ is assigned to link (v_i, v_j) if sender v_i is assigned the contiguous frequency band $[f_k, f_k + \Delta f_k]$ of bandwidth Δf_k during time interval $[t_k, t_k + \Delta t_k]$. We can therefore view the dynamic spectrum allocation problem as dynamic packing of time-spectrum blocks into a three-dimensional resource, consisting of time, frequency, and space. Suppose node $v_i \in V$ transmits to receiver v_j using a time-spectrum block B_{ij}^k. The amount of data v_j receives in $[t_k, t_k + \Delta t_k]$ can be expressed as [14]:

$$C_{ij}(B_{ij}^k) = \Delta f_k \cdot \mathcal{A}(f_k, \Delta f_k) \cdot \Delta t_k \cdot \mathcal{B}(\Delta t_k). \qquad (1)$$

The function $\mathcal{A}(f_k, \Delta f_k)$ characterizes how well the band $[f_k, f_k + \Delta f_k]$ can be utilized, which depends on the frequency, the bandwidth, the spectrum condition as well as on the hardware. The function $\mathcal{B}(\Delta t_k)$ captures the hardware and protocol-specific overhead incurred when accessing the spectrum (for example the overhead incurred by contentions and sending acknowledgements). We call $C_{ij}(B_{ij}^k)$ the *capacity* of the allocated time-spectrum block. Under ideal channel conditions and disregarding any potential overhead, the "ideal" capacity simplifies to $C_{ij}(B_{ij}^k) = \gamma \Delta f_k \Delta t_k$ for some constant γ. However, this definition is oversimplified in the sense that it allows for an overhead-free slicing of time into infinitely fine-grained blocks. Therefore, in this paper, we study a more realistic capacity definition that precludes this possibility. We assume spectrum utilization to be linear in the bandwidth, but $\mathcal{B}(\Delta t_k) = (1 + \beta/\Delta t_k)$, i.e.,

$$C_{ij}(B_{ij}^k) = \alpha \Delta f_k (\Delta t_k - \beta), \qquad (2)$$

for a constant β, that represents the overhead incurred when accessing the spectrum band. This overhead may include the time overhead of switching frequency or the time used for medium access contention.

Interference Model:

In principle, the *dynamic spectrum allocation problem* can be analyzed using a variety of underlying network and communication models, for instance the classic *protocol* and *physical models* [12]. In this paper, we consider a cognitive radio interference model based on the simple protocol model.[2] In this model, each sender v_i is associated with

[1] Our definition essentially describes a link-based notion of scheduling. All definitions and theoretical results in this paper can easily be extended to broadcast scheduling problems, in which demands are determined *per node*, instead of per link.

[2] Even in classic single-channel networks, the protocol and physical

a transmission range R_i^t and a (larger) interference range R_i^{int}. A message sent over a link (v_i, v_j) is possible if there is no simultaneous transmitter v_z such that v_j is in v_z's interference range R_z^{int}. That is, two time-spectrum blocks $B_{ij}^k(t_k, \Delta t_k, f_k, \Delta f_k)$ and $B_{gh}^\ell(t_\ell, \Delta t_\ell, f_\ell, \Delta f_\ell)$ are mutually *non-interfering* if one of the following conditions is satisfied.

- $d(v_j, v_g) > R_j^{int}$ and $d(v_i, v_h) > R_i^{int}$ (space separation)
- $\max\{f_k, f_\ell\} \geq \min\{f_k+\Delta f_k, f_\ell+\Delta f_\ell\}$ (freq. separation)
- $\max\{t_k, t_\ell\} \geq \min\{t_k + \Delta t_k, t_\ell + \Delta t_\ell\}$ (time separation)

Since a cognitive radio incorporates a scanner to detect primary signals, mitigating interference among secondary users is the key challenge facing the dynamic spectrum allocation. We define a set of *prohibited bands* $\mathcal{P} = \{P_1, \ldots, P_L\}$, where every $P_\ell \in \mathcal{P}$ denotes a spectrum band $P_\ell = [f_y, f_z]$ that is used by a primary station and detected by the scanner. A *spectrum allocation schedule* $\mathcal{S}$ is an assignment of time-spectrum blocks B_{ij}^k to links $(v_i, v_j) \in E$, such that no two assigned blocks $B_{ij}^k, B_{gh}^\ell \in \mathcal{S}$ interfere and no prohibited spectrum is used. Formally, a schedule is S *feasible* if the following conditions hold.

- No two assigned time-spectrum blocks interfere
- $[f_i, f_i + \Delta f_i] \cap P_\ell = \emptyset$ for every assigned block B_{ij}^k and every $P_\ell \in \mathcal{P}$.

Dynamic Spectrum Allocation Problem:

Equipped with these definitions, we now state the dynamic spectrum allocation problem.

Given dynamic demands $D_{ij}(t_k, \Delta t_k)$, a dynamic spectrum allocation protocol computes a feasible spectrum allocation schedule $\mathcal{S}$ that assigns non-interfering dynamic time-spectrum blocks to links such that demands are satisfied as much as possible, i.e., the spectrum is efficiently utilized.

Numerous specific measures and combinatorial optimization problems can be derived from the above formulation. The first such measure of interest that characterizes the performance of protocols is *throughput*. As $C_{ij}(B_{ij}^k)$ is the maximum amount of data that can be sent over link (v_i, v_j) in time-spectrum block B_{ij}^k, the throughput of link (v_i, v_j) in $[t_k, t_k + \Delta t_k]$ is

$$T_{ij}(B_{ij}^k) = \min\{D_{ij}(t_k, \Delta t_k), C_{ij}(B_{ij}^k)\} \quad (3)$$

and the *throughput maximization problem* asks for a feasible schedule $\mathcal{S}$ that maximizes

$$T_{max} = \sum_{(v_i,v_j)\in E} \sum_k T_{ij}(B_{ij}^k).$$

As the throughput measure does not account for any notion of fairness, we want to maintain *proportionally-fair throughput* among different demands. For some demand $D_{ij}(t_k, \Delta t_k)$, let $\mathcal{I}_{ij}$ denote all time intervals $[t, t + \omega]$ for some fixed duration ω, for which $[t, t + \omega] \in [t_k, t_k + \Delta t_k]$. Then, the *minimum proportionally-fair throughput* $T_{minfair}(\omega)$ is

$$T_{minfair}(\omega) = \min_{(v_i,v_j)\in E} \min_{[t,t+\omega]\in\mathcal{I}_{ij}} \frac{T_{ij}(t, t+\omega)}{\omega \cdot D_{ij}(t_k, \Delta t_k)},$$

where $T_{ij}(t, t+\omega)$ is the throughput achieved during interval $[t, t + \omega]$. A high minimum proportionally-fair throughput therefore guarantees that in every time-interval of length ω, every demand gets its fair share of throughput. The shorter ω is chosen, the more short-term and fine-grained this notion of fairness becomes. In particular, a protocol that guarantees good minimum proportionally-fair throughput for very small values of ω (say, in the order of a few milliseconds) leads to low latencies and minimizes jitter.

