Opportunistic Scheduling with Reliability Guarantees in Cognitive Radio Networks

Rahul Urgaonkar, *Student Member, IEEE,* Michael J. Neely, *Senior Member, IEEE*

Abstract—We develop opportunistic scheduling policies for cognitive radio networks that maximize the throughput utility of the secondary (unlicensed) users subject to maximum collision constraints with the primary (licensed) users. We consider a cognitive network with static primary users and potentially mobile secondary users. We use the technique of Lyapunov Optimization to design an online flow control, scheduling and resource allocation algorithm that meets the desired objectives and provides explicit performance guarantees.

Index Terms—Cognitive Radio, Queueing Analysis, Resource Allocation, Lyapunov Optimization

1 Introduction

Cognitive radio networks have recently emerged as a promising technique to improve the utilization of the existing radio spectrum. The key enabler is the cognitive radio [2] that can dynamically adjust its operating points over a wide range depending on spectrum availability. The main idea behind a cognitive network is for the unlicensed users to exploit the spatially and/or temporally underutilized spectrum by transmitting *opportunistically*. However, a basic requirement is to ensure that the existing licensed users are not adversely affected by such transmissions. Such interference with the licensed users may be unavoidable due to lack of precise channel state information. In this paper, we develop an opportunistic scheduling policy that maximizes the throughput utility of the secondary (or unlicensed) users subject to maximum collision constraints with the primary (or licensed) users in a cognitive radio network. Our scheme is shown to work in the presence of imperfect knowledge about primary user spectrum usage and provides tight reliability guarantees.

A survey on the technology, design issues and recent work in cognitive radio networks is provided in [3], [4]. The problem of optimal spectrum assignment to secondary users in static networks is treated in [5]–[10] where it is assumed that scheduling is aware of primary user transmissions. Scheduling the secondary users under partial channel state information is considered in [11]–[13] which use a probabilistic maximum collision constraint with the primary users.

In this paper, we use the techniques of adaptive queueing and Lyapunov Optimization to design an online flow control, scheduling and resource allocation algorithm for a cognitive network that maximizes the throughput utility of the secondary users subject to a maximum rate of collisions with the primary users. This algorithm operates without knowing the mobility pattern of the secondary users and provides explicit performance bounds. Lyapunov Optimization techniques were perhaps first applied to wireless networks in the landmark paper [14], where Lyapunov drift is used to develop a joint optimal routing and scheduling algorithm. This method has since been extended to treat problems of joint stability and utility optimization in general stochastic networks in [15]–[18] and wireless mesh networks in [19]. The analysis presented in this paper applies the stochastic network optimization framework of [18].

The main contributions of this work are described below. This paper:

- Develops throughput optimal control policies for cognitive networks with general interference and mobility models.
- Introduces the notion of "collision" queues that are used to provide strong *reliability* bounds in terms of the worst case number of collisions suffered by a primary user in any time interval. In particular, the collision queue method here is adapted from the virtual power queue technique of [17]. However, the collision queues developed here are designed to ensure reliability constraints, rather than average power constraints. Different from [17], this requires the inputs to the virtual queues to be random collision variables that can be evaluated only after packet transmission has taken place.
- Develops easier to implement constant factor approximations to the optimal resource allocation problem.

- *This work was presented in part at the IEEE INFOCOM conference, Phoenix, AZ, April 2008 [1].*

- *Rahul Urgaonkar and Michael J. Neely are with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089. web: http://www-scf.usc.edu/∼urgaonka*

- *This material is supported in part by one or more of the following: the DARPA IT-MANET program grant W911NF-07-0028, the NSF grant OCE 0520324, the NSF Career grant CCF-0747525.*

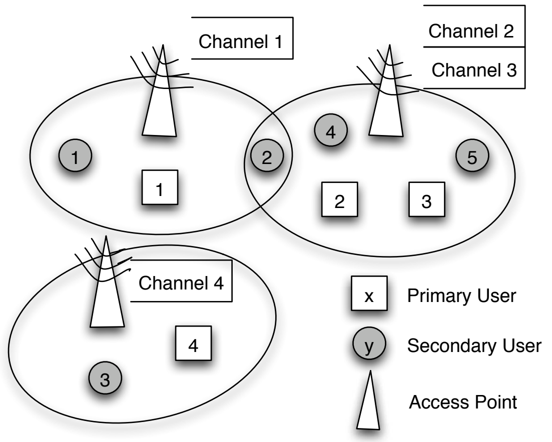


Fig. 1. Example cognitive network showing primary and secondary users

2 Network Model

We consider a cognitive radio network consisting of M primary users and N secondary users as shown in Fig. 1. Each primary user has a unique licensed channel and these are orthogonal in frequency and/or space. Thus, the primary users can send data over their own licensed channels to their respective access points simultaneously. The secondary users do not have any such channels and opportunistically try to send their data to their receivers by utilizing idle primary channels. Such opportunities are called "spectrum holes".

2.1 Mobility Model

We consider a time-slotted model. The primary users are assumed to be static. However, the secondary users could be mobile so that the set of channels they can access can change over time. In a timeslot, a secondary user can access a subset of the primary channels potentially depending on its current location. This information is concisely represented by an $N \times M$ binary channel accessibility matrix $\boldsymbol{H}(t) = \{h_{nm}(t)\}_{N\times M}$ where:

$$h_{nm}(t) = \begin{cases} 1 & \text{if secondary user } n \text{ can access} \\ & \text{channel } m \text{ in slot } t \\ 0 & \text{else} \end{cases}$$

For example, the channel accessibility matrix for the example network in Fig.1 is given by:

$$\boldsymbol{H}(t) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Specifically, secondary user 1 in Fig. 1 can currently access channel 1 only (as indicated by the first row of the $\boldsymbol{H}(t)$ matrix above), while secondary user 2 can currently access either channels $1, 2,$ or 3 (as indicated by the second row in the $\boldsymbol{H}(t)$ matrix). We assume that the mobility process of the secondary users is such that the resulting $\boldsymbol{H}(t)$ process is Markovian and has a well defined steady state distribution. However, the transition probabilities associated with this Markov Chain could be unknown.

2.2 Interference Model

Let $\boldsymbol{S}(t) = (S_1(t), S_2(t), \ldots, S_M(t))$ represent the current primary user occupancy state of the M channels. Here, $S_i(t) \in \{0, 1\}$ (for $i \in \{1, 2, \ldots, M\}$) with the interpretation that $S_i(t) = 0$ if channel i is occupied by primary user i in timeslot t and $S_i(t) = 1$ if primary user i is idle in timeslot t. We assume that exactly 1 packet can be transmitted over any channel in a timeslot. A secondary user can attempt transmission over at most 1 channel subject to the constraints in $\boldsymbol{H}(t)$. This transmission is successful only when the channel is not being used by its primary user or any other secondary user. If a secondary user transmits on a channel which is busy, there is a collision and both packets are lost[1].

To capture the interference that a secondary user transmission may cause on other channels, for all $n \in \{1, 2, \ldots, N\}, m \in \{1, 2, \ldots, M\}$, we define $\mathcal{I}_{nm}(t)$ as the set of channels that secondary user n interferes with when it uses channel m in timeslot t . We include m in the set $\mathcal{I}_{nm}(t)$. We further define the following indicator variables (to be used later):

$$I^k_{nm}(t) = \begin{cases} 1 & \text{if } k \in \mathcal{I}_{nm}(t) \qquad \forall\ k \in \{1, 2, \ldots, M\} \\ 0 & \text{else} \end{cases}$$

Clearly, $I^m_{nm}(t) = 1$ for all m, n, t. Under this interference model, the following two conditions are necessary for a transmission by secondary user n on channel m in slot t to be successful:

1) $S_m(t) = 1$
2) For all other secondary users i transmitting on a channel $j \in \{1, 2, \ldots, M\}$, we have $m \notin \bigcup \mathcal{I}_{ij}(t)$ (where $i \in \{1, 2, \ldots, N\} \setminus \{n\}$)

This interference model is general enough to capture scenarios in which the channels may not be orthogonal with respect to the secondary user transmissions although they are orthogonal for the primary user transmissions. Further, it is general enough to model scenarios where these sets could also change over time (possibly depending on the secondary user location). In most practical situations, the cardinality of the interference sets $\mathcal{I}_{nm}(t)$ would be small. An important special case is when the channels are indeed orthogonal for all secondary user transmissions, so that $\mathcal{I}_{nm}(t) = \{m\}$ for all m, n, t.

As an example, consider secondary user 4 in Fig. 1, and suppose this user transmits a packet over channel 2. Under an orthogonal channel model, we would have

1. We assume that multi-user detection/interference cancelation is not available so that if the secondary user attempts to transmit its own data when some other user is also transmitting, there is enough interference at the access point and no data is successfully received.

