This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2009 proceedings.

On Video Multicast in Cognitive Radio Networks

†Donglin Hu, †Shiwen Mao, and ‡Jeffrey H. Reed
†Dept. of Electrical and Computer Engineering, Auburn University, Auburn, AL
‡The Bradley Dept. of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA

***Abstract*— We investigate the challenging problem of enabling multicast video service in emerging cognitive radio (CR) networks. We propose a cross-layer optimization approach to multicast video in CR networks. Specifically, we model CR video multicast as an optimization problem, while considering important design factors including scalable video coding, video rate control, spectrum sensing, dynamic spectrum access, modulation, scheduling, retransmission, and primary user protection. The objective is to optimize the overall received video quality as well as achieving proportional fairness among multicast users, while keeping the interference to primary users below a prescribed threshold. Although the problem can be solved using advanced optimization techniques, we propose a sequential fixing algorithm and a greedy algorithm with low complexity and proven optimality gap. Our simulations using MPEG-4 fine grained scalability (FGS) demonstrate the efficacy and superior performance of the proposed approach as compared with an alternative equal allocation scheme.**

I. Introduction

A *cognitive radio* (CR) is a frequency-agile wireless communication device that enables *dynamic spectrum access*. The CR concept represents a significant paradigm change in spectrum regulation and utilization. Its high potential has stimulated a flurry of exciting activities in engineering, economics, and regulatory communities in searching for better spectrum management policies and techniques [1], [2]. As progress being made and understandings being gained, there is a compelling need for enabling content-rich multimedia services in CR networks to fully harvest their potential. However, since the need for efficient dynamic spectrum access has driven most of the CR research so far, multimedia services have not been the primary concern in the mainstream research in this area.

In this paper, we present a study of optimized real-time video multicast in CR networks. We consider an infrastructure-based CR network collocated with N primary networks. The availability of each channel evolves over time due to primary user transmissions. The base station of the CR network exploits the spectrum opportunities in the N channels to multicast videos to G multicast groups. We consider video because it is one of the most challenging media to deliver with stringent QoS requirements. Therefore, insights gained in solving this problem may also be helpful for solving other related problems.

In order to accommodate heterogeneous channels experienced by different users, we adopt fine grained scalability (FGS) to encode each video into a base layer and an enhancement layer [3]. With FGS, the enhancement layer can be truncated at any bit location, while the remaining bits are all useful for decoding. Therefore a user can receive a video quality commensurate to its channel condition.

We present a novel formulation of the CR video multicast system, considering important cross-layer design factors such as scalable video coding, video rate control, spectrum sensing, dynamic spectrum access, modulation, scheduling, and primary user protection. The design objective is to optimize CR video quality while protecting primary users from harmful collisions. Unlike prior work on wireless video [4]–[7], the challenge stems from the dynamic channel availability processes, tightly coupled design choices, and the need to predict future channel status under the presence of spectrum sensing errors for rate control, partition, modulation and scheduling of real-time video data.

We show that the formulated problem is a non-linear integer programming (NIP) problem, which usually has high complexity to solve. The need for real-time solutions excludes many optimization techniques, while making low-complexity and efficient algorithms highly appealing. To solve the optimized CR video multicast problem, we take a three-step approach. For each group of pictures (GOP), we first determine the optimal partition (and thus rates and modulation-coding (MC) schemes) of FGS video data. We present two computationally efficient algorithms for this purpose: (i) a *sequential fixing* algorithm [8] based on a linear relaxation of the NIP problem [9]; and (ii) a *greedy algorithm* exploiting the inherent priority structure of layered video and channel qualities, with proven complexity and *optimality gap*. Next, the computed solution is adjusted in each time slot based on more recent channel sensing results and feedback, using a *refined greedy algorithm* with a much lower complexity. Finally, during each time slot, we use a *tile scheduling algorithm* to assign video packets to available channels with opportunistic channel access.