2.2 Complexity

As it turns out, even simple and highly restricted variants of the dynamic spectrum allocation problem are computationally hard. In particular, the throughput problems are hard even if demands have infinite duration (i.e., $\Delta t = \infty$), and sometimes even in single-hop scenarios or when there are no prohibited bands, i.e., $|\mathcal{P}| = 0$.

Theorem 2.1. *The proportionally-fair throughput maximization problem is NP-complete even if $|\mathcal{P}| = 0$.*

Proof. The proof is by reduction to the NP-complete 3-chromatic number problem in unit disk graphs, which is to determine whether a given unit disk graph has a feasible vertex coloring using at most 3 colors [11]. More precisely, we consider a slightly more restricted family of unit disk graphs in which every node appears in a clique of size 3. The proof of [11] can be adjusted to show that the 3-chromatic number problem remains hard in such graphs. The reduction works as follows: Given an instance of the 3-chromatic number problem in unit disk graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in which every node appears in a clique of size 3, we create the following instance of the dynamic spectrum allocation problem: For each node $v_i \in \mathcal{V}$, there are two cognitive radio nodes, a sender w_i and a receiver w_i'. Node locations of nodes w_i in the plane correspond to the positions of v_i in V, but are scaled in such a way that R_i^{int} corresponds to the unit distance of $\mathcal{G}$. Finally, each node w_i' is placed such a way that it sees the same set of interfering nodes as w_i, i.e., for all w_j, $i \neq j$,

$$d(w_i, w_j) \leq R_j^{int} \iff d(w_i', w_j) \leq R_j^{int}.$$

A close inspection of the hard instances derived in [11] reveals that finding such a placement is always possible. Finally, let the demand D_{ii}' of (w_i, w_i') be $D_{ii}' = (f_{top} - f_{bot})/3$, and all other demands $D_{ij} = 0$. All demands are invariable in time. It holds that $\mathcal{G}$ is 3-colorable exactly if the maximum total throughput is $T_{max} = f_{top} - f_{bot}$. In particular, if $\mathcal{G}$ is 3-colorable, each sender w_i with color c_i in the original graph is assigned a time-spectrum block $(0, \infty, f_{bot} + (c_i - 1)(f_{top} - f_{bot})/3, (f_{top} - f_{bot})/3)$. By definition of the reduction to the coloring problem and the placement of receivers w_i', all these blocks are non-interfering and hence, every demand has a proportional throughput of exactly $T_{ij}/D_{ij} = 1 - \beta/\Delta t$. On the other hand, if $\mathcal{G}$ is not 3-colorable, every solution to the dynamic spectrum allocation problem assigns a time-spectrum block of bandwidth at most 1/4 to at least one link at every time. Hence, for at least one node, $T_{ij}/D_{ij} > 1 - \beta/\Delta t$. Unless $P = NP$, no efficient algorithm can distinguish these two cases, which concludes the proof. □

The following theorem can be proven analogously.

Theorem 2.2. *The total throughput maximization problem is NP-complete even if $|\mathcal{P}| = 0$.*

models allow for vastly different communication patterns [20] if transmission powers very between nodes. In case of uniform transmission powers, however, the two models exhibit similar characteristics. Studying the spectrum allocation problem with varying transmission powers significantly adds to its complexity and is an interesting avenue for future research.

Finally, the problem becomes NP-hard even in single-hop networks if there are forbidden spectrum bands.

THEOREM 2.3. *For any $|\mathcal{P}| > 0$, the proportionally-fair throughput maximization problem is NP-complete even in a single-hop environment.*

PROOF. This proof follows by reduction to the PARTITION problem. Given a set A of numbers a_i, the question is whether A can be partitioned into two sets A_1, A_2 in such a way that $\sum_{i \in A_1} a_i = \sum_{i \in A_2} a_i = \frac{1}{2}\sum_{i \in A} a_i$. Given an instance of the PARTITION problem, define a single-hop instance of the dynamic spectrum allocation problem such that each value a_i corresponds to two nodes w_i, w'_i with demand $D'_{ii} = a_i \cdot (f_{top} - f_{bot} + x)/\sum_{i \in A} a_i$. Additionally, let $\mathcal{P} = P$ with $P = (f_{top} - f_{bot} - x)/2, (f_{top} - f_{bot} + x)/2$. In this setting, it is easy to see that an optimally porportionally fair allocation is possible if and only if the set of demands can be divided into two equal partitions. □

3. CENTRALIZED ALLOCATION SCHEME

In order to shed light into the fundamental nature of the problem, we first study a simple centralized algorithm whose performance with regard to (short-term) proportionally-fair throughput is provably good even in worst-case networks. Based on our studies, we derive three desirable properties of a practical, distributed solution, which we investigate in subsequent sections.

The centralized algorithm assume that each node only has one outgoing demand and that $|\mathcal{P}| = 0$. Without loss of generality, we also assume that $f_{bot} = 0$. For notation, let Δ_{min} be the minimum duration of any demand and let $\chi = \Delta_{min}/\beta$ be the ratio between the minimum demand's duration and the switching overhead time. In practice, χ is a large constant. Let $D_{ij}(t)$ denote the current demand of link (v_i, v_j) at time t, and let $S_{ij}(t)$ be the sum of demands of all links in E_{ij} at time t. For a link (v_i, v_j), we denote by E_{ij} the set of links that cannot be scheduled together with (v_i, v_j) at the same time using the same frequency band. According to our definition, a solution which assigns to each node blocks of bandwidth $D_{ij}(t)(f_{top} - f_{bot})/S_{ij}(t)$ is proportionally-fair at time t.

In a nutshell, Algorithm 1 works as follows. Periodically, after a time-interval of size $\Gamma = \chi\beta/k$, it attempts to readjust the current spectrum assignment and assigns new time-spectrum blocks to each active link. The idea is that, on the one hand, the algorithm should adjust quickly enough to response in the demand variation, but on the other hand, too frequent reallocation of time-spectrum blocks is inefficient due to the overhead time β. The definition of Γ ensures a good balance between these two contradictory aims.

Within a single time-interval, the algorithm tries to maximize proportionally-fair spectrum usage by greedily assigning frequency intervals to nodes with *active demands*. A demand is called *active* if its duration spans the entire time-interval $[t_{cur}, t_{cur} + \Gamma]$. Particularly, demands are rounded to the next higher power of 2. In non-increasing order of this demand-size, frequency-intervals I_{ij} are then allocated to links with active demands in a simple greedy fashion. The underlying reason for thus rounding demands in Line 4 is that, in combination with the greedy allocation of bandwidths, this *avoids fragmentation* of the spectrum (cf the proof of Lemma 3.2) even in multi-hop scenarios. This guaranteed absence of unwanted fragmentation in the greedy-allocation phase is the key design principle of the algorithm.