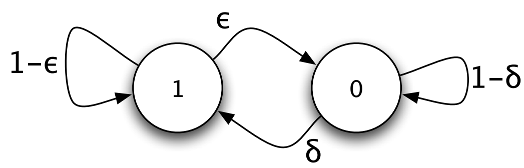


Fig. 2. Two state Markov Chain example for primary user channel occupancy process

$\mathcal{I}_{42}(t) = \{2\}$, as this transmission would not interfere with any other channels. However, in a model where channels are not necessarily orthogonal, it might be that channel 2 uses the same frequency as channel 1, in which case we would have $\mathcal{I}_{42}(t) = \{2, 1\}$, as the current location of node 4 may be close enough to interfere with channel 1 (even though it is not close enough to communicate over channel 1). Note that this $\mathcal{I}_{42}(t)$ set can potentially change over time if node 4 moves to a location that would no longer interfere with channel 1.

2.3 Primary User Traffic Model

We assume that the primary user channel occupancy process $\boldsymbol{S}(t)$ evolves according to a finite state ergodic Markov Chain on the state space $\{0,1\}^M$ and is independent of the secondary user mobility process $\boldsymbol{H}(t)$. It is further assumed to be independent of the control actions of the secondary users. In particular, we assume that the primary users do not attempt retransmissions when collisions take place. For example, the primary users may be using a voice application which can tolerate some lost packets, but has strict delay constraints so that retransmissions are not done. Another example is where the primary users use erasure codes such that the data can be recovered even when some packets are lost.

Each primary user m receives exogenous data at a rate $\nu_m \leq 1$ packet/slot and can tolerate a maximum time average rate of collisions given by $\rho_m \nu_m$, where $\rho_m < 1$ is the maximum fraction of primary user m packets that can have collisions and is known to the secondary users. For example, $\rho_m = 0.05$ means that at most 5% of primary user m packets can have collisions.

2.4 Channel State Information Model

The channel state information available to the secondary users is described by a probability vector $\boldsymbol{P}(t) = (P_1(t), P_2(t), \ldots, P_M(t))$ where $P_i(t)$ is the probability that primary user i is idle in timeslot t. The $\boldsymbol{P}(t)$ process is assumed to be *modulated* by a finite state discrete time Markov Chain (DTMC). Specifically, let $\chi(t)$ represent a finite state DTMC that represents the state of the primary users (where "state" is an abstract term here and could be different in different examples, e.g., it could be $\boldsymbol{S}(t-1)$, the channel occupancy state in the previous slot). The $\chi(t)$ process is assumed to be independent of the control actions. Then for each channel m and each slot t, we define $P_m(t) = Pr[S_m(t) = 1|\chi(t)]$. Thus, $P_m(t)$ is modulated by this process and hence is also independent of the control actions.

We assume that this information is obtained either through a knowledge of the traffic statistics of the primary users, or by sensing the channels, or a combination of these.[2] We discuss two examples of these scenarios in the following.

Example 1: Using knowledge of traffic statistics: Consider a single primary user whose channel occupancy process $S(t)$ is described by a 2 state Markov Chain as shown in Fig. 2. Suppose the last state of the Markov Chain is known at the beginning of each slot and let $\chi(t) = S(t-1)$. If the transition probabilities ϵ and δ associated with this Markov Chain are known, then one can compute $P(t) = Pr[S(t) = 1|S(t-1)]$. Specifically, $Pr[S(t) = 1|S(t-1) = 0] = \delta$ and $Pr[S(t) = 1|S(t-1) = 1] = 1-\epsilon$. A secondary user can obtain this information, for example, by querying the primary user base station that knows $\chi(t)$, so that it is able to tell the current $P(t)$ value. It can be seen that in this example $P(t)$ is modulated by the 2 state $\chi(t)$ process.

Example 2: Using a combination of channel sensing and traffic statistics: In the example above, suppose a secondary user also *senses* the current channel state $S(t)$ and uses a detection algorithm that outputs $\tilde{S}(t)$ as follows:

$$\text{if } S(t) = 0,\ \tilde{S}(t) = \begin{cases} 1 & \text{w.p. } p \\ 0 & \text{w.p. } 1-p \end{cases}$$

$$\text{if } S(t) = 1,\ \tilde{S}(t) = \begin{cases} 1 & \text{w.p. } 1-q \\ 0 & \text{w.p. } q \end{cases}$$

Here, p and q can be thought of as the probabilities of false detection associated with the sensing mechanism. Similar models have been considered in [11], [12].

Let $\chi(t) = [\tilde{S}(t), S(t-1)]$. Then, a secondary user can compute $P(t)$ as follows:

If $\tilde{S}(t) = 1$:

$$\begin{aligned} P(t) &= Pr[S(t) = 1|\tilde{S}(t) = 1, S(t-1)] \\ &= Pr[\tilde{S}(t) = 1|S(t) = 1, S(t-1)]\frac{Pr[S(t) = 1|S(t-1)]}{Pr[\tilde{S}(t) = 1|S(t-1)]} \\ &= \frac{(1-q)Pr[S(t) = 1|S(t-1)]}{(1-q)Pr[S(t) = 1|S(t-1)] + pPr[S(t) = 0|S(t-1)]} \end{aligned}$$

If $\tilde{S}(t) = 0$:

$$\begin{aligned} P(t) &= Pr[S(t) = 1|\tilde{S}(t) = 0, S(t-1)] \\ &= Pr[\tilde{S}(t) = 0|S(t) = 1, S(t-1)]\frac{Pr[S(t) = 1|S(t-1)]}{Pr[\tilde{S}(t) = 0|S(t-1)]} \\ &= \frac{qPr[S(t) = 1|S(t-1)]}{qPr[S(t) = 1|S(t-1)] + (1-p)Pr[S(t) = 0|S(t-1)]} \end{aligned}$$

In this example too, it can be seen that $P(t)$ is modulated by the $\chi(t)$ process.

2. In addition, prediction based techniques could also be used to get this information.

Our model for the channel state information captures the situations where the exact channel state may not be available to the secondary users (e.g., due to limitations in carrier sensing). These probabilities capture the inherent sensing measurement errors associated with any primary transmission detection algorithm. Intuitively, the "closer" $\boldsymbol{P}(t)$ is to $\boldsymbol{S}(t)$, the smaller the chances of collisions.

2.5 Queueing Dynamics and Control Decisions

Each secondary user n receives data according to an arrival process $A_n(t)$ that has rate λ_n packets/slot. We assume that the maximum number of arrivals to any secondary user n is upper bounded by a constant value A_{max} every timeslot. This data arrives at the transport layer and flow control decisions on how many packets to admit to the network layer are taken by each secondary user. We assume that there are no transport layer buffers and add/drop decisions are taken immediately.

Let $U_n(t)$ be the backlog in the network layer queue of secondary user n at the beginning of timeslot t. Let $R_n(t)$ be the control decision that denotes the number of new packets admitted into this queue in slot t. Define $\mu_{nm}(t)$ as the control decision that allocates channel m to secondary user n in slot t. In this model $\mu_{nm}(t) \in \{0,1\} \;\forall\; m,n$ with the interpretation that $\mu_{nm}(t) = 1$ if secondary user n transmits on channel m and $\mu_{nm}(t) = 0$ else. Note that there is a successful transmission on channel m only when the necessary conditions specified earlier are met. Then the queueing dynamics of secondary user n under these control decisions is described by:

$$U_n(t+1) = \max[U_n(t) - \sum_{m=1}^{M} \mu_{nm}(t) S_m(t), 0] + R_n(t) \tag{1}$$

where

$$\mu_{nm}(t) \in \{0,1\} \;\forall\; m,n \tag{2}$$
$$\mu_{nm}(t) \leq h_{nm}(t) \;\forall\; m,n \tag{3}$$
$$0 \leq \sum_{m=1}^{M} \mu_{nm}(t) \leq 1 \;\forall\; n \tag{4}$$
$$\mu_{nm}(t) = 1 \iff \sum_{j=1}^{M} \sum_{\substack{i=1\\ i\neq n}}^{N} I_{ij}^{m}(t)\mu_{ij}(t) = 0 \;\forall\; m,n \tag{5}$$
$$0 \leq R_n(t) \leq A_n(t) \tag{6}$$

Here, inequality (3) represents the constraint imposed by the channel accessibility matrix $\boldsymbol{H}(t)$. Inequality (4) represents the constraint that a secondary user can be allocated at most 1 channel. (5) represents the second necessary condition for successful transmission expressed in terms of the $I_{nm}^{k}(t)$ variables. In the special case of orthogonal channels, this simplifies to the constraint that a channel can be allocated to at most 1 secondary user, i.e.,

$$0 \leq \sum_{n=1}^{N} \mu_{nm}(t) \leq 1 \;\forall\; m \tag{7}$$

2.6 Discussion of Network Model

The above network model considers access point based networks with static (or locally mobile) licensed and fully mobile unlicensed users. Examples of real networks that can be modeled like this include Wi-Fi, cellular and mesh networks with both licensed and unlicensed users. In such networks, the licensed users may not schedule their transmissions and thus send at any time they desire. The unlicensed users must make an effort to opportunistically use the spectrum holes without interfering too much with the licensed users, and hence need sophisticated scheduling mechanisms.