We present simulation results to provide a comparison study with alternative schemes as well as demonstrating the impact of several key design parameters on the overall system performance. We observe excellent performance achieved by the proposed algorithms, with considerable improvement over an alternative equal allocation scheme. We also found that the opportunistic spectrum access approach makes the CR video performance highly robust to sensing errors [2], [10].

The rest of this paper is organized as follows. In Section II, we present the CR video multicast framework. The proposed algorithms are described in Section III. We present our simulation study in Section IV and discuss related work in Section V. Section VI concludes the paper.

978-1-4244-3513-5/09/$25.00 ©2009 IEEE

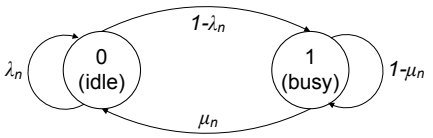

Fig. 1. The discrete-time Markov model for channel n, $n = 1, 2, \cdots, N$.

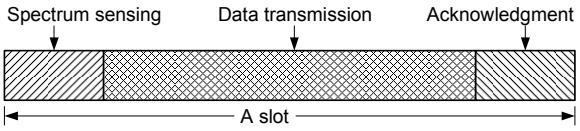

Fig. 2. The structure of a time slot.

II. The CR Video Multicast Framework

A. Network Model

We consider a spectrum band consisting of N channels, each evolving independently. For ease of explanation, we assume that the spectrum is continuous. A channel could be treated as a set of tones as in an OFDM system. An OFDM frame is transmitted through one antenna and received through one antenna, since the symbols are modulated to the channels by inverse fast Fourier transform (IFFT) (rather than using carriers with different frequencies). Therefore, there is no need to have more than one antennas at the receivers.

1) Primary Networks: We assume that the N channels are allocated to N primary networks. For ease of presentation, we assume the primary users access the channels following a synchronous slot structure. The occupancy of each of the N channels evolves following a discrete-time Markov process [2]. The network status in slot t is $\vec{S}(t) = [S_1(t), S_2(t), \cdots, S_N(t)]$, where $S_n(t)$ denotes the status of channel n with idle ($S_n(t) = 0$) and busy ($S_n(t) = 1$) states. Let λ_n and μ_n be the transition probability of remaining in state 0 and that from 1 to 0 for channel n, respectively. The channel availability model is illustrated in Fig. 1.

2) CR Network: We assume a CR network collocated with the N primary networks, within which a base station multicasts G real-time videos to G multicast groups, each having N_g users, $g = 1, 2, \cdots, G$. The base station seeks spectrum opportunities in the N channels. In each time slot, the base station chooses a set of channels $\mathcal{A}_1$ to sense and a set of channels $\mathcal{A}_2$ to access. The base station has $|\mathcal{A}_1|$ transceivers such that it can sense $|\mathcal{A}_1|$ channels simultaneously. A time slot and channel combination, termed a *tile*, is the minimum unit for resource allocation in our model.

As in [11], the basic protocol for the CR network is shown in Fig. 2. At the beginning of each time slot, the base station senses channels in $\mathcal{A}_1$ and then chooses available channels from $\mathcal{A}_2$ for opportunistic transmissions based on sensing results. After a successful transmission, the base station will receive an ACK from the user with the highest SNR in the target multicast group.[1]

As discussed, we adopt OFDM as multicast technology at the PHY layer. In OFDM, the spectrum is divided into narrowband channels and the signals are modulated on the channels in the frequency domain. Besides collisions with secondary users, another likely cause of interference to primary users is adjacent channel leakage as a result of secondary user transmissions in an adjacent channel. Adjacent channel leakage can usually be controlled through careful system design [13], [14]. By nullifying the symbols, such channels can also be easily vacated to avoid interference with primary users. Such adaptability makes OFDM highly suitable for CR networks [2], [13], [14]. We do not consider adjacent channel leakage in the rest of this paper.