Algorithm 1 Centralized Spectrum Allocation Algorithm

1: Define the constants $\Gamma = \chi\beta/k$ for some $3 \leq k < \chi$.
2: Schedule at time t_{cur}:
3: Let $A_{cur} = \{D_{ij}(t, \Delta t) \mid t + \Delta t \geq t_{cur} + \Gamma\}$ be the set of *active demands*.
4: Let $D'_{ij} = \min\{2^i, i \in \mathbb{Z} | 2^i \geq D_{ij}(t, \Delta t)\}$ for each demand in A_{cur}.
5: **for each** $D_{ij}(t,\Delta t) \in A_{cur}$ in nonincreasing order of D'_{ij}:
6: Let $\mathcal{I}_{cur}$ be the set of intervals already assigned at time t_{cur}. For $D_{ij}(t, \Delta t)$, assign a frequency interval $I_{ij} = [\ell_{ij}, \ell_{ij} + D'_{ij}]$ such that it does not overlap with any previously assigned interval in $\mathcal{I}_{cur}$. By definition of D'_{ij}, such an interval always exists and can be found easily.
7: **end for**
8: Let Φ_{max} be the highest upper boundary $\ell_{ij} + D'_{ij}$ of any interval I_{ij} assigned to any active demand in A_{cur}.
9: Set $\varphi = (f_{top} - f_{bot})/\Phi_{max}$ and assign to each link (v_i, v_j) with active demand the time-spectrum block $B^{cur}_{ij} = (t_{cur}, \Gamma, \varphi\ell_{ij}, \varphi D'_{ij})$.
10: $t_{cur} = t_{cur} + \Gamma$.

The initial allocation typically leads to an infeasible solution, and in order to fix this, all assigned frequency intervals are linearly scaled down by a factor of $\varphi = (f_{top} - f_{bot})/\Phi_{max}$, where Φ_{max} is a scaling factor that ensures feasibility. We now show that our algorithm is within a constant ratio of the optimal solution with regard to $T_{minfair}(\Delta)$ for any value between $3\beta \leq \Delta \leq \chi\beta$. Due to lack of space, some of the proofs are presented in a condensed form.

LEMMA 3.1. *All assigned time-spectrum blocks in the resulting schedule are mutually non-interfering.*

PROOF. Blocks allocated in different time-intervals $[t, t+\Gamma]$ clearly do not interfere. Within a given $[t, t+\Gamma]$, Line 6 guarantees that the intervals I_{ij} are non-interfering. As all links use the same scaling factor φ, the lemma follows. □

The next lemma states that in each time-interval $[t, t+\Gamma]$, every active demand receives a time-spectrum block whose bandwidth is close to the optimal proportionally-fair one.

LEMMA 3.2. *Let κ be a constant. Let $OPT_{ij}(t_{cur}, \Gamma)$ denote the (minimal) throughput of (v_i, v_j) in $[t_{cur}, t_{cur} + \Gamma]$ in an optimally proportionally-fair solution. For every link (v_i, v_j) with active demand $D_{ij}(t, \Delta t) \in A_{cur}$, it holds that*

$$T_{ij}(t_{cur}, \Gamma) \geq \frac{\kappa}{2} \cdot OPT_{ij}(t_{cur}, \Gamma) \cdot \frac{\Gamma - \beta}{\Gamma}.$$

PROOF. Each time a node is assigned a new time-spectrum block, it incurs an overhead of time β. It follows that for an active node,

$$T_{ij}(t_{cur}, \Gamma) \geq \varphi D'_{ij}(\Gamma - \beta) = \frac{D'_{ij}(f_{top} - f_{bot})}{\Phi_{max}}(\Gamma - \beta). \quad (4)$$

A key ingredient for the proof is now that

$$\Phi_{max} \leq 2S_{gh}(t) \quad (5)$$

for the neighborhood E_{gh} with maximal $S_{gh}(t)$. This inequality is true because in Line 4, every demand is increased by at most a factor of 2 and—crucially—there is *no fragmentation*: Because demands are all powers of 2, and time-spectrum blocks are assigned greedily in non-increasing order, it holds that when a demand D'_{ij} is assigned, every available non-overlapping free space has size at least D'_{ij}. That is, in the neighborhood of link (v_g, v_h) with maximal $S_{gh}(t)$, the only reason why Φ_{max} may be larger than $S_{gh}(t)$ is the rounding.

Furthermore, it holds that in each time-window $[t_{cur}, t_{cur}+\Gamma]$ only active demands are considered, i.e., demands that span the entire interval. By standard geometric arguments that hold in the protocol model as long as $R_i^{int}/R_i^t = O(1)$ [19], we obtain for some constant $\kappa \in O(1)$ that

$$OPT_{ij}(t_{cur}, \Gamma) \;\leq\; \frac{1}{\kappa} \cdot \Gamma \cdot \frac{f_{top} - f_{bot}}{S_{gh}(t)}. \qquad (6)$$

Plugging (5) and (6) into inequality (4) concludes the proof. □

Finally, we are ready to derive the theorem. The *volatility-ratio* Ψ denotes the largest possible ratio between minimum and maximum $S_{ij}(t)$ of any link $(v_i, v_j) \in E$ during a time interval of duration β.

THEOREM 3.3. *In every network and for every $3 \leq k < \chi$, Algorithm 1 is within a factor of $O\left(\Psi \cdot \frac{\chi}{\chi-k}\right)$ of the optimal solution with regard to $T_{minfair}(\Delta)$, where $\Delta = 3\chi\beta/k$.*

PROOF. By the definition of $\chi < \Delta_{min}/\beta$ and $\Gamma = \chi\beta/k$, we know that every demand $d_{ij}(t, \Delta t)$ appears in at least three consecutive time-windows $[t_{cur}, t_{cur}+\Gamma]$, at least one of which it spans entirely. In this time-interval, $D_{ij}(t, \Delta t)$ becomes *active* and therefore, Lemma 3.2 implies that the throughput $T_{ij}(t_{cur}, \Gamma)$ is close to the optimum. The first and last time-window $D_{ij}(t, \Delta t)$ appears in, it is not scheduled by the algorithm, but an optimal schedule may have assigned a time-spectrum block to this link. By the definition of volatility, we know that $OPT_{ij}(t_{cur}, 2\Gamma) \leq (\Psi + 1)OPT_{ij}(t_{cur}, \Gamma)$ and, similarly, $OPT_{ij}(t_{cur} - \Gamma, 2\Gamma) \leq (\Psi + 1)OPT_{ij}(t_{cur}, \Gamma)$. Because this holds for every t_{cur}, it follows from Lemma 3.2,

$$T_{ij}(t_{cur}, 3\Gamma) \geq \frac{\kappa}{2(\Psi+1)} \cdot OPT_{ij}(t_{cur}, 3\Gamma) \cdot \left(1 - \frac{\beta}{\Gamma}\right).$$

The proof of the theorem is now concluded by deriving the ratio $OPT_{ij}(t_{cur}, 3\Gamma)/T_{ij}(t_{cur}, 3\Gamma)$ and replacing β/Γ by k/χ. □

Notice that the bound in Theorem 3.3 gets tighter as k increases. That is, the smaller the fairness interval, the looser the proportionally-fair throughput guarantee. Setting $k = \chi/2$, for instance, we obtain an $O(\Psi)$ approximation of the optimum for an interval as small as 6β.