A taxonomy of different approaches to spectrum sharing in cognitive networks is provided in [4]. The network model used in this paper falls into the "spectrum overlay" approach to spectrum sharing.

3 Maximum Throughput Objective

Let r_n denote the time average rate of admitted data for secondary user n, i.e.,

$$r_n = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} R_n(\tau)$$

Let $\boldsymbol{r} = (r_1, \ldots, r_N)$ denote the vector of these time average rates.

We define the following "collision" variables for each primary user $m \in \{1, \ldots, M\}$:

$$C_m(t) = \begin{cases} 1 & \text{if there was a collision with primary user} \\ & \text{in channel } m \text{ in slot } t \\ 0 & \text{else} \end{cases}$$

Let c_m denote the time average rate of collision for primary user m, i.e.,

$$c_m = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} C_m(\tau)$$

Let $\{\theta_1, \ldots, \theta_N\}$ be a collection of positive weights. Then the control objective is to design a flow control and scheduling policy that yields time average rate vector $\boldsymbol{r}$ that solves the following optimization problem:

$$\begin{aligned} \text{Maximize:} \quad & \sum_{n=1}^{N} \theta_n r_n \\ \text{Subject to:} \quad & 0 \leq r_n \leq \lambda_n \;\forall\; n \in \{1, \ldots, N\} \\ & c_m \leq \rho_m \nu_m \;\forall\; m \in \{1, \ldots, M\} \\ & \boldsymbol{r} \in \Lambda \end{aligned}$$

Here, Λ represents the *network capacity region* for the network model as described above. It is defined as the set of all input rate vectors $\vec{\lambda} = (\lambda_1, \ldots, \lambda_N)$ of the secondary users for which a scheduling strategy exists that can support $\vec{\lambda}$ (without flow control) subject to the constraints imposed by the network. The notion of network capacity for general networks with time varying channels and energy constraints is formalized in [15], [17], [18] where it is shown to be a function of the steady state network topology distribution, channel probabilities, and time average transmission rates.

Let $\boldsymbol{r}^* = (r_1^*, \ldots, r_N^*)$ denote the optimal solution to the optimization problem defined above. In principle, it can be solved if all system parameters are known in advance including Λ. However, in practice, this region may not be known to the network controller (e.g., because the mobility patterns of the secondary users are unknown) and the above maximization problem must be done for input rates either inside or outside of the capacity region. Even if all system parameters are known, the optimal solution may be difficult to implement as it may require centralized coordination among all users.

We next present an online control algorithm that overcomes all of these challenges.

4 Optimal Control Algorithm

We now present the *Cognitive Network Control Algorithm* (CNC), a cross-layer control strategy that can be shown to achieve the optimal solution $\boldsymbol{r}^*$ to the network optimization problem presented earlier. It operates without knowledge of whether the input rate is within or outside of the capacity region Λ. Further, it provides *deterministic worst case* bounds on the maximum secondary user queue backlog at all times and the maximum number of collisions with a primary user in a given time interval. These are much stronger than probabilistic performance guarantees. Finally, it offers a control parameter V that enables an explicit trade-off between the average throughput utility and delay. This algorithm is similar in spirit to the "backpressure" algorithms proposed in [17], [19] for problems of energy optimal networking in wireless ad-hoc and mesh networks.

The algorithm is decoupled into two separate components. The first component performs optimal flow control at the transport layers and is implemented independently at each secondary user. The second component determines a network wide resource allocation every slot and needs to be solved collectively by the secondary users.

In addition to the actual queue backlog $U_n(t)$, this algorithm uses a set of collision queues $X_m(t)$ for each channel m. These queues are "virtual" in that they are maintained purely in software. These are used to track the amount by which the number of collisions suffered by a primary user m exceeds its time average collision fraction ρ_m. These could be maintained at the primary user base station for each channel. We assume that the secondary users are aware of the $X_m(t)$ value for each channel m that they can access at time t.

We define the collision queue $X_m(t)$ for channel m as follows:

$$X_m(t+1) = \max[X_m(t) - \rho_m 1_m(t), 0] + C_m(t) \qquad (8)$$

where $C_m(t)$ is the collision variable for channel m as defined in the previous section and $1_m(t)$ is an indicator variable, taking value 1 if primary user m transmits in slot t and 0 else (so that $1_m(t) = 1 - S_m(t)$).

The above equation represents the queueing dynamics of a single server system with input process $C_m(t)$ and service process $\rho_m 1_m(t)$. This system is stable only when the service rate is greater than or equal to the input rate, i.e.,

$$c_m = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} C_m(\tau) \leq \lim_{t\to\infty} \rho_m \frac{1}{t} \sum_{\tau=0}^{t-1} 1_m(\tau) = \rho_m \nu_m$$

This is precisely the collision constraint in the utility optimization problem stated earlier. *Thus, if our policy stabilizes all collision queues as defined above, the maximum average rate of collisions will meet the required constraint*. This technique of turning time average constraints into queueing stability problems was introduced in [17] where it was used for satisfying average power constraints.

4.1 Cognitive Network Control Algorithm (CNC)

Let $V \geq 0$ be a fixed control parameter. Let the flow control and resource allocation decision under the CNC algorithm be $R_n^{CNC}(t)$ and $\mu_{nm}^{CNC}(t)$ respectively. These are determined as follows:

Flow Control: At each secondary user n, choose the number of packets to admit $R_n^{CNC}(t)$ as the solution to the following problem:

$$\begin{aligned} \text{Minimize:} \quad & R_n(t)[U_n(t) - V\theta_n] \\ \text{Subject to:} \quad & 0 \leq R_n(t) \leq A_n(t) \end{aligned} \qquad (9)$$

This problem has a simple threshold-based solution. In particular, if the current queue backlog $U_n(t) > V\theta_n$, then $R_n^{CNC}(t) = 0$ and no new packets are admitted. Else, if $U_n(t) \leq V\theta_n$, then $R_n^{CNC}(t) = A_n(t)$ and all new packets are admitted. Note that this can be solved separately at each user and does not require knowledge of θ_n weights of other users.

Resource Allocation: Choose a resource allocation $\mu_{nm}^{CNC}(t)$ that solves the following problem:

$$\text{Max:} \sum_{n,m} \mu_{nm}(t) \Big[U_n(t) P_m(t) - \sum_{k=1}^{M} X_k(t)(1 - P_k(t)) I_{nm}^k(t) \Big]$$
$$\text{Subject to:} \quad \text{constraints } (2), (3), (4), (5) \qquad (10)$$

After observing the outcome of this allocation at the end of the slot, the virtual queues are updated as in (8) based on the feedback received about a collision with a primary user or a successful transmission. Note that only collisions with a primary user affect (8), collisions between secondary users do not affect the virtual collision queues.

The above problem is a generalized Maximum Weight Match problem where the weight for a pair (n,m) is given by $\left(U_n(t)P_m(t) - \sum_{k=1}^{M} X_k(t)(1-P_k(t))I_{nm}^k(t)\right)$. This is the difference between the current queue backlog $U_n(t)$ weighted by the probability that primary user m is idle and the weighted sum of all collision queue backlogs $X_k(t)$ for the channels that user n interferes with if it uses channel m. The weight for a collision queue is the probability that the corresponding primary user will transmit. Note that if this difference is non-positive, then the link (n,m) can be removed from the decision options, simplifying scheduling. This problem is hard to solve in general, though constant factor approximations exist that are easier to implement. We discuss these in Sec. 6.

For the case when all channels are orthogonal from the point of view of secondary users (which means a secondary user transmission on a channel does not cause interference to other channels), $\mathcal{I}_{nm}(t) = \{m\}$ so that $I_{nm}^m(t) = 1, I_{nm}^k(t) = 0 \ \forall \ k \neq m$. Then the above maximization simplifies to the following problem:

$$\begin{aligned} \text{Maximize: } & \sum_{n,m} \mu_{nm}(t)\Big[U_n(t)P_m(t) - X_m(t)(1-P_m(t))\Big] \\ \text{Subject to: } & \quad \text{constraints } (2), (3), (4), (7) \end{aligned} \tag{11}$$

The above maximization requires solving the Maximum Weight Match (MWM) problem on an $N \times M$ bipartite graph of N secondary users and M channels. This problem can be solved in polynomial time, though this may require centralized control. We discuss simpler constant factor approximations in Sec. 6. Also, we consider a cell partitioned network in the simulations of Sec. 7 for which a full maximum weight match can be implemented in a distributed manner.

To get an intuition behind the algorithm, consider the maximization in (11) for the orthogonal channel case. A secondary user n would attempt transmission over channel m only if $U_n(t)P_m(t) > X_m(t)(1-P_m(t))$. Intuitively, this algorithm tries to schedule secondary users with larger queue backlogs over those channels that are more likely to be idle and that have smaller "effective" collision queue values. Here, the effective collision queue value is its actual value weighted by the probability of that channel being busy with its primary user. Intuitively, these collision queues enable stochastic optimization by acting as dynamic Lagrange multipliers [18]. Using (11), the dynamic weights of $X_m(t)$ help determine the best channel for attempting transmission.

