Multicast Scheduling with Cooperation and Network Coding in Cognitive Radio Networks

Jin Jin, Hong Xu, Baochun Li
Department of Electrical and Computer Engineering
University of Toronto
{jinjin, henryxu, bli}@eecg.toronto.edu

***Abstract*—Cognitive Radio Networks (CRNs) have recently emerged as a promising technology to improve spectrum utilization by allowing secondary users to dynamically access idle primary channels. As progress are made and computationally powerful wireless devices are proliferated, there is a compelling need of enabling multicast services for secondary users. Thus, it is crucial to design an efficient multicast scheduling protocol in CRNs. However, state-of-the-art multicast scheduling protocols are not well designed for CRNs. First, due to primary channel dynamics and user mobility, there may not exist commonly available channels for secondary users, which inevitably makes the multicast scheduling infeasible. Second, the potential benefits provided by user and channel diversities are overlooked, which leads to under-utilization of the scarce wireless bandwidth.**

In this paper, we present an optimization framework for multicast scheduling in CRNs, by fully embracing its characteristics. In this framework, base station multicasts data to a *subset* of secondary users first by carefully tuning the power. Concurrently, secondary users opportunistically perform cooperative transmissions using *locally* idle primary channels, in order to mitigate multicast loss and delay effects. Network coding is adopted during the transmissions to reduce overhead and perform error control and recovery. We jointly consider important design factors in our scheduling protocols, including power control, relay assignment, buffer management, dynamic spectrum access, primary user protection, and fairness. We also incorporate user, channel, and cooperative diversities. Two forms of multicast scheduling protocols in CRNs are proposed accordingly: (i) a *greedy* protocol based on centralized optimization; (ii) an *online* protocol based on stochastic optimization in both centralized and decentralized manners. With rigorous analysis based on Lyapunov optimization, we provide closed-form bounds to characterize the performance of our protocols, in terms of the interference to primary users and throughput utility of secondary users. With extensive simulations, we show that our proposed protocols can significantly improve the multicast performance in CRNs.

I. Introduction

Cognitive radio networks (CRNs) have emerged as a promising technology to improve spectrum utilization, by allowing dynamic spectrum access (DSA). With the proliferation of powerful cognitive wireless devices, as well as the surge of the demand on service varieties and qualities, there is a compelling need for enabling multicast services in CRNs to further harvest its potential. Thus, it is crucial to design an efficient multicast scheduling protocol in CRNs.

Existing multicast scheduling protocols are hobbled by the holdover from cellular networks: their insistence on using a commonly available channel for all subscribers. Essentially, multicast is performed in a *single-hop* fashion. This works well in existing cellular networks, where channel assignment is centrally managed by BS. Such strategy, however, does not fit to CRNs, as channel availability on secondary users (SUs) is *opportunistic* and highly *dynamic* due to the bursty usage by primary users (PUs) and SU mobility. Fig. 1 shows an illustrative example where a CRN consisting of four PUs, six SUs, and one base station (BS) that is responsible for multicasting data to SUs. PUs can communicate with their respective access points over their own licensed channels. In contrast, BS and SUs do not have such resources and can only opportunistically use the idle spectrum for multicast services. According to convectional scheduling protocols, multicast on a certain channel is infeasible if the channel is occupied by any PU to avoid the interference to PUs, as BS is designed to use full power to multicast data globally to cover all SUs in one hop (represented by the outer solid circle from BS in Fig. 1). Obviously, current protocols are missing the bulk of opportunities to exploit the spectrum holes.

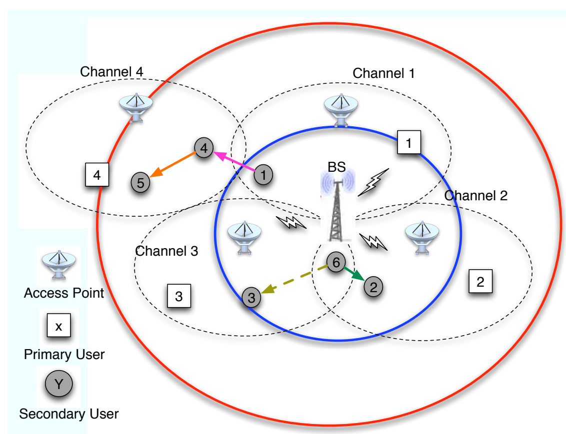

Fig. 1. An illustrative example on multicast scheduling in CRNs.

In this paper, we present a novel multicast scheduling framework to optimize multicast performance by efficiently utilizing channel resources. Rather than confining to the single-hop tradition, we advocate that multicast should be performed in a *multi-hop* fashion, with power control at BS and cooperative transmissions among SUs. The key observation is that, the set of accessible channels for different SUs are different depending on their locations. For example in Fig. 1, SU 1 can use channel 2, 3, and 4 since it only resides in the service area of PU 1. With multi-hop multicasting, the requirement of finding a

This work is supported by NSERC Discovery, CRD and Strategic Grants (RGPIN 238994-06, CRDPJ 379623-08, STPGP 364910-08).

globally available channel for all SUs is relaxed. BS and SUs only need to use channels that are *locally* accessible within one hop, providing abundant transmission opportunities with channel reuse and thus largely improving the performance.

Specifically, in our framework, BS carefully tunes the multicast transmission power to feasibly transmit data to a subset of SUs. Concurrently, to mitigate multicast loss and delay on SUs who can not get data directly from BS, cooperative communication is applied via locally available channels. Fig. 1 shows the intuition. When all channels are occupied by PUs, BS uses a smaller power to multicast data in the consequentially smaller region (the inner circle from BS in Fig. 1). In the multicast area, channel 4 is used for multicasting as it is available for every SU involved. At the same time, SU 1 helps SU 4 through cooperative transmission via channel 2 which is available for both of them. Further, SU 4 uses channel 3 to coordinately help SU 5, who can not get data from any connection. Although there may exist commonly available channels for all SUs in the whole service area (some PUs are idle), such power control and cooperative communication are able to create more opportunities for using spectrum holes, achieving higher channel utilization.

Cooperative communication can be performed even within the multicast area, as long as the channels other than those being used by BS for multicast are commonly available for the involved SUs. Such cooperation is beneficial with diversity gains as different downlink SUs experience different channel conditions, especially when user mobility is considered. In the example, SU 2 and SU 3 can be served by BS as well as SU 6 at the same time, who is much closer to BS. Intuitively, wireless channels are effectively utilized by exploiting user, channel, and cooperative diversities.