Summary: The centralized algorithm discussed in this section does not lend itself for practical application in real multi-hop cognitive radio networks. The purpose of our studies has been to gain a deeper understanding of the inherent algorithmic challenges that a practical protocol for the dynamic spectrum allocation problem faces, and consequently, what desirable properties such a protocol should have.

Opportunistic usage: Spectrum allocation should divide the overall bandwidth B of white spaces to accommodate the contending links by tuning the operating bandwidth. In the centralized solution, the bandwidth assigned to the transmission is $D_{ij}B/S_{ij}$, which allows the spectrum to be adaptively bundled together to deliver high throughput for heavy traffic from few users, or be opportunistically fragmented and shared among a large number of contending devices.

Fine-timescale control: Links should share the spectrum in fine timescale in order to adapt to instantaneous traffic demand and control latency. $\Gamma = \chi\beta/k$ is defined to achieve a balance between the agility of solutions and the time overhead associated with each allocation.

Non-interfering allocation: All time-spectrum blocks are mutually non-interfering. The exclusive access to a time-spectrum block largely reduces time overhead of contentions in the given band, mitigates the hidden terminal problem after frequency switching [15], and encourages packet aggregation for high efficiency.

4. B-SMART: A DISTRIBUTED SCHEME

We now present b-SMART, a distributed and practical spectrum allocation scheme. In order to realize dynamic and fine-timescale allocation, nodes running the b-SMART protocol maintain the instantaneous spectrum usages of all their neighbors. The sender and the intended receiver coordinate with each other to reserve a time-spectrum block that is available at both nodes. The size of the block is determined using a local algorithm that is executed at the sender and the receiver. First, we briefly describe the MAC protocol, called CMAC [30], that we use to support the reservation of a time-spectrum block. We then describe our spectrum allocation algorithm that is run locally at each node.

4.1 CMAC: A Cognitive Radio MAC

CMAC works as follows. Each node that has packets to send contends for spectrum access on the control channel. It uses a *three-way handshake* with the destination node to decide on the spectrum block to use for the transmission, and reserve the time-spectrum block in the neighborhood. Each neighboring node equipped with an extra receiver overhears the handshake process and stores this information in a local table. When the time arrives, both the sender and the receiver configure their reconfigurable transceivers to switch to the selected frequency band and exchange data packets in the time-spectrum block. The nodes switch back to the control channel after the reserved block is consumed.

CMAC's three-way handshake is substantially different from the RTS (request-to-send) and CTS (clear-to-send) handshake used in 802.11. Although we reuse—for the ease of exposition—some of the 802.11 terminology, all our control packets have different structure and functions. A sender initiates the handshake using an RTS packet. This packet has information about its traffic load and suggestions for time-spectrum blocks, calculated using b-SMART . The traffic load is described using the queue length and the average packet size. Each suggested time-spectrum block is indicated using a quadruple $(f, \Delta f, t, \Delta t)$, where f is the offset from 470 MHz, the lowest frequency in the TV bands.

On receiving an RTS packet, the destination node verifies the time-spectrum blocks proposed by the sender and selects one that is also available locally. It informs the sender about its choice using a CTS packet. On receiving the CTS packet, the sender announces the reserved time-spectrum block to its neighbors using a DTS (Data Transmission reServation) packet. Each node uses the overheard CTS and DTS packets to build a two-dimensional (time and frequency) local view of the current spectrum allocations and reservations. To simplify control and avoid starvation, we limit each node to have at most one valid reservation at any time, i.e., it can contend for a new time-spectrum block only after completion of the previous transmission.

On the arrival of a block's start time, the sender and receiver tune their transceivers to the reserved band $[f, f + \Delta f]$. The transmission power is adjusted to $\Delta f \cdot P_c / b_c$ to maintain approximately the same power spectrum density. b_c and P_c denote the control channel bandwidth and the power that is used to send the control packets. Before transmitting, nodes first perform physical-layer carrier sensing in order to avoid interference with primary users in the TV-bands. If the selected spectrum is clear, nodes exchange packets without further back-off and the data rate is decided by applying a standard rate control algorithm. In the unlikely case that the selected spectrum is busy (for example, due to loss on the control channel, deep fading, or interference from adjacent frequency bands), the current time-spectrum block is not used and the sender resumes contention on the control channel.

Enhancements to CMAC: In order to avoid the control channel from becoming a bottleneck and to ensure fairness among contending nodes, it is crucial to appropriately adjust the random back-off mechanism. In particular, it is well-known that even in the basic 802.11 protocol and even for a single communication channel, the problem of setting the back-off window to its minimum size upon completion of a transmission causes unfairness. [10]. A node that has just finished transmitting has a higher chance of recapturing the channel than other nodes, because its back-off window is smaller.

Whereas this fundamental problem has a relatively minor impact in single-channel scenarios, it becomes much more dramatic in multi-channel settings or cognitive radio networks. In such scenarios, a large number of nodes can simultaneously transmit data on a band other than the control channel and hence, the rate of nodes returning to the control channel is significantly higher than in single-channel environments. If each of these returning nodes sets its contention window to the minimum size, other nodes have virtually no chance of gaining access to the channel and starve.

In order to alleviate this problem, our protocol implements the following enhancement to the 802.11-based back-off protocol: *When returning to the control channel upon completion of a transmission, a node chooses the smallest possible back-off window size larger than N, i.e., the smallest window for which its transmission probability is smaller than $1/N$.* This improvement of the protocol prevents an over-increase of the sum of transmission probabilities of all nodes, which, in turn, guarantees that the *control channel does not become a bottleneck* even when the number of nodes n becomes very large and significantly exceeds the maximum number of available channels. This is also backed by our analysis in Section 5.

4.2 Dynamic Spectrum Allocation Algorithm

The MAC scheme discussed previously only regulates which sender-receiver pair may reserve some time-spectrum block at a specific time. At the heart of our protocol is the dynamic spectrum allocation algorithm that decides on the four variables t, Δt, f, and Δf to shape the block. The algorithm is invoked when a sender sends a RTS packet to an eligible receiver. The sender considers the overall bandwidth of the white spaces $B = f_{top} - f_{min} - |\mathcal{P}|$, the local spectrum allocation table, and the corresponding queue size $qLen$, to decide on a time-spectrum block.