4.2 Performance Analysis

We now characterize the performance of the CNC algorithm. This holds for general secondary user mobility processes that are described by finite state ergodic Markov Chains.

Theorem 1: (Algorithm Performance) Assume that all queues are initialized to 0. Suppose all arrivals $A_n(t)$ are upper bounded so that $A_n(t) \leq A_{max}$ for all n, t. Also suppose the $\boldsymbol{H}(t)$ and $\boldsymbol{P}(t)$ processes are Markovian and have a well defined steady state distribution. Then, implementing the CNC algorithm every slot for any fixed control parameter $V \geq 0$ stabilizes all real and virtual queues (thereby satisfying the maximum time average collision constraints) and yields the following performance bounds:

1) The worst case queue backlog for each secondary user n is upper bounded by a finite constant U_{max}^n for all t:

$$U_n(t) \leq U_{max}^n \triangleq V\theta_n + A_{max} \tag{12}$$

Let $\theta_{max} = \max_{n \in \{1,\ldots,N\}}\{\theta_n\}$. Then, from (12) we have for any n

$$U_n(t) \leq U_{max} \triangleq V\theta_{max} + A_{max} \tag{13}$$

2) For all m, t such that $P_m(t) \neq 1$, let $\epsilon > 0$ be such that $P_m(t) \leq 1 - \epsilon$.[3] Then, the worst case collision queue backlog for all channels m is upper bounded by a finite constant X_{max}:

$$X_m(t) \leq X_{max} \triangleq U_{max}\frac{(1-\epsilon)}{\epsilon} + 1 \tag{14}$$

Further, the worst case number of collisions suffered by any primary user m is no more than $\rho_m T + X_{max}$ over any interval (of size greater than or equal to T slots) over which the primary user transmits T times, for any positive integer T.

3) The time average throughput utility achieved by the CNC algorithm is within $\tilde{B}/V$ of the optimal value:

$$\liminf_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{n=1}^{N}\theta_n \mathbb{E}\left\{R_n(\tau)\right\} \geq \sum_{n=1}^{N}\theta_n r_n^* - \frac{\tilde{B}}{V} \tag{15}$$

where $\tilde{B} = B + C_U + C_X + N + M$ and where B, C_U, C_X are constants (defined precisely in (18), (31), (32)). The constants C_U and C_X are determined by the stochastics of the mobility and channel state probability processes and it is shown in Appendix A that these are $O(\log V)$ when these processes evolve according to any finite state ergodic Markov model.

Therefore, by part (3) of the theorem, the achieved average throughput utility is within $O(\log V/V)$ of the optimal value. This can be pushed arbitrarily close to the optimal value by increasing the control parameter V. However, this increases the maximum queue backlog bound U_{max} linearly in V, leading to a utility-delay tradeoff.

The above bounds are quite strong. In particular, the maximum collisions bound in part (2) gives deterministic performance guarantees that hold for *any* interval size. This is quite useful in the context of cognitive networks since it implies that the licensed users are guaranteed to suffer at most these many collisions. Probabilistic guarantees (e.g [11], [13]) do not provide such bounds.

3. Such an ϵ exists for any finite state ergodic Markov Chain.

We next prove the first two parts of Theorem 1. Proof of part (3) uses the technique of Stochastic Lyapunov Optimization and is provided in the next section.

Proof of part (1): Suppose that $U_n(t) \leq U_{max}^n$ for all $n \in \{1, \ldots, N\}$ for some time t. This is true for $t = 0$ as all queues are initialized to 0. We show that the same holds for time $t+1$. We have 2 cases. If $U_n(t) \leq U_{max}^n - A_{max}$, then from (1), we have $U_n(t+1) \leq U_{max}$ (because $R_n(t) \leq A_{max}$ for all t). Else, if $U_n(t) > U_{max}^n - A_{max}$, then $U_n(t) > V\theta_n + A_{max} - A_{max} = V\theta_n$. Then, the flow control part of the algorithm chooses $R_n(t) = 0$, so that by (1):

$$U_n(t+1) \leq U_n(t) \leq U_{max}^n$$

This proves (12). □

Proof of part (2): Suppose that $X_m(t) \leq X_{max}$ for all $m \in \{1, \ldots, M\}$ for some time t. This is true for $t = 0$ as all queues are initialized to 0. We show that the same holds for time $t+1$. First suppose $P_m(t) = 1$. Then, by definition, there is no collision with the primary user in channel m in slot t so that $C_m(t) = 0$. Then, from (8), we have $X_m(t+1) \leq X_{max}$. Next, suppose $P_m(t) < 1$. We again have 2 cases. If $X_m(t) \leq X_{max} - 1$, then from (8), we have $X_m(t+1) \leq X_{max}$ (because $C_m(t) \leq 1$ for all t). Else, if $X_m(t) > X_{max} - 1 = U_{max}\frac{(1-\epsilon)}{\epsilon}$, then $X_m(t)\epsilon > U_{max}(1-\epsilon)$. This implies $X_m(t)(1 - P_m(t)) \geq X_m(t)\epsilon > U_{max}(1-\epsilon) \geq U_{max}P_m(t) \geq U_n(t)P_m(t)$ for all $n \in \{1, \ldots, N\}$. Thus, the resource allocation part of the algorithm chooses $\mu_{nm}(t) = 0$ for all n. This would yield $C_m(t) = 0$ (since no collision takes place with primary user m), so that by (8):

$$X_m(t+1) \leq X_m(t) \leq X_{max}$$

This proves (14). □

Now consider any interval (t_1, t_2) in which primary user m transmits T times. Then, from the queueing equation (8) we have that:

$$X_m(t_2+1) \geq X_m(t_1) + \sum_{\tau=t_1}^{t_2} C_m(\tau) - \rho_m T$$

This follows by noting that $\rho_m T$ is the maximum number of "departures" that can take place in the queueing dynamics (8) during the interval (t_1, t_2). From this, we can bound the worst case number of collisions suffered by primary user m over any interval in which it transmits T times as:

$$\sum_{\tau=t_1}^{t_2} C_m(\tau) \leq \rho_m T + X_{max}$$

5 Stochastic Lyapunov Optimization

Let $\boldsymbol{Q}(t) = (Q_1(t), \ldots, Q_K(t))$ be a vector process of queue lengths for a discrete time stochastic queueing network with K queues (possibly including some virtual queues like the collision queues defined in the previous subsection). Let $L(\boldsymbol{Q})$ be any non-negative scalar valued function of the queue lengths, called a Lyapunov function. Define the Lyapunov drift $\Delta(t)$ as follows:

$$\Delta(t) \triangleq \mathbb{E}\left\{L(\boldsymbol{Q}(t+1)) - L(\boldsymbol{Q}(t))\right\}$$

Suppose the network accumulates "rewards" every timeslot (where rewards might correspond to utility measures of control actions). Assume rewards are real valued and bounded, and let the stochastic process $f(t)$ represent the reward earned during slot t. Let f^* represent the target reward. The following result (a variant of related results from [17], [18]) specifies a drift condition which ensures that the time average of the reward process $f(t)$ is close to meeting or exceeding f^*.

Theorem 2: (Delayed Lyapunov Optimization with Rewards) Suppose there exist finite constants $V > 0, B > 0, d > 0$, and a non-negative function $L(\boldsymbol{Q})$ such that $\mathbb{E}\left\{L(\boldsymbol{Q}(d))\right\} < \infty$ and for every timeslot $t > d$, the Lyapunov drift satisfies:

$$\Delta(t) - V\mathbb{E}\left\{f(t)\right\} \leq B - Vf^* \tag{16}$$

then we have:

$$\liminf_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{f(\tau)\right\} \geq f^* - \frac{B}{V}$$

Proof: Inequality (16) holds for all $t > d$. Summing both sides over $\tau \in \{d, \ldots, t-1\}$ yields:

$$\begin{aligned}\mathbb{E}\left\{L(\boldsymbol{Q}(t))\right\} - \mathbb{E}\left\{L(\boldsymbol{Q}(d))\right\} \leq & B(t-d) - V(t-d)f^* \\ & + V\sum_{\tau=d}^{t-1} \mathbb{E}\left\{f(\tau)\right\}\end{aligned}$$

Rearranging terms, dividing by t, and using non-negativity of $L(\boldsymbol{Q})$ yields:

$$\frac{(t-d)f^*}{t} - \frac{(t-d)B}{tV} - \frac{\mathbb{E}\left\{L(\boldsymbol{Q}(d)\right\}}{tV} \leq \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{f(\tau)\right\}$$

The result follows by taking limit as $t \to \infty$. □

We now use Theorem 2 to prove part (3) of Theorem 1. This is done by comparing the Lyapunov drift of the CNC algorithm with that of a stationary randomized algorithm $STAT$ that makes control decisions every slot purely as a function of the current channel state information $\boldsymbol{P}(t)$ and $\boldsymbol{H}(t)$.

We first obtain an expression for the Lyapunov drift under any control policy for our cognitive network model.