The bad news, however, is that it is challenging to schedule transmissions in a cooperative fashion. SUs that are assigned as relays do not have sufficient knowledge on which packets their neighbors need. Blindly "pushing" packets that are not useful to other SUs will incur a substantial degree of overhead and thus be detrimental for data recovery with losses, leading to dramatic degradation of multicast performance. To address these challenges, we propose to employ *network coding* in the transmission, which has emerged as one of the most promising information theoretic approaches to improve throughput performance, especially in wireless networks [1]. With network coding, all packets are encoded with random linear codes, and all coded data blocks could be considered equally useful and innovative. With the data fully mixed, relays can freely "push" innovative blocks, and receivers only need to "hold" a "bucket" and collect all useful data without dictating which packet is from which source. Further, we impose structure in the coefficients to encode data [2], with which the receivers are able to partially decode the important data even with a subset of all blocks. Intuitively, network coding can help to mitigate the overhead significantly and effectively perform error control and recovery.

The salient highlight of our contributions in this paper is a novel CR multicast scheduling framework to exploit all the potential benefits above, considering important cross-layer design factors including BS power control, relay assignment, cooperative communication, QoS guarantee, dynamic spectrum access, licensed user protection, buffer management, and resource allocation. Our contributions are three fold: (i) We propose a *greedy* centralized multicast scheduling. (ii) We design an *online* multicast scheduling protocol based on stochastic Lyapunov optimization, and characterize their performance guarantees in terms of throughput utility and interference to PUs. (iii) We develop an efficient scheme for implementing channel allocation in our online multicast scheduling based on maximum weighted bipartite matching algorithm. It can be performed in both centralized and distributed fashion which is practical in realistic systems. To the best of our knowledge, this is the first work studying cooperative multicast scheduling in CRNs. Our proposed protocols are analyzed theoretically and evaluated via extensive simulations which show substantial performance improvement.

The remainder of the paper is organized as follows. In Sec. II, we present the network model and multicast settings in CRNs. In Sec. III, we present our greedy centralized multicast scheduling protocol. In Sec. IV, we describe our realistic multicast scheduling protocol based on stochastic optimization with performance analysis. We conduct extensive simulations to evaluate the performance of our proposed protocols in realistic CRN scenarios in Sec. V. In Sec. VI, we review related work of CRNs. Finally, we conclude our paper in Sec. VII.

II. Models on Multicast Scheduling Framework

We consider a CRN consisting of C PUs, N SUs and one BS in the same fashion as in Fig. 1. Each PU has a unique licensed channel to communicate with its access point, and all C channels are orthogonal supported by OFDMA. PUs are static with fixed positions, while SUs can be mobile and opportunistically utilize the idle spectrum. Such opportunities are commonly called "spectrum holes." The entire network operates in a time-slotted fashion, where channel conditions and user actions remain the same during a given time slot, and vary independently from one time slot to another.

Let $\mathbf{S(t)} = \{S_c(t)\}_C$ represent the channel states on each time slot t, where $S_c(t)$ is a binary value capturing the channel availability. $S_c(t) = 0$ means channel c is occupied by PU c. Otherwise, $S_c(t) = 1$. We assume the channel availability state process $\mathbf{S(t)}$ evolves according to a finite state ergodic Markov chain. Within a time slot, a SU can access a subset of the licensed channels potentially depending on its current location. This information is concisely represented by a binary channel accessibility matrix $\mathbf{H(t)} = \{h_n^c(t)\}_{N\times C}$ where:

$$h_n^c(t) = \begin{cases} 1 & \text{If SU } n \text{ can access channel } c \\ 0 & \text{Otherwise} \end{cases}$$

Note the mobility process of SUs is independent of channel availability. Channel availability information for SUs can be described by a probability vector $\mathbf{P(t)} = \{P_c(t)\}_C$ where $P_c(t)$ is the probability that channel c is available. This information can be obtained via spectrum sensing or prediction according to traffic statistics. Assume BS can precisely sense the spectrum with ultra-sensitive CRs. Intuitively, the closer $\mathbf{P(t)}$ is to $\mathbf{S(t)}$ (better techniques employed), the smaller interference that can be potentially generated to PUs.

BS does not have the authority to access licensed spectrum and opportunistically utilizes "spectrum holes" to multicast data to SUs on available channels with tunable power. In the multicast sessions, BS holds all the original data, and separates the data into segments. A data segment is further divided into n *blocks* with fixed size. We can easily compute the number of blocks in one segment if the segment size is pre-determined. The BS further separates these blocks into a series of m groups, $X_1, \cdots, X_m$ ($X_1 \subseteq \cdots \subseteq X_m$), according to data priority, where X_1 is the most important set of data, followed by X_2, *etc*. Accordingly, the data blocks are encoded by *structured network coding* [2] by imposing structure in the coefficients according to data priority. Random coefficients are embedded in the transmitted blocks used for decoding at the receivers. Essentially, each coded block is a linear combination of a set of original blocks, and all linearly independent blocks are equally innovative. In this way, a virtually unlimited number of coded blocks can be generated, referred to as *rateless* property.

Evolved in the multicast service, all SUs not only receive data directly from BS but also get help from others if cooperation opportunities exist. When serving as a relay, SU encodes all the received blocks and sends the recoded blocks which are still linear combinations of the original blocks [2]. According to the property of structured network coding, SUs can have a high probability of decoding the data groups with high priorities using *Gaussian Elimination*, even with only a portion of the whole data [2]. To support cooperative transmission, SUs are equipped with multiple radios which can perform concurrent transmissions via separate channels. Note only a small number of radios are required which is practical in realistic systems. This is because the distribution of SUs is sparse and dynamic, and thus the probability that multiple SUs are within the interference regions of each other is very low.

Essentially, BS tunes the power to reach a limited number of SUs, and SUs help with each other in a local neighborhood. The multicast data thus are propagated via multi-hop dissemination with network coding efficiently. The objective of our multicast scheduling is to find the optimal power control policy on BS, as well as the most efficient cooperative communication schedule, to maximize the aggregate throughput on all SUs under a fairness criteria. To achieve this objective and meet all the requirements described above, there are a number of challenges including but not only limited to the following:

- ▷ How to design the scheduling protocol to jointly optimize the power control and cooperative communication?
- ▷ Can the optimized scheduling protocol be practically feasible and be implemented in a distributed fashion?
- ▷ Can PUs be effectively protected in CRNs?

Our responses to these challenges constitute the flow of presentation in this paper.

III. Greedy Cooperative Scheduling

In this section, we present a greedy centralized multicast scheduling protocol to perform optimization at each time slot, so that the overall performance is optimized in the long term [3]. The problem is trivially decomposed to each time slot.