Algorithm 2 Distributed Spectrum Allocation Algorithm

Let the available bandwidth options be: $\{b_1, b_2, ..., b_n\}$, with $b_1 < b_2 < \ldots < b_n$.
$I := \min\{i | b_i \geq B/N\}$;
for $i = I, \ldots, 1$ **do**
  $\Delta f = b_i; \Delta t := \min\{T_{max}, \text{Tx\_Duration}(b_i, qLen)\}$
  if $\Delta t == T_{max}$ or $i == 1$ **then**
    Find the best placement of the $\Delta f \times \Delta t$ block in the local spectrum allocation table that minimizes the finishing time and is compatible with the existing allocations and prohibited bands.
    if the block can be placed in the local spectrum allocation table **then**
      return the allocation $(t, \Delta t, f, \Delta f)$.
    end if
  end if
end for

This dynamic decision is guided by the following principles, which are derived in Section 3. First the bandwidth Δf should be determined based on the current demand; it must be large enough to achieve a high data-rate, but it should not be too large considering the fragmentation in the white spaces and the fairness expectation. Hence, b-SMART attempts to assign to each sender-receiver pair (v_i, v_j) a time-spectrum blocks with bandwidth B/N, where we define N is the number of current disjoint transmissions in the interference range of (v_i, v_j). Two transmissions are considered disjoint if they do not share either endpoint. This definition of N aims at achieving *per-node fairness* and is particularly appealing because it can easily be implemented in the distributed setting. The second design trade-off involved is the block duration Δt. While using a shorter block reduces delay and improves connectivity, it results in higher contention on the control channel. Our approach sets an upper bound T_{max} on the maximum block-duration and nodes always try to send for duration T_{max}. As we motivate later, our choice of T_{max} amortizes the incurred overhead in the control channel, thereby preventing the control channel from becoming a bottleneck.

The details of this algorithm are presented in Algorithm 2. The algorithm evaluates the available bandwidth options in decreasing order, starting with the bandwidth option just exceeding B/N. For each bandwidth option $\Delta f = b_i$ in consideration, the algorithm estimates its corresponding transmission time Δt based on the current queue size (i.e., how long it uses the specific bandwidth to empty the queue). If the resulting transmission time exceeds T_{max}, then Δt is set to T_{max}. Given Δt and Δf, we optimize the placement of the time-spectrum block in the local allocation table. This is done by minimizing the finishing time while not overlapping with the existing allocations and prohibited bands. If the time-spectrum block of size $\Delta t \times \Delta f$ cannot be placed due to the existence of prohibited bands, then the next smaller bandwidth option is considered.

We next discuss our choice for the two important parameters, T_{max} and N.

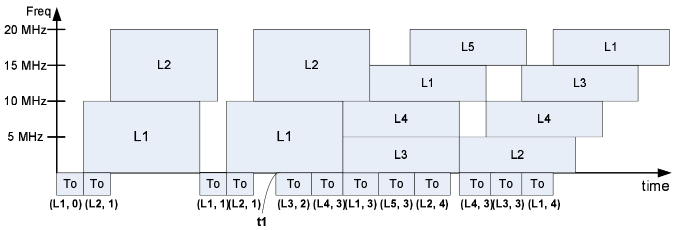

Figure 1: Illustration of the spectrum allocation of 20MHz white spaces. Bandwidth options are 10 and 5 MHz. At time t_1, Links $L3, L4, L5$ join $L1$ and $L2$. All links are disjoint, blacklogged and within each other's transmission range. The bracket denotes the link ID and the number of valid reserved blocks N, at the time of the handshake.

Setting T_{max}: Let $\Lambda \triangleq B/b_{min}$ denote the maximum number of parallel transmissions that the white spaces of bandwidth B can accommodate, where b_{min} is the smallest bandwidth option. In b-SMART , T_{max} is set to satisfy the following condition:

$$T_{max} = \Lambda \cdot T_o = B \cdot T_o / b_{min}, \tag{7}$$

where T_o denotes the average time spent on one successful handshake in the control channel. Before validating our choice by means of analysis and simulation in Sections 5, we briefly explain the main intuition behind the formula. In order to keep the white spaces fully utilized, we need to prevent the control channel from becoming the bottleneck. Therefore, it is important to ensure that the rate at which handshakes are generated, R_l, is not less than the rate at which nodes return to the control channel, R_r, i.e., $R_l \geq R_r$. The handshake is generated at the rate of $1/T_o$, thus $R_l = 1/T_o$. Since the maximal number of parallel transmissions in the spectrum is Λ and each regular transmission lasts for T_{max}, the maximal returning rate is Λ/T_{max}. The definition of T_{max} now follows from the fact that in a fully utilized spectrum, we want Λ/T_{max} to exceed $1/T_o$. The empirical formula indicates that by increasing T_{max} or reducing the handshake overhead, more parallel transmissions can be supported by the control channel.

Estimating N: The algorithm chooses the bandwidth Δf for a link L based on N, the instantaneous number of disjoint transmissions in the interference range of link L. In the distributed scenario, however, it is almost impossible to get a perfect estimation of N considering such a process is repeated in a fine timescale. We therefore approximate N using the number of valid time-spectrum blocks stored in the local table. At time t_{cur}, a block is valid if $t + \Delta t > t_{cur}$. Figure 1 depicts an example of this online approximation of N. Initially, $L1$ and $L2$ each get a half of the spectrum, because the number of pending blocks is 1, hence N is 2 and Δf for each link is 10MHz. As three more links join the network, the number of valid blocks increases. Accordingly the protocol forces each link to reduce its bandwidth share Δf to 5 MHz in order to increase parallelism. In the initial stage when many new links join the network, it takes a time period to collect relevant reservations and learn the existence of the local contending transmissions. As we show in the simulation section, this learning period is short for various traffic types and even large number of new nodes.

Notice that since N is derived based on the up-to-date reservations, b-SMART is responsive to user and traffic dynamics. The number of disjoint transmissions is effectively tracked especially when flows are long-lived and backlogged with packets. In the unsaturated case, we find that the method is conservative since B/N constrains the upper limit of the bandwidth. Hence, in such cases, more complex greedy strategies could potentially achieve better performance. Deriving different strategies for deciding the time-spectrum blocks is interesting future work.

Receiver Scheduling: To meet the T_{max} duration requirement, our solution incorporates a packet aggregation procedure and implements a round-robin scheduler to handle the packet queues in a node. A node periodically examines the output packet queues for its neighbors. A neighbor becomes *eligible* if (i) it does not have an outstanding reservation, and (ii) the output packet queue for this neighbor has accumulated enough packets to satisfy the T_{max} requirement, or the queue has timed out for packet aggregation.

5. PERFORMANCE EVALUATION

In this section, we first establish a theoretical model that describes the throughput achieved by b-SMART in fully connected topologies. We also validate the choice of T_{max} in (7) and motivate our selection of control channel bandwidth. We then present simulation results to show that b-SMART performs well under most scenarios.

5.1 Theoretical Analysis

We first analyze the throughput achieved by b-SMART. In line with the rich literature on Markov-based performance modeling and analysis of randomized back-off protocols initiated by Bianchi [7], we focus on the *saturation throughput.* We assume N disjoint flows in a single-hop network in which each sender has backlogged queues with packets of equal size. The minimum bandwidth option is b_{min} and that the white space of bandwidth B can accommodate at most Λ parallel transmissions. To show that even for large number of stations ($N > \Lambda$), the control channel can be prevented from becoming a bottleneck by setting T_{max} according to Inequality (7), we model the control channel using a Markov model, and verify the model using simulations.