5.1 Lyapunov Drift

Let $\boldsymbol{Q}(t) = (U_1(t), \ldots, U_N(t), X_1(t), \ldots, X_M(t))$ represent the collection of all real and virtual queue backlogs in the cognitive network. We define the following Lyapunov function:

$$L(\boldsymbol{Q}(t)) \triangleq \frac{1}{2}\Big[\sum_{n=1}^{N} U_n^2(t) + \sum_{m=1}^{M} X_m^2(t)\Big]$$

Using queueing dynamics (1) and (8), the Lyapunov drift $\Delta(t)$ under any control policy (including CNC) can be computed as follows:

$$\Delta(t) \leq B - \mathbb{E}\left\{\sum_{n=1}^{N} U_n(t)\Big(\sum_{m=1}^{M}\mu_{nm}(t)S_m(t) - R_n(t)\Big)\right\} - \mathbb{E}\left\{\sum_{m=1}^{M} X_m(t)(\rho_m 1_m(t) - C_m(t))\right\} \quad (17)$$

where

$$B \triangleq \frac{N(A_{max}^2 + 1) + \sum_{m=1}^{M}\rho_m^2 + M}{2} \quad (18)$$

The collision variable $C_m(t)$ can be expressed in terms of the control decisions $\mu_{ij}(t)$ and channel state $\boldsymbol{S}(t)$ as follows:

$$C_m(t) = \sum_{i=1}^{N}\sum_{j=1}^{M}\mu_{ij}(t)I_{ij}^m(t)1_{[U_i(t)>0]}(1 - S_m(t)) \quad (19)$$

where $1_{[U_i(t)>0]}$ is an indicator variable of non-zero queue backlog in secondary user i. This follows by observing that a collision with the primary user occurs in channel m if the primary user is busy (i.e. $S_m(t) = 0$) and if $\mu_{ij}(t) = 1$ for some secondary user i with non-zero backlog using channel j that interferes with channel m. We will find it useful to define the following related variable:

$$\hat{C}_m(t) = \sum_{i=1}^{N}\sum_{j=1}^{M}\mu_{ij}(t)I_{ij}^m(t)(1 - S_m(t)) \quad (20)$$

For a given control parameter $V \geq 0$, we subtract the reward metric $V\mathbb{E}\left\{\sum_{n=1}^{N}\theta_n R_n(t)\right\}$ from both sides of the drift inequality (17) and use the fact that $\hat{C}_m(t) \geq C_m(t)\ \forall t$ to get the following:

$$\Delta(t) - V\mathbb{E}\left\{\sum_{n=1}^{N}\theta_n R_n(t)\right\} \leq B - \mathbb{E}\left\{\sum_{n=1}^{N} U_n(t)\Big(\sum_{m=1}^{M}\mu_{nm}(t)S_m(t) - R_n(t)\Big)\right\} - \mathbb{E}\left\{\sum_{m=1}^{M} X_m(t)(\rho_m 1_m(t) - \hat{C}_m(t))\right\} - V\mathbb{E}\left\{\sum_{n=1}^{N}\theta_n R_n(t)\right\} \quad (21)$$

5.2 Optimal Stationary, Randomized Policy

We now describe the stationary, randomized policy $STAT$ that chooses control actions only as a function of $\boldsymbol{P}(t)$ and $\boldsymbol{H}(t)$ every slot. We have the following fact:

Optimal Stationary, Randomized Policy: For any rate vector $(\lambda_1, \ldots, \lambda_N)$ (inside or outside of the network capacity region Λ), there exists a stationary randomized scheduling policy $STAT$ that chooses feasible allocations $R_n^{STAT}(t), \mu_{nm}^{STAT}(t)$ for all $n \in \{1, \ldots, N\}, m \in \{1, \ldots, M\}$ every slot as a function of the channel state information $\boldsymbol{P}(t)$ and $\boldsymbol{H}(t)$ and yields the following steady state values:

$$\mathbb{E}\left\{R_n^{STAT}(t)\right\} = r_n^*\ \forall\ t \quad (22)$$

$$\mu_n^{STAT} \triangleq \lim_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{\sum_{m=1}^{M}\mu_{nm}^{STAT}(\tau)S_m(\tau)\right\} \geq r_n^* \quad (23)$$

$$\hat{c}_m^{STAT} \triangleq \lim_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{\hat{C}_m^{STAT}(t)\right\} \leq \rho_m\nu_m \quad (24)$$

Specifically, the flow control decision $R_n^{STAT}(t)$ under this policy is determined as follows. At each secondary user n, observe $A_n(t)$ and choose $R_n(t)^{STAT}$ as follows:

$$R_n^{STAT}(t) = \begin{cases} A_n(t) & \text{with probability } r_n^*/\lambda_n \\ 0 & \text{else} \end{cases}$$

These probabilistic decisions are made every slot independent of the current queue backlogs and are i.i.d with probability $r_n^*/\lambda_n \leq 1$. Thus, we have

$$\mathbb{E}\left\{R_n(t)^{STAT}\right\} = \mathbb{E}\left\{A_n(t)\right\}\frac{r_n^*}{\lambda_n} = r_n^*$$

The above facts can be proven using techniques similar to the ones used in [15]–[17] for showing the existence of capacity achieving stationary, randomized policies that make control decisions independent of queue backlog and is omitted for brevity. We now prove an important property of the CNC algorithm.

Claim: Suppose the CNC algorithm is implemented on all slots up to time t. Thus, the queue backlogs $U_n(t)$ and $X_m(t)$ are determined by the history before time t and are not affected by the control decisions made on slot t. Then, given the current queue backlogs, the CNC control decisions for slot t minimize the right hand side of inequality (21) over all alternative feasible policies that could be implemented on slot t, including the stationary, randomized policy $STAT$.[4]

Proof: By changing the order of summations and using (20), the right side of (21) can be expressed in a more convenient form:

$$B - \sum_{m=1}^{M}\rho_m\mathbb{E}\left\{X_m(t)1_m(t)\right\} + \mathbb{E}\left\{\sum_{n=1}^{N}R_n(t)(U_n(t) - V\theta_n)\right\} - \mathbb{E}\left\{\sum_{n,m}\mu_{nm}(t)\Big[U_n(t)S_m(t) - \sum_{k=1}^{M}X_k(t)(1 - S_k(t))I_{nm}^k\Big]\right\} \quad (25)$$

where we have omitted the t subscript in $I_{nm}^k(t)$. Note that $\mathbb{E}\left\{S_m(t)|\chi(t)\right\} = Pr[S_m(t) = 1|\chi(t)] = P_m(t)\ \forall m$. By writing the last two terms on the right hand side as an iterated expectation by conditioning on the queue

4. Note that we are not claiming that the CNC policy, implemented over time, minimizes the right hand side expectation of (21) at time t. Indeed, another policy may result in a smaller expected queue size outcome at time t. Rather, we are claiming that, given CNC is used up to (but not including) time t (so that queue sizes at time t are already determined by the sample path outcome of CNC up to this time), the CNC control decisions made *at time t* act to greedily minimize the right hand side over any other decisions that can be made at time t.

backlog and $\chi(t)$, it can be seen that CNC chooses control decisions (9) and (10) that minimize these terms for every possible value of the backlog and $\chi(t)$, so that the actual expectation is also minimized. We note that the unconditioning is done with respect to the queue backlog distribution that arises as a result of implementing the CNC algorithm for all slots up to time t.

Using this fact, we have:

$$\begin{aligned}
&\Delta^{CNC}(t) - V\mathbb{E}\left\{\sum_{n=1}^{N}\theta_n R_n^{CNC}(t)\right\} \leq B \\
&- \mathbb{E}\left\{\sum_{n=1}^{N} U_n(t)\Big(\sum_{m=1}^{M}\mu_{nm}^{STAT}(t)S_m(t) - R_n^{STAT}(t)\Big)\right\} \\
&- \mathbb{E}\left\{\sum_{m=1}^{M} X_m(t)(\rho_m 1_m(t) - \hat{C}_m^{STAT}(t))\right\} \\
&- V\mathbb{E}\left\{\sum_{n=1}^{N}\theta_n R_n^{STAT}(t)\right\} \qquad (26)
\end{aligned}$$

In Appendix A, we show that for all $t > d$ (where d is a finite positive integer and is computed in Appendix A), this can be expressed as:

$$\Delta^{CNC}(t) - V\mathbb{E}\left\{\sum_{n=1}^{N}\theta_n R_n^{CNC}(t)\right\} \leq \tilde{B} - V\sum_{n=1}^{N}\theta_n r_n^* \qquad (27)$$

This is in a form that fits (16). Thus, applying Theorem 2 proves (15). □

6 Distributed Implementation

Here we discuss constant factor approximations to the resource allocation problem (10) that are easier to implement in a distributed network. We focus on the orthogonal channel case in which a secondary user transmission on a channel does not cause interference to other channels. As noted earlier, in this case, the resource allocation problem (11) reduces to a Maximum Weight Match (MWM) problem on an $N \times M$ bipartite graph between N secondary users and M channels. An edge exists between nodes n and m of this graph if $h_{nm}(t) = 1$, i.e., if secondary user n can access channel m in slot t. The weight of this edge is given by $(U_n(t)P_m(t) - X_m(t)(1 - P_m(t)))$. While the MWM problem can be solved in polynomial time in a centralized way, here we are interested in simpler implementations. In particular, we use the idea of *Greedy Maximal Weight Match Scheduling* that has been investigated in several recent works including [20]–[22].