[1] Although ACKs are not adopted in many multicast applications, it has been shown that the *feedback implosion* problem can be effectively solved using properly designed timers [12]. For example, each user in a group can start a timer reversely proportional to its channel SNR. The user who times out first will send an ACK, while all other users will cancel their scheduled ACKs when overhear the transmitted one (as in the IGMP protocol). The ACKs are also important in predicting future channel status (see Section II-B).

B. Spectrum Sensing

Although precise and timely channel state information is desirable for spectrum access and primary user protection, continuous full-spectrum sensing is both energy inefficient and hardware demanding. We assume $|\mathcal{A}_1|$ channels are sensed in each time slot, while sensing is carried out on every K channels [11]. The indices of the channels to be sensed are

$$n = (hK + q) \bmod N, \text{ for } h = 0, 1, \cdots, |\mathcal{A}_1| - 1. \quad (1)$$

The base station senses the channels in an increasing order with $q = (t \bmod K)$, where t is the discrete time index. It can be seen that the sensing process is periodic with period K. At the beginning of each slot, the base station chooses $|\mathcal{A}_1|$ channels to sense following (1). It then predicts the status of each channel based on the sensing results and channel history.

During the sensing process, two kind of detection errors may occur. With a *false alarm*, a spectrum opportunity will be wasted, while a *miss detection* may lead to collision with primary users. Let ϵ_n and δ_n denote the probabilities of false alarm and miss detection on channel n, respectively. The spectrum sensing performance can be represented by the Receiver Operation Characteristic (ROC) curve, which gives $1 - \delta_n$ as a function of ϵ_n [2], [10].

Let $\vec{R}(t) = [R_0(t), R_1(t), \cdots, R_{N-1}(t)]$ denote the sensing results in the current time slot. As shown in Fig. 3, the sensing error probabilities for a sensed channel n are

$$\begin{cases} P(R_n(t) = 1 | S_n(t) = 0) = \epsilon_n \\ P(R_n(t) = 0 | S_n(t) = 1) = \delta_n, \end{cases} \quad (2)$$

Let $\vec{a}(t) = [a_1(t), \cdots, a_{N-1}(t)]$ be the *belief vector*, where each element $a_n(t)$ is the estimate that channel n is available in time slot t, a conditional probability based on past sensing and transmission results on channel n. We derive the expressions of $a_n(t)$ for the following three cases, while considering both types of sensing errors. First, if channel n is not sensed in time slot t, $a_n(t)$ will be estimated from channel state $a_n(t-1)$. For example, if there is a data transmission on channel n and the base station receives an ACK, $a_n(t-1)$ is 1. If the receiver

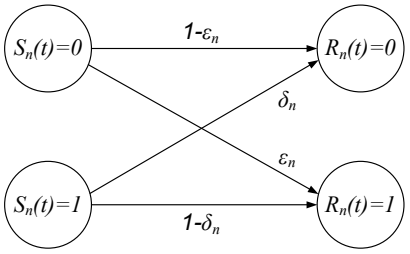

Fig. 3. Illustration of false alarm and miss detection sensing errors.

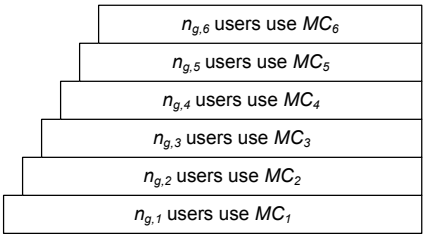

Fig. 4. Classification of group g users based on channel condition.

does not acknowledge the transmission due to collision with a primary user, $a_n(t-1)$ is 0. We have

$$\begin{aligned} a_n(t) &= P(S_n(t)=0|\theta_n) \\ &= \lambda_n a_n(t-1) + \mu_n[1-a_n(t-1)], \end{aligned} \quad (3)$$

where θ_n represents the history of channel n status. We can recursively expand (3) to write $a_n(t)$ as a function of $a_n(t-\tau)$ if there was neither transmission nor sensing on channel n in the past $\tau-1$ time slots.