A. Centralized Optimization Framework

We consider proportional fairness in the centralized optimization framework, which is able to strike a good balance between utilization and fairness [3]. The objective can be stated as follows:

$$\max_{\mathbf{P_{BS}},\mathbf{\Theta}} \quad \sum_{n=1}^{N} \frac{U_n}{\overline{r}_n} \tag{1}$$

U_n represents throughput on SU n. $\overline{r}_n$ is the average throughput that SU n obtains over previous time slots, and it brings the proportional fairness to the objective. $\mathbf{P_{BS}} = \{P^c_{BS}\}_C$ denotes the multicast power used on each channel from BS. $\mathbf{\Theta} = \{\mu^c_{mn}\}_{NN\times C}$ is the set of feasible channel assignments, where μ^c_{mn} is the binary function capturing the assignment of channel c to the cooperative transmission link from SU m to SU n. $\mathbf{P_{BS}}$ and $\mathbf{\Theta}$ represent the power control and resource allocation policies which we seek to optimize and are the cornerstones of our multicast scheduling protocol. According to the network model, we have the following constraints:

$$P^c_{BS} \leq P^c_{max} \qquad \forall c \tag{2}$$

$$P^c_{BS} \cdot g^c \cdot S_c \leq \beta \qquad \forall c \tag{3}$$

$$\mu^c_{mn} \leq h^c_m, \mu^c_{mn} \leq h^c_n \qquad \forall m,n,c \tag{4}$$

$$\mu^c_{mn} \leq l^c_m, \mu^c_{mn} \leq l^c_n \qquad \forall m,n,c \tag{5}$$

$$0 \leq \sum_{m=1}^{N} \mu^c_{mn} \leq 1 \qquad \forall n,c \tag{6}$$

$$0 \leq \sum_{c=1}^{C} \mu^c_{mn} \leq 1 \qquad \forall m,n \tag{7}$$

$$0 \leq \sum_{m=1}^{N} \mu^c_{mn} + \sum_{m'=1}^{N} \mu^c_{nm'} \leq 1 \qquad \forall n,c \tag{8}$$

Inequalities (2) and (3) describe the constraints on power control. (2) shows that the power used on each channel has an upper bound P^c_{max}. To avoid interference to PUs, the multicast power received by PUs on each channel should not exceed the tolerant level β, if the corresponding channel is being used. (3) expresses these constraints, where $g^c \in (0,1]$ is the propagation gain from BS to PU c.

Inequalities (4) and (5) represent the constraints on channel availability for cooperative communication. (4) shows that cooperative communication is constrained by the channel availability on each SU represented by $\mathbf{H(t)}$. (5) shows the constraint imposed by the channel availability with regards to multicast transmission from BS. Similar to $\mathbf{H(t)}$, we use $\mathbf{L(t)} = \{l^c_n(t)\}_{N\times C}$ to capture this information:

$$l^c_n = \begin{cases} 1 & \text{If } P^c_{BS} \cdot g^c_n \leq \gamma \\ 0 & \text{Otherwise} \end{cases} \tag{9}$$

The definition of $\mathbf{L(t)}$ indicates whenever the multicast power received by a SU exceeds the threshold γ on a channel, this SU can not use this channel for cooperative transmission. g_n^c is the propagation gain from BS to SU n on channel c.

Inequalities (6)-(8) capture the constraints regarding the avoidance of potential interference generated by cooperative communication. (6) shows that one SU can not be helped by multiple SUs simultaneously via the same channel. To take full advantage of the user diversity and channel reuse, we constrain one cooperative transmission link can be allocated with at most one channel in order to encourage more SUs to participate in cooperative communication, represented by (7). (8) indicates the incoming and outgoing transmissions on each SU can not be performed on the same channel.

Now, we are ready to calculate SU throughput:

$$U_n = \sum_{c=1}^{C} BW \cdot \log_2(1 + \frac{P_{BS}^c \cdot g_n^c}{N_n^c}) + \sum_{c=1}^{C}\sum_{m=1}^{N} \mu_{mn}^c \omega_{mn}^c \quad (10)$$

$$\omega_{mn}^c = \min\{\omega_{max}, \max\{0, \frac{B_m - B_n}{T}\}\} \quad (11)$$

BW and N_n^c denote channel bandwidth and noise respectively. ω_{mn}^c represents the achievable cooperative transmission rate from SU m to SU n on channel c with an upper bound ω_{max} due to SU power constraint. It is also limited by the amount of innovative data that SU m is able to contribute to SU n. As network coding is employed and the packets are fully random, we use $(B_m - B_n)/T$ to represent this information, where B_m denotes the amount of innovative data buffered at SU m, and B_n indicates the same information at SU n. T is the duration of one time slot, and we can set $T = 1$ without loss of generality. (11) fully captures all the constraints regarding ω_{mn}^c, and we set the buffer of each SU is sufficiently large to store at least one data segment.

Overall, the cooperative multicast scheduling in CRNs can be formulated as a centralized optimization problem with the objective (1), subject to constraints (2)-(11).

B. Optimization Solution

The formulated problem above is a non-linear integer programming (NIP) problem, which is NP-hard. In the following, we discuss problem linearization and use *branch-and-bound* algorithm to solve it with **polynomial** complexity.

We first relax the binary indicators μ_{mn}^c into fractional values in $[0, 1]$. Besides, it is important to linearize constraints (10) and (9) which are not convex. To address this challenge, we adopt Reformulation-Linearization Technique (RLT) [4], which is used to produce LP relaxations for an underlying nonlinear, non-convex programming problem by providing a tight upper bound for a maximization problem. According to RLT, we linearize the logarithmic relationship in (10) using polyhedral outer approximation with several tangential supports [4]. The intuition on the approximation of $y = \log(x)$ is shown in Fig. 2.

(9) contains a nonlinear relationship that couples power control and channel allocation variables together. Essentially, l_n^c in (9) constrains the value of μ_{mn}^c, which is already relaxed into the fractional variable in $[0, 1]$. Accordingly, we can also relax l_n^c into the range of $[0, 1]$, and the value should depend on multicast power P_{BS}^c. Intuitively, the larger power sensed by SUs, the more likely the channel is occupied and can not be allocated for cooperative transmission, and thus l_n^c should be more like approaching to 0, and vice versa. If the sensed power is just around the threshold γ/g_n^c, it will be very ambiguous to determine the availability of the corresponding channels. In this case, we can relax the value of l_n^c to be around 0.5. According to the analysis above, we use the following linear constraints for approximation, the intuition of which is shown in Fig. 3.

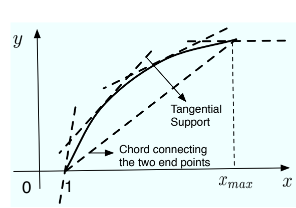

Fig. 2. Intuition on linearization for logarithmic relationship, which uses a four-point tangential approximation.