A maximal match is defined as any set of edges (m, n) that do not interfere with each other such that adding any new edge to this set necessarily violates a matching constraint. A Greedy Maximal Weight Match can be achieved as follows: First select the edge (m, n) with the largest positive weight and label it "active". Then select the edge with the second largest positive weight (breaking ties arbitrarily) that does not conflict with an active edge and label it active. Continue in the same way, until no more edges can be added. It is not difficult to see that this final set of edges labeled "active" has the desired maximal property. A Greedy Maximal Weight Match can be computed with much less overhead as compared to the Maximum Weight Match.

It can be shown that using such greedy maximal weight matches instead of the maximum weight match every slot can still support any rate within $\frac{1}{2}\Lambda$. In particular, in Appendix C, we show that resource allocation $\mu_{nm}^{GMM}(t)$ chosen according to a Greedy Maximal Weight Match has the following property:

$$\begin{aligned}
&\sum_{n,m}\mu_{nm}^{GMM}(t)\Big[U_n(t)P_m(t) - X_m(t)(1 - P_m(t))\Big] \\
&\geq \frac{1}{2}\sum_{n,m}\mu_{nm}^{CNC}(t)\Big[U_n(t)P_m(t) - X_m(t)(1 - P_m(t))\Big] \qquad (28)
\end{aligned}$$

where $\mu_{nm}^{CNC}(t)$ is the optimal solution to (11). Using this, we get the following result:

Theorem 3: (Performance Bound for Orthogonal Channels with Greedy Maximal Weight Match Scheduling) The time average throughput utility achieved by the CNC algorithm with Greedy Maximal Weight Match Scheduling is within B^{GMM}/V of $\frac{1}{2}\sum_{n=1}^{N}\theta_n r_n^*$:

$$\liminf_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{n=1}^{N}\theta_n\mathbb{E}\left\{R_n(\tau)\right\} \geq \frac{1}{2}\sum_{n=1}^{N}\theta_n r_n^* - \frac{B^{GMM}}{V} \qquad (29)$$

where $B^{GMM} = (\tilde{B} + B)/2$.

We note that while using Greedy Maximal Weight Match Scheduling provides a factor of 2 approximation in terms of the time average throughput utility, the deterministic bounds on maximum queue backlog and worst case number of collisions remain the same as in parts (1) and (2) of Theroem 1. This is because the arguments there were based only on the fact that only positive weight transmissions are scheduled, which also holds for GMM.

Proof: Let $R_n^{GMM}(t)$ and $\mu_{nm}^{GMM}(t)$ denote the flow control and resource allocation decisions under Greedy Maximal Match Scheduling. Let $\Delta^{GMM}(t)$ be the corresponding Lyapunov drift. Note that for any given queue backlog $\boldsymbol{Q}(t)$, $R_n^{GMM}(t) = R_n^{CNC}(t)$. Then, using (25), we have:

$$\begin{aligned}
&\Delta^{GMM}(t) - V\mathbb{E}\left\{\sum_{n=1}^{N}\theta_n R_n^{GMM}(t)\right\} \leq B - \\
&\sum_{m=1}^{M}\rho_m\mathbb{E}\left\{X_m(t)1_m(t)\right\} + \mathbb{E}\left\{\sum_{n=1}^{N}R_n^{GMM}(t)(U_n(t) - V\theta_n)\right\} \\
&- \mathbb{E}\left\{\sum_{n,m}\mu_{nm}^{GMM}(t)\Big[U_n(t)S_m(t) - X_m(t)(1 - S_m(t))\Big]\right\}
\end{aligned}$$

Using property (28) and the fact that $R_n^{GMM}(t) =$

$R_n^{CNC}(t)$, the above can be written as:

$$\Delta^{GMM}(t) - V\mathbb{E}\left\{\sum_{n=1}^{N}\theta_n R_n^{GMM}(t)\right\} \leq B-$$
$$\sum_{m=1}^{M}\rho_m\mathbb{E}\left\{X_m(t)1_m(t)\right\} + \mathbb{E}\left\{\sum_{n=1}^{N}R_n^{CNC}(t)(U_n(t)-V\theta_n)\right\}$$
$$-\frac{1}{2}\mathbb{E}\left\{\sum_{n,m}\mu_{nm}^{CNC}(t)\Big[U_n(t)S_m(t) - X_m(t)(1-S_m(t))\Big]\right\}$$

From (9), note that $R_n^{CNC}(t) \geq 0$ if $U_n(t) \leq V\theta_n$, else $R_n^{CNC}(t) = 0$. Therefore the second to last term under the flow control of CNC is non-positive. Thus, the above can be rewritten as:

$$\Delta^{GMM}(t) - V\mathbb{E}\left\{\sum_{n=1}^{N}\theta_n R_n^{GMM}(t)\right\} \leq B$$
$$-\frac{1}{2}\sum_{m=1}^{M}\rho_m\mathbb{E}\left\{X_m(t)1_m(t)\right\}$$
$$+\frac{1}{2}\mathbb{E}\left\{\sum_{n=1}^{N}R_n^{CNC}(t)(U_n(t)-V\theta_n)\right\}$$
$$-\frac{1}{2}\mathbb{E}\left\{\sum_{n,m}\mu_{nm}^{CNC}(t)\Big[U_n(t)S_m(t) - X_m(t)(1-S_m(t))\Big]\right\}$$

Using (26) and (27), we get the following:

$$\Delta^{GMM}(t) - V\mathbb{E}\left\{\sum_{n=1}^{N}\theta_n R_n^{GMM}(t)\right\} \leq$$
$$B^{GMM} - \frac{V}{2}\sum_{n=1}^{N}\theta_n r_n^*$$

This is in a form that fits (16). Thus, applying Theorem 2 proves (29). □

7 SIMULATIONS

We simulate the CNC algorithm on an example cognitive network consisting of 9 primary users and 8 secondary users as shown in Fig. 3. We consider a simple cell-partitioned network with one primary user per cell. The primary users are static and each has its own licensed channel that can be used by them simultaneously. A secondary user can only attempt to transmit on the channel associated with the primary user in its current cell.

The secondary users move from one cell to another according to a Markovian random walk. In particular, at the end of every slot, a secondary user decides to stay in its current cell with probability $1-\beta$, else decides to move to an adjacent cell with probability $\beta/4$ (where $\beta = 0.25$ for the simulations). If there is no feasible adjacent cell (e.g, if the previous cell is a corner cell and the new chosen cell does not exist), then the user remains in the current cell. It can be shown that the resulting $\boldsymbol{H}(t)$ process forms an irreducible, aperiodic Markov Chain

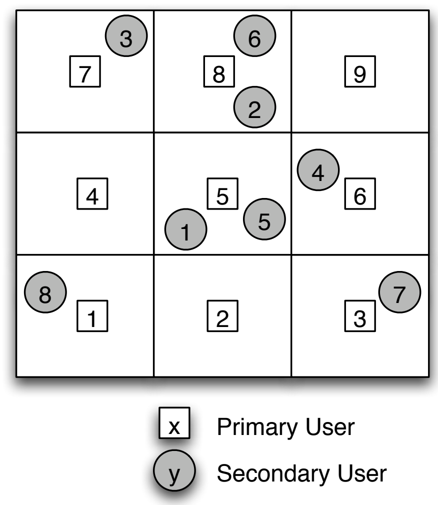


Fig. 3. Example cell-partitioned network used in simulation

where the steady state location distribution is uniform over all cells.

The channel state process $S_m(t)$ for each primary user m is governed by an ON/OFF Markov Chain with symmetric transition probabilities between the ON and OFF states given by $0.2\ \forall m$. The maximum collision fraction $\rho_m = 0.05\ \forall m$ so that for each primary user, at most 5% of its packets can have collisions.

New packets arrive at the secondary users according to independent Bernoulli processes, so that a single packet arrives i.i.d. with probability λ every slot. We assume there are no transport layer storage buffers, so that all packets that are not immediately admitted to the network layer are necessarily dropped. Flow control is performed according to (9) (with $\theta_n = 1\ \forall n$) and resource allocation decisions are made every slot according to (11). In this particular cell-partitioned network structure with one channel per cell, the maximum weight match can be decoupled into a distributed algorithm implemented in each cell, and is the same as the greedy maximal match that selects the largest weight user to transmit in each cell.