If channel n is sensed in time slot t, its availability in t is conditioned on both the channel history and the sensing result. When sensing result is 0, we have

$$\begin{aligned} a_n(t) &= P(S_n(t)=0|R_n(t)=0,\theta_n) \\ &= \frac{P(S_n(t)=0,R_n(t)=0|\theta_n)}{\sum_{X_n\in\{0,1\}} P(S_n(t)=X_n,R_n(t)=0|\theta_n)} \\ &= \frac{P(R_n(t)=0|S_n(t)=0,\theta_n)P(S_n(t)=0|\theta_n)}{\sum_{X_n} P(R_n(t)=0|S_n(t)=X_n,\theta_n)P(S_n(t)=X_n|\theta_n)} \\ &= \frac{P(R_n(t)=0|S_n(t)=0)P(S_n(t)=0|\theta_n)}{\sum_{X_n} P(R_n(t)=0|S_n(t)=X_n)P(S_n(t)=X_n|\theta_n)} \\ &= \frac{\pi_n(t)(1-\epsilon_n)}{\pi_n(t)(1-\epsilon_n)+[1-\pi_n(t)]\delta_n}, \end{aligned} \quad (4)$$

where $\pi_n(t) = \lambda_n a_n(t-1) + \mu_i[1-a_n(t-1)]$. The fourth step in (4) is due to the memoryless property of the channel process. Similarly, we can derive the expression for $a_n(t)$ when sensing result is 1 as

$$\begin{aligned} a_n(t) &= P(S_n(t)=0|R_n(t)=1,\theta_n) \\ &= \frac{\pi_n(t)\epsilon_n}{\pi_n(t)\epsilon_n+[1-\pi_n(t)](1-\delta_n)}. \end{aligned} \quad (5)$$

C. Opportunistic Spectrum Access

Based on spectrum sensing results, the base station determines which channels to access for transmission of video data. We take an opportunistic spectrum access approach, aiming to exploit unused spectrum while bounding the interference to primary users. Let $\gamma_n \in (0,1)$ be the maximum allowed collision probability with primary users on channel n (a prescribed constraint), and $p_n^{tr}(t)$ the transmission probability on channel n for the base station in time slot t. The probability of collision caused by the base station should be kept below γ_n, i.e.,

$$p_n^{tr}(t)\,[1-a_n(t)] \le \gamma_n. \quad (6)$$

In addition to primary user protection, another important objective is to exploit unused spectrum as much as possible. Therefore, we have

$$p_n^{tr}(t) = \min\left\{1, \gamma_n/[1-a_n(t)]\right\}. \quad (7)$$

If $p_n^{tr}(t) = 1$, channel n will be accessed deterministically. If $p_n^{tr}(t) = \gamma_n/[1-a_n(t)] < 1$, channel n will be accessed opportunistically with probability $p_n^{tr}(t)$.

D. Modulation-Coding Schemes

At the PHY layer, we consider various modulation and channel coding schemes. Without loss of generality, we assume three choices of modulation, namely, QPSK, 16-QAM and 64-QAM, combined with three choices of forward error correction schemes, with rates 1/2, 2/3, and 3/4, respectively. We consider six combinations (i.e., Modulation-Coding(MC) schemes) in this paper: QPSK 1/2, QPSK 3/4, 16-QAM 1/2, 16-QAM 3/4, 64-QAM 2/3, and 64-QAM 3/4. Under the same channel condition, different MC schemes will give different data rates and symbol error rates. When a user's channel is good, it should adopt a high-rate modulation (e.g., 64-QAM) for a higher target symbol error rate. Conversely, it should adopt a low-rate modulation (e.g., QPSK) when the channel condition is poor. Let $\{MC_m\}_{m=1,\cdots,M}$ be the list of MC schemes.