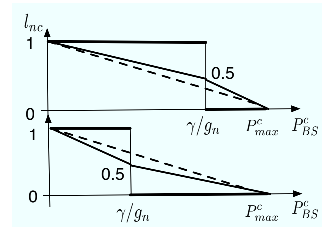

Fig. 3. Intuition on linearization of constraint (9).

$$\left.\begin{array}{l} l_n^c \geq -\frac{1}{P_{max}^c} P_{BS}^c + 1 \\ l_n^c \leq -\frac{1}{2\gamma/g_n^c} P_{BS}^c + 1 \\ l_n^c \leq -\frac{0.5}{P_{max}^c - \gamma/g_n^c}(P_{BS}^c - P_{max}^c) \end{array}\right\} \text{if } \frac{P_{max}^c}{2} \leq \gamma/g_n^c < P_{max}^c$$

$$\left.\begin{array}{l} l_n^c \leq -\frac{1}{P_{max}^c} P_{BS}^c + 1 \\ l_n^c \geq -\frac{1}{2\gamma/g_n^c} P_{BS}^c + 1 \\ l_n^c \geq -\frac{0.5}{P_{max}^c - \gamma/g_n^c}(P_{BS}^c - P_{max}^c) \end{array}\right\} \text{if } 0 < \gamma/g_n^c < \frac{P_{max}^c}{2}$$

With linear relaxation, we now apply *branch-and-bound algorithm* in our problem. Under this approach, we aim to provide a $(1 - \epsilon)$ **optimal** solution, where ϵ is a small positive constant reflecting our desired accuracy in the final solution. We first solve the LP relaxation and get the fractional solutions $\hat{\mu}_{mn}^c$ and corresponding upper bound (UB) of the objective. With this solution, we then use a local search algorithm to find a feasible lower bound (LB) of the objective. In our problem, we use randomized rounding on $\hat{\mu}_{mn}^c$ to get LB, considering the problem feasibility at the same time. If $LB \geq (1 - \epsilon)UB$, then we get the overall optimal solution. If not, we have to close this gap by obtaining a tighter linear relaxation. This could be achieved by selecting a partition variable with the maximum relaxation errors and dividing its value set into two sets by its value in the relaxation solution. In our problem, we choose a $\hat{\mu}_{mn}^c$ with maximum value of relaxation error captured by $\min\{\hat{\mu}_{mn}^c, 1 - \hat{\mu}_{mn}^c\}$, and divide the original problem into two subproblems with μ_{mn}^c equal to 0 and 1 respectively.

For the two subproblems, we again solve the LP relaxation and run local search to get their bounds: (UB_2, LB_2) and (UB_3, LB_3). We update $UB = \max\{UB_2, UB_3\}$ and $LB = \max\{LB_2, LB_3\}$. Then, if $LB \geq (1 - \epsilon)UB$, the entire procedure is terminated. Otherwise, we will iteratively repeat the problem dividing until we get the optimal solution. During this process, we remove any subproblem i when $(1-\epsilon)UB_i \leq LB$. It has been shown that under general conditions, a branch-and-bound procedure always converges efficiently.

IV. Multicast Scheduling With Stochastic Lyapunov Optimization

The centralized protocol has some implementation problems. It requires network condition information, such as propagation gain on SUs, which cannot be precisely estimated in practical systems, especially when SUs are mobile. Moreover, branch-and-bound algorithm does not provide performance guarantees over time and may have high complexity due to LP relaxation and search. These problems may cause the centralized scheduling infeasible or inefficient for practical use.

To address these challenges, we design an online scheduling, including power control and channel allocation policies, based on a stochastic optimization with realistic system settings *without* the requirement of global knowledge of channel conditions. Especially, we design a distributed algorithm for channel allocation which can be performed efficiently in realistic systems *without* linear approximation and search efforts. Via rigorous analysis, we show that our stochastic multicast scheduling is able to provide explicit over-time performance guarantees.

A. Stochastic Network Model with Practical Settings

Each SU involved in the multicast session collects data and maintains a data buffer to store the data. As we employ structured network coding, each SU is able to perform *partially* decoding with a limited number of encoded blocks. The more data a SU gets from the buffer, the more data groups can be decoded and forwarded to upper layers, if a sufficient number of groups are applied in network coding. All SUs in the network are *greedy* and take off as much data as possible from the buffer to upper layers for decoding and other processing.

Let $B_n(t)$ be the amount of data in the buffer of SU n at time slot t with upper bound as B_{max}. $R_n(t)$ represents the rate that SU n takes data off from the buffer at time slot t. Clearly, it is bounded by $\min\{B_n(t), R_{max}\}$, where R_{max} is the maximum rate that can be achieved due to the computation and bandwidth limit of SUs. Then, we have the following buffer dynamics:

$$B_n(t+1) = \max\{B_n(t) - R_n(t), 0\} + M_n(t) + \sum_{c=1}^{C}\sum_{m=1}^{N} \mu_{mn}^c(t) S_n^c(t) \omega_{mn}^c(t). \quad (12)$$

$M_n(t)$ is the multicast throughput obtained directly from BS at time slot t for SU n, and $\sum_{c=1}^{C}\sum_{m=1}^{N} \mu_{mn}^c(t) S_n^c(t) \omega_{mn}^c(t)$ is the throughput it obtains from cooperative communication. In the network, BS keeps on multicasting data, and we denote r_n as the time average throughput of SU n. Without loss of generality, we have the following:

$$r_n = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} R_n(\tau) \quad (13)$$

Let $\mathbf{r} = (r_1, \cdots, r_N)$ denote the rate vector on all SUs.

In multicast sessions, cooperative communication may generate interference to PUs. The examples are shown in Fig. 4, where SU 1 fails to detect the transmission that the access point makes to PU 1, and intends to utilize channel 1 for cooperative transmissions. Under this condition, a collision will occur and data errors on PU 1 will be generated. PU 1 may also be affected by the cooperative transmissions from SU 3 to SU 4. We define the following variables $E_c(t)$ to capture the total number of such collisions caused by cooperative transmissions for each PU:

$$E_c(t) = \sum_{m=1}^{N}\sum_{n=1}^{N} \mu_{mn}^c(t) I_m^c(t) \big(1 - S_c(t)\big) \quad (14)$$

where $I_m^c(t)$ is the binary variable indicating whether the cooperative communication issued by SU m may generate interference to PU c at time slot t. This information can be captured by each SU according to the location information (if PU c is in the transmission range of SU m, then $I_m^c(t) = 1$). It is intuitive that the more interference incurred, the more severe PUs would suffer from the packet loss. Let e_c denote the time average rate of interference for PU c:

$$e_c = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} E_c(\tau) \quad (15)$$

In the network, this interference information can be tracked using interference queues $X_c(t)$ for PUs, and all SUs are aware of these queues. The interference on each PU can not exceed a time average tolerant rate ρ_c. Thus, we have the following interference queue dynamics:

$$X_c(t+1) = \max\{X_c(t) - \rho_c, 0\} + E_c(t) \quad (16)$$

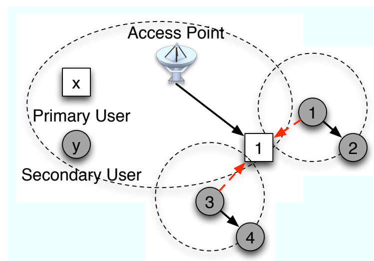

Fig. 4. Cooperative communication may generate interference to PUs.