In Fig. 4 we plot the average total occupancy (summing all packets in the queues of the secondary users) versus the input rate λ . Each data point represents a simulation over $500,000$ timeslots, and the different curves correspond to values of the flow control parameter $V \in \{1, 2, 5, 10, 100\}$, and the case $V = \infty$ (no flow control) is also shown. In this case, the average total occupancy increases without bound as the input rate approaches network capacity. The vertical asymptote which appears roughly at $\lambda = 0.13$ packets/slot corresponds to this value. Fig. 5 illustrates the achieved throughput versus the raw data input rate λ for various V parameters. The achieved throughput is almost identical to the input rate λ for small values of λ, and the throughput saturates at a value that depends on V, being very close to the 0.13

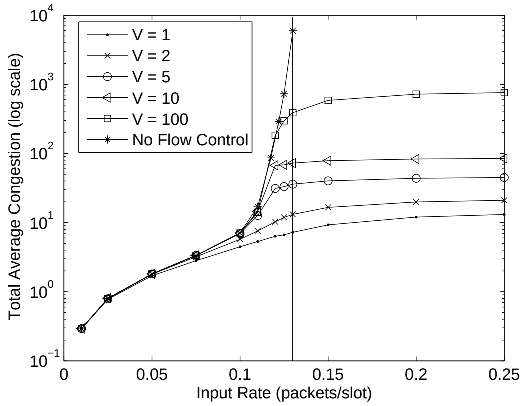


Fig. 4. Total average congestion vs. input rate for different values of V

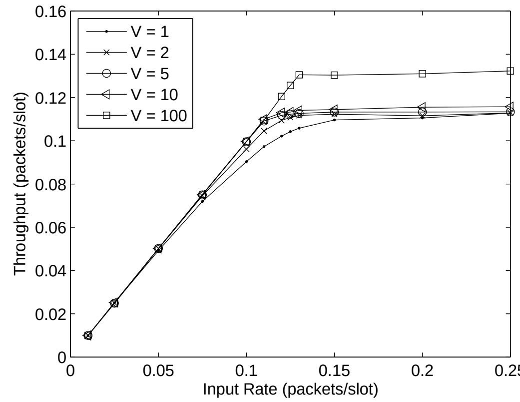


Fig. 5. Achieved throughput vs. input rate for different values of V

capacity level when V is large.

Also, it was found that all real and virtual queue backlogs are always bounded by the maximum values given in (12) and (14). In particular, $\epsilon = 0.2$ for this network, so that $X_m(t) \leq X_{max} = U_{max}\frac{1-\epsilon}{\epsilon}+1 = 4U_{max}+1 = 4V+5$. Finally, the maximum average fraction of collisions was very close to the target $\rho_m = 5\%$.

8 Conclusions

In this paper, we developed an opportunistic scheduling algorithm for cognitive radio networks that maximizes the throughput utility of the secondary users subject to maximum collision constraints with the primary users. We used the recently developed technique of Lyapunov Optimization along with the notion of collision queues to design an online flow control, scheduling and resource allocation algorithm. This algorithm provides tight reliability guarantees in terms of the worst case number of collisions suffered by a primary user in any time interval. Further, its performance can be pushed arbitrarily close to the optimal value with a trade-off in the average delay.

Appendix A
Lyapunov Drift under policy STAT

Here, we use "delayed" queue backlogs to express the Lyapunov drift of the CNC algorithm in a form that fits (16). Recall that $R_n^{STAT}(t)$ and $\mu_{nm}^{STAT}(t)$ denote the resource allocation decisions under the stationary, randomized policy $STAT$ introduced in Sec. 5.2. We use the following sample path inequalities. Specifically, for all $t > d$, we have for each secondary user queue $U_n(t)$ and for each collision queue $X_m(t)$:

$$U_n(t-d) + dA_{max} \geq U_n(t) \geq U_n(t-d) - d$$
$$X_m(t-d) + d \geq X_m(t) \geq X_m(t-d) - d\rho_m$$

These follow by noting that the queue backlog at time t cannot be smaller than the queue backlog at time $(t-d)$ minus the maximum possible departures in duration $(t-d, d)$. Similarly, it cannot be larger than the queue backlog at time $(t-d)$ plus the maximum possible arrivals in duration $(t-d, d)$. Using these in (26) and using $\mathbb{E}\left\{R_n^{STAT}(t)\right\} = r_n^*$ (from (22)), we get:

$$\Delta^{CNC}(t) - V\mathbb{E}\left\{\sum_{n=1}^{N}\theta_n R_n^{CNC}(t)\right\} \leq B + C_U + C_X$$
$$- \mathbb{E}\left\{\sum_{n=1}^{N} U_n(t-d)\Big(\sum_{m=1}^{M}\mu_{nm}^{STAT}(t)S_m(t) - R_n^{STAT}(t)\Big)\right\}$$
$$- \mathbb{E}\left\{\sum_{m=1}^{M} X_m(t-d)(\rho_m 1_m(t) - \hat{C}_m^{STAT}(t))\right\} - V\sum_{n=1}^{N}\theta_n r_n^* \quad (30)$$

where C_U and C_X are given by:

$$C_U \triangleq dMN + dA_{max}^2 N \quad (31)$$
$$C_X \triangleq d\sum_{m=1}^{M}(1+\rho_m^2) \quad (32)$$

Using iterated expectations, we have the following:

$$\mathbb{E}\left\{\sum_{n=1}^{N} U_n(t-d)\sum_{m=1}^{M}\mu_{nm}^{STAT}(t)S_m(t)\right\} =$$
$$\mathbb{E}\left\{\sum_{n=1}^{N} U_n(t-d)\cdot\mathbb{E}\left\{\sum_{m=1}^{M}\mu_{nm}^{STAT}(t)S_m(t)|\mathcal{T}(t-d)\right\}\right\} \quad (33)$$

$$\mathbb{E}\left\{\sum_{m=1}^{M} X_m(t-d)(\rho_m 1_m(t) - \hat{C}_m^{STAT}(t))\right\} =$$
$$\mathbb{E}\left\{\sum_{m=1}^{M} X_m(t-d)\cdot\mathbb{E}\left\{\rho_m 1_m(t) - \hat{C}_m^{STAT}(t)|\mathcal{T}(t-d)\right\}\right\} \quad (34)$$

where $\mathcal{T}(t-d) = (\boldsymbol{H}(t-d), \chi(t-d), \boldsymbol{Q}(t-d))$ represents the composite system state at time $(t-d)$ and includes the topology state and queue backlogs.

By the Markovian property of the $\boldsymbol{H}(t), \chi(t)$ (and therefore $\boldsymbol{P}(t)$) processes, any functionals of these states converge exponentially fast to their steady state values (this is formalized in Appendix B). Since the policy $STAT$ makes control decisions only as a function of $\boldsymbol{P}(t)$ and $\boldsymbol{H}(t)$, the resulting allocations are functionals of these Markovian processes. Thus, there exist positive constants α_1, α_2 and $0 < \gamma_1, \gamma_2 < 1$ such that:

$$\mathbb{E}\left\{\sum_{m=1}^{M} \mu_{nm}^{STAT}(t)S_m(t)|\mathcal{T}(t-d)\right\} \geq \mu_n^{STAT} - \alpha_1\gamma_1^d$$
$$\mathbb{E}\left\{\rho_m 1_m(t) - \hat{C}_m^{STAT}(t)|\mathcal{T}(t-d)\right\} \geq \rho_m\nu_m - \hat{c}_m^{STAT} - \alpha_2\gamma_2^d$$

where $\mu_n^{STAT}, \hat{c}_m^{STAT}$ are the steady state values as defined in $(23), (24)$. Using these, the above can be written as:

$$\mathbb{E}\left\{\sum_{m=1}^{M} \mu_{nm}^{STAT}(t)S_m(t)|\mathcal{T}(t-d)\right\} \geq r_n^* - \alpha_1\gamma_1^d \qquad (35)$$
$$\mathbb{E}\left\{\rho_m 1_m(t) - \hat{C}_m^{STAT}(t)|\mathcal{T}(t-d)\right\} \geq -\alpha_2\gamma_2^d \qquad (36)$$

Thus, using $(35), (36)$ in $(33), (34)$, inequality (30) can be expressed as:

$$\Delta^{CNC}(t) - V\mathbb{E}\left\{\sum_{n=1}^{N}\theta_n R_n^{CNC}(t)\right\} \leq B + C_U + C_X$$
$$+ \mathbb{E}\left\{\sum_{n=1}^{N} U_n(t-d)\alpha_1\gamma_1^d\right\} + \mathbb{E}\left\{\sum_{m=1}^{M} X_m(t-d)\alpha_2\gamma_2^d\right\}$$
$$- V\sum_{n=1}^{N}\theta_n r_n^* \;\leq\; B + C_U + C_X + NU_{max}\alpha_1\gamma_1^d$$
$$+ MX_{max}\alpha_2\gamma_2^d - V\sum_{n=1}^{N}\theta_n r_n^*$$

The last step follows from the bounds on $U_n(t-d)$ and $X_m(t-d)$ established in (12) and (14).