We assume that each CR user measures its own channel and feedbacks such measurements to the base station. At the beginning of a time slot, the base station is able to collect the number $n_{g,m}$ of users in each multicast group g who can successfully decode MC_m signals for $m = 1, 2, \cdots, M$, as illustrated in Fig. 4.

E. Video Performance Measure

We use the quality of reconstructed video (i.e., peak-signal-noise-ratio (PSNR) in dB) as performance measure. To address the heterogeneous channel conditions among multicast users, each video g is encoded into one base layer with rate R_g^b and one enhancement layer with rate R_g^e. The total rate for video g is $R_g = R_g^b + R_g^e$. We adopt the FGS coding technique, such that the enhancement layer could be truncated at any bit location while *all* the remaining bits still being useful at the decoder [3]. The approximate "quality-rate" model used in [7] is adopted in our formulation:

$$Q_g(R_g) = Q_g^b + \beta_g(R_g - R_g^b) = Q_g^b + \beta_g R_g^e, \quad (8)$$

where Q_g is the PSNR of video g, Q_g^b the PSNR of the base layer, and β_g a constant depending on the specific video sequence and codec.

Since the base layer carries the most important data, the most reliable MC scheme $MC_{b(g)}$ should be used, where $b(g) = \max_i\{i : n_{g,i} = N_g\}$, for all g. We assume that the base layer is always transmitted using MC_1. If a user's channel is so poor that it cannot decode the MC_1 signal, we consider it disconnected from the CR network.

We further divide the enhancement layer into M sub-layers, where sub-layer m has rate $R_{g,m}^e$ and uses MC_m. For the same video frame, various MC schemes will produce symbols occupying different number of tiles. Assuming that MC_m can carry $b_{g,m}$ bits of video g in one tile, we denote the number of tiles for sub-layer m of video g as $l_{g,m} \geq 0$. We have $R_g^e = \sum_{m=1}^M R_{g,m}^e = \sum_{m=1}^M b_{g,m} l_{g,m}$.

F. Proportional Fair Allocation

We now consider the fairness criterion for deciding the rates of different users. For data communications, *proportional fairness* is a widely adopted measure, which can be achieved by maximizing the sum of logarithms of user rates (i.e., utilities) [15]. Since we consider video quality in this paper, we define the utility for user i in group g as $U_{g,i} = \log\left(Q_g^b + \beta_g R_{g,i}^e\right)$,

The total utility for group g is $U_g = \sum_{i=1}^{N_g} U_{g,i}$. Intuitively, a lower layer should use a lower (i.e., more reliable) MC scheme. This is because if a lower layer is lost, a higher layer cannot be used at the decoder even if it is correctly received. Considering the user classifications based on their MC schemes, we can rewrite U_g as follows [6]:

$$U_g = \sum_{k=1}^{M} (n_{g,k} - n_{g,k+1}) \log\left(Q_g^b + \beta_g \sum_{m=1}^{k} R_{g,m}^e\right), \quad (9)$$

where $n_{g,M+1} = 0$. The utility function of the entire CR video multicast system is $U = \sum_{g=1}^G U_g$.

III. Optimized Video Multicast in CR Networks

A. Outline of the Proposed Approach

As discussed, the CR video multicast problem is highly challenging since many design choices are tightly coupled. First, as users see different channels, such heterogeneity should be accommodated so that a user can receive a video quality commensurate to its channel quality. Second, we need to determine the video rates before transmission, which, however, depend on the channel evolution in the future and choice of MC schemes. Third, when the rates are known, we need to schedule the packets to the channels, with a trade-off between primary user protection and spectrum utilization. Finally, all the optimization decisions should be made in real-time. Low-complexity, but efficient algorithms are needed, while theoretical bounds on the optimality gap would be highly appealing.