Overall, we aim to maximize the aggregate throughput of SUs over time under a fairness criteria (consistent with the objective of centralized optimization (1)). Let $\{\theta_1, \cdots, \theta_N\}$ be a collection of positive weights, representing user priority (fairness). Then, the multicast scheduling can be stated as:

$$\max \quad \sum_{n=1}^{N} \theta_n r_n$$
$$\text{subject to:} \quad (2) - (16). \quad (17)$$

We seek to design a scheduling scheme with practical power control and resource allocation by solving this problem.

B. Optimal Online Scheduling Policies

The online scheduling includes the following two policies based on stochastic optimization:

(i) *Power Control Policy*: At each time slot t, BS chooses multicast power as the solution of the following problem:

$$\max \quad \sum_{n=1}^{N} R_n(t) M_n(t)$$
$$\text{subject to:} \quad (2), (3). \quad (18)$$

(18) indicates that the optimal power control policy on BS is to achieve the maximum aggregate throughput on all SUs with regards to direct multicast transmissions from BS without cooperative communication. With such convenience, it is easy to see that (18) can be solved by using maximum feasible power of BS to multicast data on each channel under interference constraint (2) and (3). Thus, the power control policy can be simplified as:

$$P_{BS}^c(t) = \begin{cases} P_{max}^c, & \text{if } S_c(t) = 1, \\ \min\{P_{max}^c, \beta/g^c\}, & \text{otherwise.} \end{cases}$$

(ii) *Channel Allocation Policy*: Channel resources are allocated as the solution of the following:

$$\max \sum_{n,m,c} \mu_{mn}^c(t)(\omega_{mn}^c(t)B_n(t)P_c(t) - X_c(t)I_m^c(t)(1-P_c(t)))$$
$$\text{subject to:} \quad (4)-(8),(11). \tag{19}$$

(19) is a LP and can be solved in polynomial time, though this may require centralized control (feasible and practical in cellular CRNs such as the one we work on in the paper). We also propose a distributed solution for channel allocation, and its discussion is deferred to Sec. IV-D. Note, (19) only requires buffer, interference, and capacity information, which is practically feasible. The SU data buffer and PU interference queue will be updated after observing the outcome of this allocation at the end of each time slot.

C. Performance Analysis

We now characterize the performance of our scheduling policies with the following bounds:

(i) (*Interference Performance*) Initialize $X_c(0) = 0, \forall c$. $\forall t > 0$, if $P_c(t) < 1$, set $0 < \varepsilon < 1$ and $P_c(t) \le 1 - \varepsilon$. Then the worst case interference queue backlog for all PUs is upper bounded by:

$$X_c(t) \le X_{max} \triangleq B_{max}\omega_{max}(\frac{1-\varepsilon}{\varepsilon}) + \lfloor \frac{N}{2} \rfloor \quad \forall c \tag{20}$$

Proof: $X_c(0) = 0 < X_{max}$. Now, suppose that $X_c(t) \le X_{max}$. We show the same holds for $X_c(t+1)$. First, suppose $P_c(t) = 1$. Then, there is no interference to PU c as channel c is idle. Thus, we have $X_c(t+1) \le X_{max}$ from (16) as $E_c(t) = 0$. Next, suppose $P_c(t) < 1$, and we have two cases. (*a*) $X_c(t) \le X_{max} - \lfloor \frac{N}{2} \rfloor$. $\lfloor \frac{N}{2} \rfloor$ represents the maximum number of cooperative transmission links in the network, which is also the maximum value of $E_c(t)$. Under this case, $X_c(t+1) \le X_{max}$. (*b*) $X_c(t) > X_{max} - \lfloor \frac{N}{2} \rfloor = B_{max}\omega_{max}(\frac{1-\varepsilon}{\varepsilon})$. Then, $X_c(t)\varepsilon > B_{max}\omega_{max}(1-\varepsilon)$. Further, $X_c(t)(1-P_c(t)) > B_{max}\omega_{max}P_c(t)$. If $I_m^c(t) = 1$, according to our *channel allocation* policy, $\mu_{mn}^c = 0, \forall m, n$, which means there is no cooperative communication on channel c. If $I_m^c(t) = 0$, the transmissions from all SUs can not reach PU c. Thus, no interference will be generated to it. Hence, $X_c(t+1) \le X_c(t) \le X_{max}$. Overall, (20) is proved. ■

(ii) (*Utility performance*) Initialize $B_n(0) = 0, \forall n$. The time average throughput utility achieved by our protocol is within $\tilde{B}/V$ of the optimal value:

$$\lim_{t\to\infty} \inf \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{n=1}^{N} \theta_n \mathbb{E}\{R_n(\tau)\} \ge \sum_{n=1}^{N} \theta_n r_n^* - \frac{\tilde{B}}{V} \tag{21}$$

where r_n^* is the optimal solution (maximum achievable rates) of stochastic problem (17), and $V, \tilde{B} > 0$ are constants.

We use the technique of *Stochastic Lyapunov Optimization* to prove it. Let $\mathbf{Q}(t) = (Q_1(t), \cdots, Q_K(t))$ be a vector of queue lengths for a discrete time stochastic queueing network. Let $W(\mathbf{Q})$ be any non-negative scalar valued function of the queue lengths, called a Lyapunov function. Define the *Lyapunov drift* $\Delta(t)$ as follows:

$$\Delta(t) \triangleq \mathbb{E}\{W(\mathbf{Q}(t+1)) - W(\mathbf{Q}(t))\} \tag{22}$$

The network accumulates *utility* every time slot, with bounded value. We have the stochastic process $f(t)$ to represent the utility earning during time slot t with optimal value f^*.