Define $d_1 = \frac{\log(\alpha_1 U_{max})}{\log(1/\gamma_1)}$, $d_2 = \frac{\log(\alpha_2 X_{max})}{\log(1/\gamma_2)}$. Then choosing $d = \max(d_1, d_2)$, we have:

$$\Delta^{CNC}(t) - V\mathbb{E}\left\{\sum_{n=1}^{N}\theta_n R_n^{CNC}(t)\right\} \leq B + C_U + C_X$$
$$+ N + M - V\sum_{n=1}^{N}\theta_n r_n^* \qquad (37)$$

Since U_{max} and X_{max} are $O(V)$, we have $d \sim O(\log V)$.

Appendix B
Convergence of Markov Chains

Let $Z(t)$ be a finite state, discrete time ergodic Markov Chain. Let $\mathcal{S}$ denote its state space and let $\{\pi_i\}_{i\in\mathcal{S}}$ be the steady state probability distribution. Then, for all integers $d \geq 0$, there exist constants α, γ such that:

$$|Pr\{Z(t) = j|Z(t-d) = i\} - \pi_j| \leq \alpha\gamma^d \qquad (38)$$

where $\alpha \geq 0$ and $0 < \gamma < 1$. This implies that the Markov Chain converges to its steady state probability distribution exponentially fast (see [23]).

Let $f(Z(t))$ be a positive random function of $Z(t)$ (negative case can be treated similarly). Define $\overline{f} = \sum_{j\in\mathcal{S}}\pi_j m_j$ where $m_j \triangleq \mathbb{E}\{f(Z(t))|Z(t) = j\}$. Then:

$$\mathbb{E}\{f(Z(t))|Z(t-d) = i\}$$
$$= \sum_{j\in\mathcal{S}}\mathbb{E}\{f(Z(t))|Z(t) = j\}\, Pr\{Z(t) = j|Z(t-d) = i\}$$
$$\leq \sum_{j\in\mathcal{S}} m_j(\pi_j + \alpha\gamma^d) \qquad \text{(using(38))}$$
$$\leq \overline{f} + s m_{max}\alpha\gamma^d$$

where $m_{max} \triangleq \max_{j\in\mathcal{S}} m_j$ and $s = \text{card}\{\mathcal{S}\}$. This shows that functionals of the states of a finite state ergodic Markov Chain converge to their steady state value exponentially fast.

Appendix C
On Greedy Maximal Weight Matchings

Here, we prove property (28) for Greedy Maximal Weight Matchings (GMM) on a weighted graph. While we need this property to hold only for bipartite graphs, it is true in general for arbitrary graphs with non-negative weights.

Let $G = (V, E)$ be a graph with vertices V and edges E. Let w_e denote the weight of an edge $e \in E$. We assume that $w_e \geq 0\ \forall e \in E$. Let $C^{MWM}(G)$ denote the value of the Maximum Weight Match on G and let n be its size. Also, let $C^{GMM}(G)$ denote the value of a Greedy Maximal Weight Match on G. Note that the size of any Greedy Maximal Weight Match must be at least $n/2$. This is true because GMMs have the maximal property, and any maximal match has a size that is at least a factor of 2 away from the size of any other maximal match. We have the following:

Claim: $C^{MWM}(G) \leq 2C^{GMM}(G)$

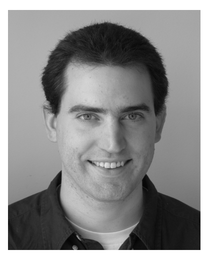
Proof: Suppose w_1 is the weight of the first edge e_1 that is chosen by the greedy procedure (as described in Sec. 6) while constructing a Greedy Maximal Weight Match on G. Then we know that w_1 is also the maximum edge weight in G. Once e_1 is chosen, all edges that share a common vertex with it are labeled "inactive" and are not considered for addition into the match. This means that at most 2 edges of the Maximum Weight Match may be labeled inactive. Further, the sum of their weights cannot exceed $2w_1$. The other $(n-2)$ or more edges of the Maximum Weight Match are candidates for selection during the next iteration of the greedy procedure. This argument can be repeated for each of the first $n/2$ iterations of the greedy procedure and yields

$$C^{MWM}(G) \leq 2\sum_{i=1}^{n/2} w_i \leq 2C^{GMM}(G)$$

REFERENCES

[1] R. Urgaonkar and M. J. Neely. Opportunistic scheduling with reliability guarantees in cognitive radio networks. *Proc. IEEE INFOCOM*, April 2008.
[2] J. Mitola. Software radios: wireless architecture for the 21st century. *John Wiley & Sons Inc*, 2000.
[3] I. F. Akyildiz, W-Y Lee, M. C. Vuran and S. Mohanty. NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey. *Computer Networks Journal (Elsevier)*, vol. 50, pp. 2127-2159, Sept. 2006.
[4] Q. Zhao and B.M. Sadler. A survey of dynamic spectrum access: signal processing, networking, and regulatory policy. *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 79-89, May 2007.
[5] C. Peng, H. Zheng, and B.Y. Zhao. Utilization and fairness in spectrum assignment for opportunistic spectrum access. *ACM/Springer MONET*, vol. 11, issue 4, Aug. 2006.
[6] L. Cao, H. Zheng. Distributed spectrum allocation via local bargaining. *Proc. IEEE SECON*, Sept. 2005.
[7] W. Wang, X. Liu and H. Xiao. Exploring opportunistic spectrum availability in wireless communication networks. *Proc. IEEE VTC*, Sept. 2005.
[8] Y.T. Hou, Y. Shi, and H.D. Sherali. Optimal spectrum sharing for multi-hop software defined radio networks. *Proc. IEEE INFOCOM*, May 2007.
[9] Y. Yuan, P. Bahl, R. Chandra, T. Moscibroda, S. Narlanka, and Y. Wu. Allocating dynamic time-spectrum blocks in cognitive radio networks. *Proc. ACM MobiHoc*, Sept. 2007.
[10] Yi Shi, Thomas Hou. A distributed optimization algorithm for multi-hop cognitive radio networks. *Proc. IEEE INFOCOM*, April 2008.
[11] Y. Chen, Q. Zhao, and A. Swami. Joint design and separation principle for opportunistic spectrum access in the presence of sensing errors. *IEEE Transactions on Information Theory*, vol. 54, no. 5, pp. 2053-2071, May 2008
[12] Q. Zhao, L. Tong, A. Swami, and Y. Chen. Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: A POMDP framework. *IEEE Journal on Selected Areas in Communications (JSAC): Special Issue on Adaptive, Spectrum Agile and Cognitive Wireles Networks*, vol. 25, no. 3, pp. 589-600, April 2007.
[13] Senhua Huang, Xin Liu, and Zhi Ding. Opportunistic spectrum access in cognitive radio networks. *Proc. IEEE INFOCOM*, April 2008.
[14] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transacations on Automatic Control*, vol. 37, no. 12, pp. 1936-1949, Dec. 1992.
[15] M. J. Neely, E. Modiano, and C. E Rohrs. Dynamic power allocation and routing for time varying wireless networks. *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 89-103, Jan. 2005.
[16] M. J. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. *Proc. IEEE INFOCOM*, March 2005.
[17] M. J. Neely. Energy optimal control for time varying wireless networks. *IEEE Transactions on Information Theory*, vol. 52, no. 7, July 2006.
[18] L. Georgiadis, M. J. Neely, L. Tassiulas. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1-149, 2006.
[19] M. J. Neely and R. Urgaonkar. Cross layer adaptive control for wireless mesh networks. *Ad Hoc Networks (Elsevier)*, vol. 5, no. 6, pp. 719-743, Aug. 2007.
[20] X. Lin and N. B. Shroff. The impact of imperfect scheduling on cross-layer rate control in wireless networks. *Proc. IEEE INFOCOM*, March 2005.
[21] P. Chaporkar, K. Kar, and S. Sarkar. Throughput guarantees through maximal scheduling in wireless networks. *Proc. of 43rd Allerton Conference*, Sept. 2005.
[22] X. Wu and R. Srikant. Bounds on the capacity region of multi-hop wireless networks under distributed greedy scheduling. *Proc. IEEE INFOCOM*, April 2006.
[23] S. Ross. Stochastic Processes. *John Wiley & Sons, Inc.*, New York, 1996.

Rahul Urgaonkar (SM '05) obtained the B.Tech. degree in Electrical Engineering from the Indian Institute of Technology (IIT) Bombay in 2002 and the M.S. degree in Electrical Engineering from the University of Southern California, Los Angeles in 2005. He is currently a PhD student in Electrical Engineering at USC. His research interests are in the areas of stochastic network optimization, resource allocation, and scheduling in next generation wireless networks.



Michael J. Neely (M '01) received B.S. degrees in both Electrical Engineering and Mathematics from the University of Maryland, College Park, in 1997. He was then awarded a Department of Defense NDSEG Fellowship for graduate study at the Massachusetts Institute of Technology, where he received an M.S. degree in 1999 and a Ph.D. in 2003, both in Electrical Engineering. In 2004 he joined the faculty of the Electrical Engineering Department at the University of Southern California, where he is currently an Assistant Professor. His research is in the area of stochastic network optimization for wireless and ad-hoc mobile networks.