To address heterogeneous user channels, we adopt FGS to produce a base layer with rate R_g^b and an enhancement layer with rate $\bar{R}_g^e$. Without loss of generality, we assume R_g^b is prescribed for an acceptable video quality, while $\bar{R}_g^e$ is set to a large value that is allowed by the codec. During transmission, we determine the *effective rate* for each enhancement layer $R_g^e \leq \bar{R}_g^e$ that depends on channel availabilities, sensing, and choices of MC schemes. We also need to find the optimal partition of the enhancement layer, such that each sub-layer uses a different MC scheme.

We determine the optimal partition of enhancement layers for each GOP, the choices of MC schemes, and the number of tiles for each sub-layer using a two-step procedure. First, we solve the optimal partition problem for every GOP based on an estimated (i.e., average) number of available tiles T_e in the next GOP window with complexity $O(MGT_e)$. Second, we dynamically adjust the tile allocations in each time slot according to more recent channel information. During each time slot, video packets are scheduled to the available channels such that the overall system utility is maximized. The adjustment algorithm has complexity $O(MGK)$, where $K \ll T_e$, and the scheduling algorithm has complexity $O(N \log N)$. They are thus suitable for execution in each time slot.

In real-time video, overdue packets generally do not contribute to improving the received quality. We assume that the data from a GOP should be be delivered in the next GOP window consisting of T_{GOP} time slots. Since the base layer is essential for decoding a video, we assume that the base layers of all the videos are coded using MC_1. For the M sub-layers of the enhancement layer, a more important sub-layer will be coded using a more reliable (i.e., lower) MC scheme. At the beginning of each GOP window, all the base layers are transmitted taking the available tiles. *Retransmissions* will be scheduled if no ACK received for a base layer packet. After the base layers are transmitted, we allocate the remaining available tiles in the GOP window for the enhancement layer. The same rule applies, such that a higher sub-layer will be transmitted if and only if all the lower sub-layers are acknowledged. This is due to the decoding dependencies of layered video.

In each time slot t, the base station opportunistically access every channel n with probability $p_n^{tr}(t)$ given in (7). Specifically, for each channel n, the base station generates a random number $x_n(t) \in [0, 1)$. If $x_n(t) \leq p_n^{tr}(t)$, the most important packet among those not ACKed in the previous GOP will be transmitted on channel n. If an ACK is received for this packet at the end of time slot t, this packet is successfully received by at least one of the users and will be removed from the transmission buffer. Otherwise, there is a collision with primary user and this packet will remain in the transmission buffer and will be retransmitted.

In the following, we describe in detail the algorithms for tile allocation (or, optimal partition of the enhancement layers) and for scheduling the video packets in each time slot.

B. Enhancement Layer Partition and Tile Allocation

As a first step, we need to determine the effective rate for each enhancement layer $R_g^e \leq \bar{R}_g^e$ that depends on network conditions. We also need to determine the optimal partition of

each enhancement layer. Clearly, the solutions will be highly dependent on the channel availability processes.

Recall that the base layers are transmitted using MC_1 first in each GOP window. The remaining available tiles can then be allocated to the enhancement layers. We assume that the number of tiles used for enhancement layers in a GOP window, T_e, is known at the beginning of the GOP window. For example, we can estimate T_e by computing the total average "off" intervals of all the N channels based on the channel model (see Fig. 1), decreased by the number of tiles used for the base layers. We then split the enhancement layer of each video g into M sub-layers, each occupying $l_{g,m}$ tiles when coded with MC_m, $m = 1, 2, \cdots, M$.