Theorem 1: Suppose there exist finite constants $V > 0, \tilde{B} > 0, d > 0$, and a non-negative function $W(\mathbf{Q})$ such that $\mathbb{E}\{W(\mathbf{Q}(d))\} < \infty$. For every time slot $t > d$, if the Lyapunov drift satisfies:

$$\Delta(t) - V\mathbb{E}\{f(t)\} \le \tilde{B} - Vf^* \tag{23}$$

then we have:

$$\lim_{t\to\infty} \inf \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{f(\tau)\} \ge f^* - \frac{\tilde{B}}{V} \tag{24}$$

Proof: Refer to [5]. ■

In our stochastic multicast scheduling problem, $\mathbf{Q}(t) = (B_1(t), \cdots, B_N(t), X_1(t), \cdots, X_C(t))$ including the data buffer queues and interference queues. Let $f(t) \triangleq \sum_{n=1}^{N} \theta_n R_n(t)$ be the aggregated throughput utility at each time slot according to the objective of problem (17). Thus, $f^* \triangleq \sum_{n=1}^{N} \theta_n r_n^*$. Then, we further define the Lyapunov function as follows:

$$W(\mathbf{Q}(t)) \triangleq \frac{1}{2}\Big(\sum_{n=1}^{N} \big(B_n(t)\big)^2 + \sum_{c=1}^{C} \big(X_c(t)\big)^2\Big)$$

Now, we calculate the Lyapunov drift as follows:

$$\Delta(t) \le B - \mathbb{E}\Big\{ \sum_{n=1}^{N} R_n(t)\big(B_n(t) + M_n(t)\big) + R_n(t) \sum_{c=1}^{C} \sum_{m=1}^{N} \mu_{mn}^c(t) S_c(t) \omega_{mn}^c(t) \Big\} - \mathbb{E}\Big\{ \sum_{c=1}^{C} X_c(t)(\rho_c - E_c(t)) \Big\} \tag{25}$$

where $B \triangleq \frac{1}{2}(N(B_{max} + T_{max})^2 + \sum_{c=1}^{C}(\rho_c)^2 + C)$, and T_{max} is the upper bound of the throughput on each SU at each time slot due to the buffer constraint and network capacity.

Now we subtract $V\mathbb{E}\{\sum_{n=1}^{N} \theta_n R_n(t)\}$ from both sides of the drift inequality (25) and use (14) to obtain:

$$\Delta(t) - V\mathbb{E}\Big\{ \sum_{n=1}^{N} \theta_n R_n(t) \Big\} \le B - \sum_{c=1}^{C} \rho_c \mathbb{E}\Big\{ X_c(t) \Big\}$$

$$-\mathbb{E}\Big\{\sum_{n=1}^{N}R_n(t)(B_n(t)+V\theta_n)\Big\}-\mathbb{E}\Big\{\sum_{n=1}^{N}R_n(t)M_n(t)\Big\}$$
$$-\mathbb{E}\Big\{\sum_{m,n,c}\mu_{mn}^c(t)\Big(\omega_{mn}^c(t)B_n(t)S_c(t)$$
$$-X_c(t)I_m^c(t)(1-S_c(t))\Big)\Big\} \quad (26)$$

The last two terms of inequality (26) above are exactly our scheduling policies stated in Sec. IV-B (replace $S_c(t)$ as $P_c(t)$ by considering the sensing errors; it is necessary to consider it especially in the distributed algorithms without accurate sensing). Note the direct multicast rate from BS is dominant in the aggregate throughput on SUs. Thus, we can optimize the last two terms separately although they have common constraints. It is clear to see that our online scheduling policies minimize the right side of inequality (26) over all alternate feasible scheduling policies that can be made at each time slot.

We now define the stationary, randomized policy *SR*, that chooses a feasible power control and channel allocation $M_n^{SR}(t), \mu_{mn}^{c,SR}(t)$ at every time slot as a function of only the channel state information $\mathbf{S(t)}$ and $\mathbf{P(t)}$, which will yield the following steady state values:

$$\mathbb{E}\{R_n^{SR}(t)\}=r_n^* \quad (27)$$
$$e_c^{SR}\triangleq\lim_{t\to\infty}\sum_{\tau=0}^{t-1}\mathbb{E}\{E_c^{SR}(\tau)\}\leq\rho_c \quad (28)$$

Note our online scheduling policies minimize the right side of (26) including this stationary, randomized policy [6]. Using all facts above, we can show that:

$$\Delta(t)-V\mathbb{E}\{f(t)\}\leq B-\mathbb{E}\big\{\sum_{n=1}^{N}R_n^{SR}(t)\big(B_n(t)+T_n(t)\big)\big\}$$
$$-\mathbb{E}\big\{\sum_{c=1}^{C}X_c(t)\big(\rho_c-E_c^{SR}(t)\big)\big\}-Vf^* \quad (29)$$

Finally, we get the following result by using *delayed* queue backlogs and properties of Markov process (refer to the appendix in [7] for proof):

$$\Delta(t)-V\mathbb{E}\Big\{\sum_{n=1}^{N}\theta_nR_n(t)\Big\}\leq\tilde{B}-V\sum_{n=1}^{N}\theta_nr_n^*$$

This form fits (23). Thus, applying Theorem 1 proves (21).

D. Efficient Implementation

In this section, we seek to solve channel allocation problem stated in (19), with practically efficient implementation. We observe, without considering (8), (19) can be formulated into a maximum weighted bipartite matching (WBM) problem which can be solved **optimally** with **polynomial** time complexity.

Construct a bipartite graph $A=(\Phi\times\chi,E)$. The vertices in Φ denote all the possible cooperative links (*e.g.* $(1,2)$ indicates the transmission link from SU 1 to SU 2. Note it is different from $(2,1)$, which represents the transmission link from SU 2 to SU 1). The set of channels for cooperative transmissions is denoted by the vertex set χ. The edge set E corresponds to $|\Phi|\times|\chi|$ edges connecting all possible pairs. The weight of each edge carries $w_{mn}^c=\omega_{mn}^c(t)B_n(t)P_c(t)-X_c(t)I_m^c(t)(1-P_c(t))$. Before solving WBM, we exclude all pairs connecting (m,n) in Φ and c in χ if $h_m^c(t)\cdot h_n^c(t)\cdot l_m^c(t)\cdot l_n^c(t)\leq 0$, which indicates the channel is already occupied according to constraint (4) and (5). We also exclude vertices (m,n) if SU m is not within the transmission range of SU n.

According to constraint (7), one SU can not accept cooperative transmission from multiple SUs on the same channel. Thus, we can solve the WBM problem in groups. Each group includes all links with the same destination, *e.g.*, $(1,1),(2,1),\cdots,(N,1)$. We denote the set of vertices with destination SU n as Φ_n, and $|\Phi_n|$ may be not equal to $|\chi|$. Then, we patch void vertices to Φ_n or χ to make $|\Phi_n|=|\chi|$. If an edge connects any void node, its weight is set to be zero.