At any given time, let Q denote the number of stations that contend on the control channel. Conversely, the number of stations that are currently transmitting in the white spaces is $N - Q$. In order to make the Markov system analyzable, consider the following simplification. In the saturated case, every time-spectrum block used by a station is of duration $\Delta t = T_{max}$. Because nodes obtain such a block only after a successful handshake on the control channel, the time between two consecutive nodes return to the control channel is at least T_o, the average time overhead spent on one successful handshake. Instead of taking into account these complex timing-dependencies between different nodes sending in white spaces, we assume that in any (small enough) time-interval of duration t, the probability that a transmitting node returns to the control channel is $\lambda_{ret}(Q) = (N - Q) \cdot t / T_{max}$. On the other hand, the probability that in a time-interval t, there is at least one successful handshake on the control channel can be approximated as

$$\mu_{suc}(Q) = \frac{p_{suc}(Q) \cdot t}{p_{suc}(Q)t_{suc} + p_{col}(Q)t_{col} + p_{idle}(Q)t_{idle}}.$$

In this formula, the probabilities $p_{suc}(Q), p_{col}(Q), p_{idle}(Q)$ denote the respective likelihood of a successful handshake on the control channel, a collision, or an idle time-slot, given that Q nodes currently contend on the control channel. Further, $t_{suc}, t_{col}, t_{idle}$ denote the respective duration of such a time-slot. Specifically, it follows from our discussion in Section 4.1 that in b-SMART , t_{idle} equals the empty slot-time σ and

$$\begin{aligned} t_{col} &= t_{RTS} + t_{DIFS} + \sigma \\ t_{suc} &= t_{col} + t_{CTS} + t_{DTS} + 2t_{SIFS} + 2\sigma. \end{aligned}$$

The probabilities $\lambda_{ret}(Q)$ and $\mu_{suc}(Q)$ can now be interpreted as transition probabilities in a one-dimensional birth-death Markov process $\{s(t)\}$ with $\Lambda+1$ states $C_{N-\Lambda}, \ldots, C_N$. In this process, state C_i signifies a state in which i nodes are contending on the control channel and consequently, $\Lambda - i$ nodes are currently in the midst of a transmission. Let $\mathbf{q} = (q_{N-\Lambda}, \ldots, q_N)$ be the stationary distribution of the chain. It is easy to show by induction that for all $N - \Lambda + 1 \leq i \leq N$,

$$q_i = q_{N-\Lambda} \cdot \frac{\prod_{j=N-\Lambda}^{N-\Lambda+i-1} \lambda_{ret}(j)}{\prod_{j=N-\Lambda+1}^{N-\Lambda+i} \mu_{suc}(j)}.$$

As the t cancel out in the above formulas and because of the additional condition $\sum_{i=N-\Lambda}^{N} q_i = 1$, the only remaining unknowns are the relative success, collision, and idle probabilities $p_{suc}(j), p_{col}(j), p_{idle}(j)$ for all values of $N - \Lambda \leq j \leq N$. For each such j, we approximate these probabilities in the steady state using the respective probabilities in a regular 802.11 DCF system consisting of j nodes as derived by Bianchi in [7]. Notice that this simplification is justified due to our improvement of the back-off protocol discussed in Section 4.1. In the improved protocol version, the system state upon the return of transmitting stations closely resembles a steady state with C stations immediately. Hence, if W and m denote the minimal size of the contention window and the maximum number of backoff stages, respectively, then

$$p_{suc}(j) = \frac{j\tau(1-\tau)^{j-1}}{1-\theta(j)} \quad \text{and} \quad p_{col}(j) = 1 - \theta(j)(1 - p_{suc}(j)),$$

where $\theta(j) = (1-\tau)^j$ and τ follows from the unique solution to the nonlinear system of equations in the two unknowns τ and p defined by $\tau = \frac{2(1-2p)}{(1-2p)(W+1)+pW(1-(2p)^m)}$ and $p = 1 - (1-\tau)^{j-1}$. Finally, every time slot is either a success, a collision, or idle and hence, $p_{idle}(j) = 1 - p_{suc}(j) - p_{col}(j)$. This completely specifies the Markov process.

Because the throughput achieved in state C_i is proportional to $(N - i) \cdot b_{min}$, the saturated throughput H of b-SMART can be described by the expression

$$H = b_{min} \cdot \frac{T_{packet}}{T_{packet} + T_{ACK}} \cdot \sum_{j=N-\Lambda}^{N} (N - j) \cdot q_j.$$

In view of this formula, it follows that, ideally, the process should always be in state $C_{N-\Lambda}$ or at least remain in states C_i with low i for most of the time. Figure 2 plots the saturation throughput predicted by this model and compares it with our simulation results. The figure shows that the model closely matches our empirical findings.

Our model allows us to study the impact of T_{max} and control channel bandwidth on network throughput. Figure 2 shows the throughput performance as Λ increases with different T_{max} settings. In particular, the plot shows a setting with $N = \Lambda + 1$; and other parameters are listed in Section 5.2.

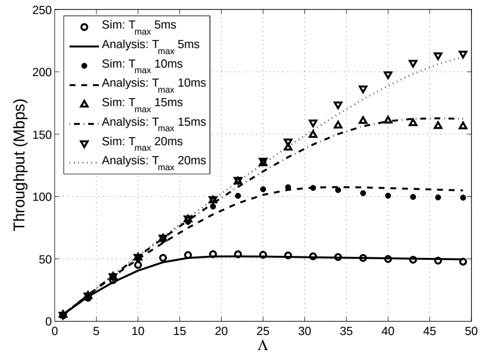

Figure 2: Network throughput v.s Λ

As pointed out, our analytical result matches well with the simulations. The reason why the choice of T_{max} is crucial is because it determines the number of parallel transmissions Λ, in line with the bandwidth of white spaces B. When T_{max} is sufficiently large, b-SMART's throughput increases proportionally with B. In this specific example, $T_{max} = 20ms$ can support up to 50 parallel transmissions or 250 MHz white spaces. On the other hand, with smaller settings of T_{max}, the network throughput stops from increasing as Λ grows over a certain point. At that specific point, the value of T_{max} is the minimal setting of T_{max} to support the corresponding number of Λ. Further extensive simulation results give strong validation of this relationship between T_{max} and Λ and, consequently, the definition of formula (7).

The choice of 5 MHz control channel bandwidth tries to balance the following two contradicting aims. The first is to minimize the spectrum resource consumed by the control channel. The other goal is to minimize the average handshake overhead, T_o, to control T_{max} within the fine timescale, say tenths of a millisecond. In this sense, the control channel bandwidth determines T_o and hence, contributes to the length of T_{max}. Our choice of 5MHz control channel bandwidth to regulate the white spaces in the TV bands has been guided by thorough analysis and simulation study.

5.2 Simulation Results

We have implemented b-SMART in QualNet [4]. In previous work [30], we studied the effectiveness of adaptively allocating bandwidth to nodes, and showed that a basic version of b-SMART outperforms any scheme that allocates fixed bandwidth to all nodes. In this section, we evalaute b-SMART in more detail, and show that it can efficiently handle a variety of scenarios, including fragmented white spaces, varying traffic patterns, different packet sizes and varying node density.