Given the above graphical setup, channel allocation problem can be solved by solving WBM problems for all groups, getting all the matched pairs $((m,n),c)$. The intuition is shown in Fig. 5. Now we consider the constraint (8) which we have previously ignored. Solving the WBM problem stated above may violate this constraint if the same channel is assigned for both uplink and downlink communication on the same SU. In the network, SUs are greedy and selfish, and they always prefer incoming traffic (get help from others) rather than outgoing traffic (helping others). Thus, we allocate channels according to this policy when (8) is violated. The WBM problem can be solved in a centralized fashion using network flow algorithms [8]. To be efficiently implemented in realistic systems, we design a distributed algorithm, stated in **Algorithm 2**, based on the WBM formulation and selfish policy.

We perform a set of simulations to specifically evaluate our distributed algorithm. From the results, we observe the distributed algorithm is able to **converge** within 3 rounds in average, which is faster than centralized approach by a 20% gain, and thus suitable in real world environment. It also achieves good throughput and fairness performance, close to the optimal centralized approach for solving WBM problem, within only 7% difference. We further elaborate the evaluation in the Sec. V.

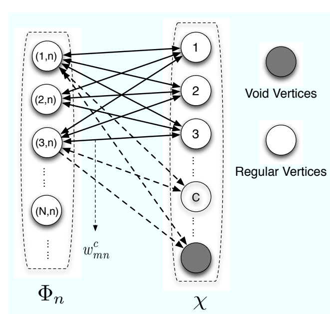

Fig. 5. Solving the channel allocation problem using maximum weighted bipartite matching algorithm.

V. Performance Evaluation

We are now ready to resort to extensive simulations to study the performance of our cooperative multicast scheduling protocols with network coding. To be realistic, practical settings

Algorithm 1 Distributed Algorithm for Channel Allocation

Each SU n carries out the following steps:

1. Senses channels at the beginning of each time slot, and get the channel availability information.
2. Broadcasts its buffer information on available channels (interference information $X_c(t)$ is known by all SUs).
3. Solves WBM $(\Phi_n \times \chi, E)$ according to channel availability, and buffer and interference information.
4. If a vertex (m, n) in Φ_n and a vertex c in χ are matched, sends *helping* requests to SU m on the channel c.
5. Collects all messages sent by its neighbors.
6. Upon receiving a *helping* request from SU m on channel c
   - if it did not send any *helping* request to other SUs on channel c (no matching on channel c in WBM), then sends an *agree* message back to SU m.
   - if it sent a request to any other SU on channel c (there is matching on channel c in WBM), then just stores this *helping* message.
7. Upon receiving an *agree* reply from SU m on channel c, it knows SU m agrees to provide help on channel c, and sends *drop* messages to all other SUs who request *helping* on channel c. The transmission from SU m to SU n is allowed, and set $\mu^c_{mn} = 1$.
8. Upon receiving a *drop* message from SU m on channel c, it knows channel c is used by SU m, and excludes the link $(m, n) \to c$ in its bipartite graph.
9. If it has no free neighbors or no available links in its bipartite matching graph, no further action is taken. Otherwise, it will repeat step (3)-(7).

TABLE I
SIMULATION PARAMETERS.

Channel Type	Rayleigh fading and AWGN
Transmitter Power (BS)	25 dBm
Transmitter Power (SU)	5 dBm
Noise Power	-129.5 dBW
Adaptive Modulation	used

of a CRN, as summarized in Table. I, are adopted according to the IEEE 802.22 draft [9]. A total of 10 PUs reside in the service area, while a number of SUs move randomly with randomly initial locations. The channel availability state evolves according to a Markov chain with symmetric transition probabilities between the ON and OFF states given by 0.5.

To evaluate the performance, we compare four multicast scheduling protocols: (i) Centralized cooperative scheduling following the design in Sec. III, referred to as “Centralized.” (ii) Our online cooperative scheduling with distributed implementation based on the design in Sec. IV, referred to as “Distributed.” (iii) Multicast scheduling with power control according to the policy stated in Sec. IV-B, but without cooperative communication, referred to as “NOCoop.” and (iv) Multicast scheduling with no power control nor cooperative communication, referred to as “NOPower,” where multicast is only performed when commonly available channels exist for all SUs in the network, and is provisioned with maximum feasible power.

We first examine the throughput performance. Fig. 6 shows

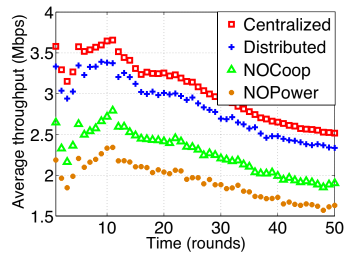

Fig. 6. Average throughput performance of all protocols in realistic CRN scenarios.

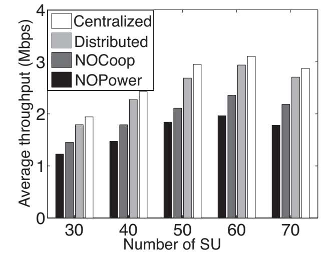

Fig. 7. Throughput performance of all protocols with different numbers of SUs, which represent the degree of possible cooperation among SUs.

the resultsfor a total of 60 SUs via a 15000-second simulation, with the algorithms running every 30 seconds. We observe that “Centralized” performs the best, and outperforms “NOCoop” and “NOPower” by 40% and 60%, respectively, on average. “Distributed” also outperforms “NOCoop” and “NOPower,” with 35% and 53% gains on average, respectively. Such a throughput advantage should be considered substantial by any standard. It coincides with our intuition that multicast scheduling with cooperative communication, power control, network coding, and other important cross-layer designs naturally fits in the design of CRNs, and is able to achieve significant throughput improvement due to its effective use of wireless spectrum. From the results, we also observe that “Centralized” and “Distributed” perform close to each other (within a 5% difference), which indicates that our decentralized scheduling based on stochastic optimization is efficient and near-optimal. Another trend to notice is that the average throughput is slowly decreasing over time. The reason is that our objective takes fairness into account, which makes the optimization favor a “slower” SU as time goes.

Next, we specifically investigate the benefits and impact of cooperative communication. Fig. 7 shows the average throughput performance as a function of the number of active SUs. Evidently, the margin that “Centralized” and “Distributed” outperform “NOCoop” and “NOPower” becomes more substantial as the number of SUs increases. This observation indicates that a larger number of SUs creates a higher degree of cooperation which is beneficial for the performance. However, when the number of SUs becomes overly large, throughput degrades since the interference effect begins to dominate.

Regarding the fairness and delay performance, we further examine the variance of the average throughput over SUs. At each time slot t, we calculate, for each SU, the average throughput over time horizon $[1, t]$, and then compute the throughput variance, which is the ratio between standard deviation of the time average throughput and the time average throughput itself. Fig. 8 plots the CDF of this metric for a total of 60 SUs in the network. Not surprisingly, both “Centralized” and “Distributed” outperform “NOCoop” and “NOPower,” which shows that our protocols are able to achieve good fairness performance in the multicast service. This result also indicates that our protocols are helpful to decrease the delay on the SUs, who do not have spectrum resources and can not get data directly from BS.