Our experimental setup is as follows. We set the total bandwidth of all available white spaces to be 80 MHz. We allow nodes to only use discrete bandwidth values of 5, 10, 20 and 40 MHz, and assume that every 1 MHz of spectrum delivers 1.2 Mbps [14]. We simulate the above bandwidths using data rates of 6, 12, 24 and 48 Mbps respectively, and by disabling auto-rate control in our simulations. We set the control channel at the lowest data rate of 6 Mbps (5 MHz of spectrum) and T_{max} to 5 ms. For simplicity, we use the IEEE 802.11a OFDM waveform, with the same physical layer parameters [2].

In our experiments, we use TCP and UDP flows among

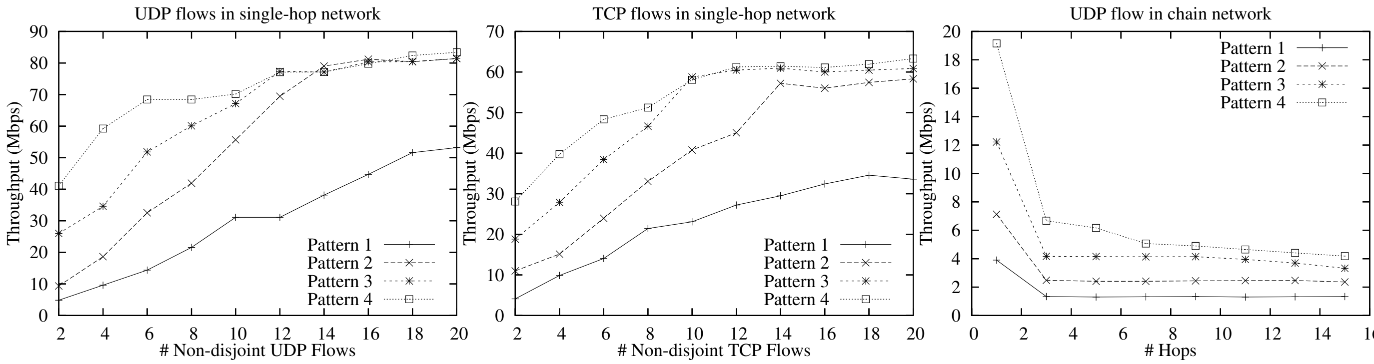

Figure 3: Network throughput achieved by b-SMART in different fragmentation patterns

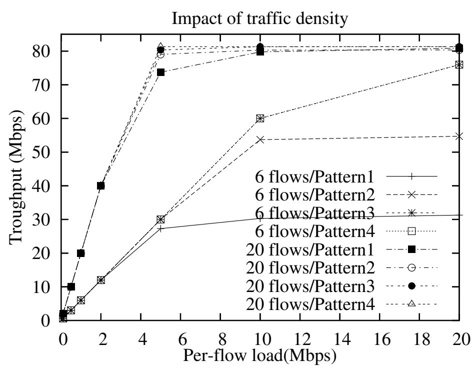

Figure 4: b-SMART adapts to varying traffic patterns to use all the available bandwidth

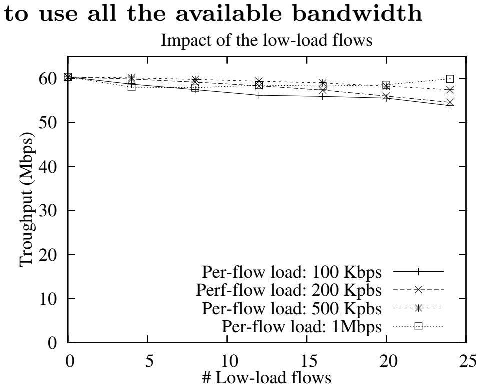

Figure 5: The system throughput drops slightly in the presence of non-backlogged flows

nodes. Unless otherwise noted, we set all flows to be backlogged, and use 1500 byte packets.

5.2.1 *Impact of Fragmentation*

We first analyze the performance of b-SMART when the white spaces are fragmented and non-contiguous. This can occur when portions of the spectrum are occupied by licensed users. We study 4 simple fragmentation patterns, namely 1, 2, 3, 4, which correspond to 16 fragments of 5 MHz, 8 fragments of 10 MHz, 4 fragments of 20 MHz and 2 fragments of 40 MHz respectively. We evaluate the performance of b-SMART for all 4 patterns under 3 different scenarios: single hop TCP and UDP flows on changing the number of non-disjoint flows (flows many share an endpoint) in the network, and a chain network on increasing the number of hops in the chain. We present the results in Figure 3.

In single hop networks, the total throughput of the system increases with an increase in the number of flows in the network. This is because b-SMART attempts to maximize the use of available spectrum. Hence, it is able to sustain a larger number of parallel flows when there are more flows in the network. The increase in throughput is more gradual after a certain number of flows, since b-SMART is able to nearly utilize all the available bandwidth. The number of flows when the increase becomes gradual depends on the particular pattern. For example, in the 5 MHz case (Pattern 1), the network throughput steadily increases with an increase in the number of flows. However, the total throughput in this case is the least among all patterns as there is always chunks of bandwidth that b-SMART is unable to use when there are only 20 non-disjoint flows. In the case of a chain network, the total system throughput achieved by b-SMART stabilizes after 2 hops. This is because b-SMART can simultaneously schedule two non-consecutive nodes in the chain to use two non-overlapping parts of the spectrum.

5.2.2 *Adapting to Traffic Patterns*

We now show how b-SMART adapts to different traffic patterns to utilize the maximum available spectrum. To study this effect, we change the number of flows (6 and 20) and the amount of traffic generated by them (from 100 Kbps to 20 Mbps), and plot the total system throughput in Figure 4. We present the throughput numbers for fragmentation patterns presented in Section 5.2.1. When there are 20 flows in the network, b-SMART is able to use all the available spectrum when each flow generates more than 5 Mbps of traffic. Since the minimum bandwidth in our experiments is 5 MHz, the minimum amount of traffic per-flow required to saturate this spectrum chunk is 6 Mbps. However, when there are only 6 flows in the network, a demand of 5 Mbps is not enough to saturate all the available bandwidth. Furthermore, the system throughput for all fragmentation patterns is the same when each flow generates less than 5 Mbps of traffic. When each flow generates more than 5 Mbps of traffic, b-SMART adaptively allocates more bandwidth to each flow, and hence fragmentation patterns with larger white space fragments achieve higher throughput.

5.2.3 *Impact of Low Load Flows*

We now study the impact of non-backlogged flows on the performance of b-SMART . In Figure 5, we plot the system throughput when there are 2 backlogged flows coexisting with a number of non-backlogged low-load flows that generate smaller (512 byte) packets at a lower rate (100 Kbps to 1 Mbps). As we increase the number of low-load flows, the system throughput degrades by up to 12%. This is because the overhead of control channel contention is not sufficiently amortized when transmitting a smaller data packet.