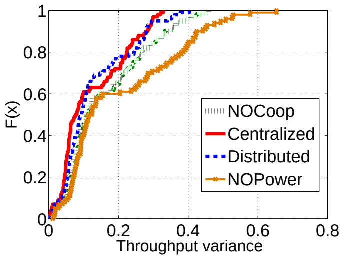

Fig. 8. CDF of all SUs with respect to throughput variance, which indicates fairness performance.

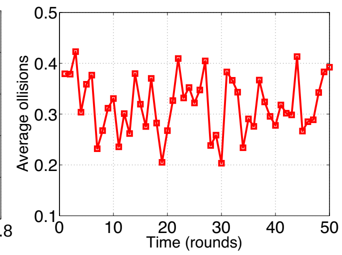

Fig. 9. Evaluation of interference on PUs, which remains stable and bounded in our protocols.

Finally, we examine the interference on PUs. Fig. 9 captures the average interference queue length on all PUs. It clearly shows that the interference level remains bounded over the long term, which is desirable in the system design.

In closing, we comment on the protocol overhead. As the BS has no constraints on energy and computational power, we are only concerned with the computation overhead at SUs. Nowadays, a mobile device such as a cellphone has sufficient memory cache and strong computation capability. According to the results in [10], random network coding performs efficiently on the iPhone family of mobile devices in a realistic P2P streaming scenario. As we studied, our proposed protocols are in low complexity, and our extensive simulation shows that proposed algorithms have an average running time of less than 1 ms (over Intel Core Duo machine running at 1.83 GHz and a memory of 2 GB), and are therefore suitable for typical WiMAX with scheduling durations of 5-10 ms.

VI. Related Work

Cognitive radio is a revolution in radio technology to efficiently utilize the wireless spectrum. IEEE 802.22 [9] is the first standardization effort to define cognitive radio and so far has drawn much research attention in both academia and industry. Dynamic spectrum access [11] is one of the key issues in CRNs and has driven most of the CR research. [5] develops an opportunistic spectrum access framework for CRNs that maximizes SU aggregate throughput. [12] and [13] study the dynamic access issues in ad hoc mode of CRNs, where scheduling and routing are jointly considered. Our work adopts similar network models with previous work for dynamic spectrum access.

Other than most concerns of previous studies, our paper focuses on multicast scheduling in CRNs, which is more challenging but has been scantly investigated so far. [14] proposes an energy-efficient multicast scheduling in CRNs, but it is still restricted in single-hop transmission mode without cross-layer designs and works in different network settings. Another important work regarding this is [15] that proposes a video multicast protocol in CRNs. Our work differs from it in several aspects. First, [15] only focuses on multicast in one cell, and assumes all SUs and PUs are within the transmission range of each other. Our protocols are tightly integrated with the design of CRNs and work in more realistic scenarios with multiple PU cells in a wide area. Second, our protocols employ power control, cooperative communication, and network coding in multicast scheduling. Third, we have cross-layer designs considering the important issues in CRNs. Last but not least, we design our protocols based on both greedy and stochastic optimization frameworks with both centralized and decentralized implementations.

VII. Concluding Remarks

In this paper, we have studied multicast scheduling in CRNs. The main challenge is due to the dynamic spectrum availability and diverse channel conditions on SUs. We propose *multi-hop* multicast protocols, tightly integrated with the design of CRNs, by employing techniques of power control, cooperative communication, and network coding. We have jointly considered primary user protection, relay assignment, QoS guarantees, and buffer management. Our protocols fully exploit multicast opportunities and incorporate user, channel, and cooperative diversities. They are designed based on a sound theoretical foundation using centralized greedy optimization and stochastic Lyapunov optimization, but not without careful considerations of the practicality, feasibility, and efficiency of implementing these solutions. With this paper, we are convinced that multicast performance can be significantly improved in CRNs with the effective use of scarce wireless spectrum, by applying power control, cooperative communication and network coding.

References

[1] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading Structure for Randomness in Wireless Opportunistic Routing," in *Proc. of ACM SIGCOMM*, 2007.

[2] X. Liu, G. Cheung, and C.-N. Chuah, "Structured Network Coding and Cooperative Local Peer-to-peer Repair for MBMS Video Streaming," in *IEEE International Workshop on Multimedia Signal Processing*, October 2008.

[3] G. Song and Y. G. Li, "Cross-layer Optimization for OFDM Wireless Network - Part I and Part II," *IEEE Transactions on Wireless Communications*, vol. 4, no. 2, March 2005.

[4] H. D. Sherali and L. Liberti, "Reformulation-Linearization Methods for Global Optimization," *Journal of Global Optimization*, 1991.

[5] R. Urgaonkar and M. J. Neely, "Opportunistic Scheduling with Reliabilitty Guarantees in Cognitive Radio Networks," in *Proc. of IEEE INFOCOM*, 2008.

[6] M. J. Neely, E. Modiano, and C. Li, "Dynamic Power Allocation and Routing for Time Varying Wireless Networks," *IEEE Journal on Selected Areas in Communications*, 2005.

[7] J. Jin, H. Xu, and B. Li, "Multicast Scheduling with Cooperation and Network Coding in Cognitive Radio Networks," University of Toronto, http://iqua.ece.toronto.edu/paper/jinjin-mCRNs.pdf, Tech. Rep., 2009.

[8] R. K. Ahuja, T. L. Magnati, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Prentic Hall, February 1993.

[9] C. Cordeiro, K. Challapali, D. Birru, and N. S. Shankar, "IEEE 802.22: The First Worldwide Wireless Standard Based on Cognitive Radios," in *Proc. of 1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN 05)*, 2005.

[10] H. Shojania and B. Li, "Random Network Coding on the iPhone: Fact or Fiction?" in *Proc. of the 19th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDA)*, 2009.

[11] Q. Zhao, S. Geirhofer, L. Tong, and B. Sadler, "Opportunistic Spectrum Access via Periodic Channel Sensing," *IEEE Transactions on Signal Processing*, 2008.

[12] Y. Shi and Y. T. Hou, "Optimal Power Control for Multi-hop Software Defined Radio Networks," in *Proc. of IEEE INFOCOM*, 2007.

[13] ——, "A Distributed Optimization Algorithm for Multi-hop Cognitive Radio Networks," in *Proc. of IEEE INFOCOM*, 2008.

[14] L. Geng, Y.-C. Liang, and F. Chin, "Network Coding for Wireless Ad Hoc Cognitive Radio Networks," in *Proc. of IEEE PIPRC*, 2007.

[15] D. Hu, S. Mao, and J. H. Reed, "On Video Multicast in Cognitive Radio Networks," in *Proc. of IEEE INFOCOM*, 2009.