6. RELATED WORK

Approaches to regulate spectrum allocation can broadly be classified into two strategies: centralized and distributed schemes. IEEE 802.22 [3], DSAP [27] and DIMSUMnet [8] adopt a centralized scheme, where a central controller allocates spectrum to all nodes. In contrast, b-SMART uses a distributed approach that enables each cognitive radio to make real-time decision on spectrum usage based on locally collected information. Within the distributed category, most prior solutions are based on channels of fixed bandwidth. In the context of 802.11, the proposed multichannel MAC schemes, for example SSCH [21], MMAC [15], DCA [29], xRDT [24], HMCP [22], assume a set of channels with preallocated fixed bandwidths. There has been prior work on MAC protocols for cognitive radio networks [16, 23, 18]. These protocols can be used by two cognitive radio nodes to coordinate on a free channel of *fixed* bandwidth to communicate with each other. b-SMART takes these approaches one step further to enable nodes to coordinate and communicate on channels of *varying* bandwidth.

In the more theoretical space, a vast amount of scheduling and frequency allocation problems have been studied for both stationary and ad hoc networks. These problems, which often provide the theoretical foundation for practical MAC-layer protocols, have typically been modeled by mapping them to variants of coloring and matching problems, e.g. [25, 17, 6, 28]. More recently, researchers have derived more realistic, non-graph-based interference models for wireless communication, e.g. [9, 26, 13, 20]. Although often requiring novel solution techniques, the problem space and hence the underlying algorithmic challenges in these models remain the similar.

Our work stands apart from this prior work in that it adds a new dimension to the problem itself: It combines not only the question of determining *when* and/or *at what frequency* each node should transmit, but also with how wide a spectrum-band. In combination with the dynamic nature of white spaces, the reconfigurability offered by cognitive radios therefore leads to novel and practically important theoretical and algorithmic questions.

7. CONCLUSIONS

In this paper, we introduced the spectrum allocation problem in cognitive radio networks. We defined a time-spectrum block as a unit of spectrum reservation, and used it to formalize the spectrum allocation problem. In the process, we showed that this problem is radically different from any existing work based on the fixed channelization. We also proposed and evaluated a distributed solution, called b-SMART, which enables nodes to share the white spaces in a fine timescale.

8. REFERENCES

[1] FCC, Unlicensed Operation in the TV Broadcast Bands Notice of Proposed Rulemaking (NPRM), ET Docket No. 04-186, May, 2004.

[2] IEEE 802.11a-1999 IEEE Standard Part 11: wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: high-speed physical layer in the 5 GHz band.

[3] IEEE 802.22 WRAN WG, www.ieee802.org/22/.

[4] QualNet, http://www.qualnet.com/.

[5] Shared Spectrum Company, www.sharedspectrum.com.

[6] Arunesh Mishra and Vivek Shrivastava and Dheeraj Agrawal and Suman Banerjee and Samrat Ganguly. Distributed Channel Management in Uncoordinated Wireless Environments. In *ACM MobiCom 2006*.

[7] G. Bianchi. Performance Analysis of the IEEE 802.11 Distributed Coordination Function. *IEEE JSAC*, 18(3), 2000.

[8] M. M. Buddhikot, P. Kolodzy, S. Miller, K. Ryan, and J. Evans. DIMSUMNet: New directions in wireless networking using coordinated dynamic spectrum access. In *IEEE WoWMoM05, June 2005*.

[9] T. ElBatt and A. Ephremides. Joint Scheduling and Power Control for Wireless Ad Hoc Networks. *IEEE Transactions on Wireless Communications*, 3(1), 2004.

[10] Z. Fang, B. Bensaou, and Y. Wang. Performance Evaluation of a Fair Backoff Algorithm for IEEE 802.11 DFWMAC. In *ACM MobiHoc*, 2002.

[11] A. Graf, M. Stumpf, and G. Weissenfels. On Coloring Unit Disk Graphs. *Algorithmica*, 20, 1998.

[12] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, (2):388–404, 2000.

[13] Gurashish Brar and Douglas Blough and Paolo Santi. Computationally Efficient Scheduling with the Physical Interference Model for Throughput Improvement in Wireless Mesh Networks. In *ACM MobiCom 2006*.

[14] J. Proakis, Digital Communications, McGraw Hill, 2001.

[15] J. So and N.H. Vaidya. Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using a Single Transceiver. In *ACM MobiHoc 2004*.

[16] J. Zhao, H. Zheng, G. Yang. Distributed Coordination in Dynamic Spectrum Allocation Networks. In *Dyspan 2005*.

[17] S. O. Krumke, M. V. Marathe, and S. S. Ravi. Models and Approximation Algorithms for Channel Assignment in Radio Networks, journal = Wireless Networks, volume = 7, number = 6, year = 2001, pages = 575–584,.

[18] L. Ma, X. Han, C. Shen. Dynamic open spectrum sharing MAC protocol for wireless ad hoc networks. In *IEEE Dyspan 2005*.

[19] M. Marathe, H. Breu, H. Hunt III, S. S. Ravi, and D. Rosenkrantz. Simple Heuristics for Unit Disk Graphs. *Networks*, 25, 1995.

[20] T. Moscibroda, R. Wattenhofer, and Y. Weber. Protocol Design Beyond Graph-Based Models. In *ACM HotNets 2006*.

[21] P. Bahl, R. Chandra, and J. Dunagan. SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wirelessnetworks. In *ACM MobiCom 2004*.

[22] P. Kyasanur, J. So, C. Chereddi, and N. H. Vaidya. Multi-Channel Mesh Networks: Challenges and Protocols. In *IEEE Wireless Communications, April 2006*.

[23] Q. Zhao, L. Tong, Swami, A. Decentralized cognitive mac for dynamic spectrum access. In *IEEE Dyspan 2005*.

[24] R. Maheshwari, H. Gupta, S. Das. MAC Protocol for Multiple Channels. In *IEEE SECON 2006*.

[25] S. Ramanathan and E. L. Lloyd. Scheduling Algorithms for Multi-Hop Radio Networks. In *Proc of SIGCOMM*, 1992.

[26] Thomas Moscibroda and Roger Wattenhofer. The Complexity of Connectivity in Wireless Networks. In *IEEE Infocom 2006*.

[27] V. Brik, E. Rozner, S. Banerjee, and P. Bahl. DSAP: A Protocol for Coordinated Spectrum Access. In *IEEE Dyspan 2005*.

[28] W. Wang and Y. Wang and X. Y. Li and W. Song and O. Frieder. Efficient Interference-Aware TDMA Link Scheduling for Static Wireless Networks. In *ACM MobiCom 2006*.

[29] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu. A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Mobile Ad Hoc Networks. In *I-SPAN 2000*.

[30] Y. Yuan, P. Bahl, R. Chandra, P. A. Chou, J. I. Ferrell, T. Moscibroda, S. Narlanka, and Y. Wu. KNOWS: Kognitive Networking Over White Spaces. In *IEEE Dyspan 2007*.