We formulate an optimization problem, OPT-Part, as follows. OPT-Part is solved at the beginning of each GOP window to determine the optimal partition of the enhancement layer.

$$\text{maximize: } U(\vec{l}) = \sum_{g=1}^{G}\sum_{k=1}^{M}(n_{g,k} - n_{g,k+1}) \times \log\left[Q_g^b + \beta_g \sum_{m=1}^{k} b_{g,m} l_{g,m}\right] \quad (10)$$

$$\text{subject to: } \sum_{g=1}^{G}\sum_{m=1}^{M} l_{g,m} \le T_e \quad (11)$$

$$\sum_{m=1}^{M} b_{g,m} l_{g,m} \le \bar{R}_g^e,\ g \in [1, \cdots, M] \quad (12)$$

$$l_{g,m} \ge 0,\ m \in [1, \cdots, M], g \in [1, \cdots, G]. \quad (13)$$

The objective is to maximize the overall system utility by choosing optimal values for the $l_{g,m}$'s. We can derive the effective video rates as $R_g^e = \sum_{m=1}^{M} b_{g,m} l_{g,m}$. The formulated problem is a non-linear integer programming (NIP) problem, which we conjecture is NP-hard. However we leave the hardness proof for future work. In the following, we present two algorithms for computing near-optimal solutions to problem OPT-Part: (i) a sequential fixing (SF) algorithm based on a linear relaxation, and (ii) a greedy algorithm with proven optimality gap. Both algorithms have polynomial complexity.

1) A Sequential Fixing Algorithm : We now present an SF algorithm for near-optimal solutions. In this algorithm, the original NIP is first linearized to obtain a linear programming (LP) relaxation. Then we iteratively solve the LP, while fixing one integer variable in every iteration [8].

We use the Reformulation-Linearization Technique (RLT) to obtain the LP relaxation [9]. RLT is a technique that can be used to produce LP relaxations for an underlying nonlinear, nonconvex polynomial programming problem. This relaxation will provide a tight upper bound for a maximization problem. Specifically, We linearize the logarithm function in (10) over some suitable, tightly-bounded interval using a polyhedral outer approximation comprised of a convex envelope in concert with several tangential supports. We further relax the integer constraints, i.e., allowing the $l_{g,m}$'s to take fractional values. Then we obtain an upper-bounding LP relaxation that can be solved in polynomial time. We refer interested readers to [9] for a detailed description of the technique.

TABLE I

THE SEQUENTIAL FIXING (SF) ALGORITHM

1:	Use RLT to linearize the original problem.
2:	Solved the LP relaxation
3:	Suppose $l_{\hat{g},\hat{m}}$ is the integer variable with the minimum $\lceil l_{\hat{g},\hat{m}} \rceil - l_{\hat{g},\hat{m}}$ or $l_{\hat{g},\hat{m}} - \lfloor l_{\hat{g},\hat{m}} \rfloor$ value among all $l_{g,m}$ variables that remain to be fixed, round it up or down to the nearest integer.
4:	If all $l_{g,m}$'s are fixed, got to Step 6.
5:	Otherwise, reformulate and solve a new relaxed LP with the newly fixed $l_{g,m}$ variables, and go to Step 3.
6:	Formulate and solve an LP based on all fixed $l_{g,m}$ variables.

TABLE II

THE GREEDY ALGORITHM (GRD1)

1:	Initialize $l_{g,m} = 0$ for all g and m
2:	Initialize $A = \{1, 2, \cdots, G\}$
3:	While $\left(\sum_{g=1}^{G}\sum_{m=1}^{M} l_{g,m} \le T_e$ and A is not empty$\right)$
4:	Find $l_{\hat{g},\hat{m}}$ which can be increased by one: $\vec{e}_{\hat{g},\hat{m}} = \arg\max_{g\in A, m\in[1,\cdots,M]} \left\{ \frac{U(\vec{l}+\vec{e}_{g,m}) - U(\vec{l})}{b_{g,m} + R/T_e} \right\}$
5:	$\vec{l} = \vec{l} + \vec{e}_{\hat{g},\hat{m}}$
6:	If $\left(\sum_m b_{\hat{g},m} l_{\hat{g},m} > \bar{R}_g^e\right)$
7:	$\vec{l} = \vec{l} - \vec{e}_{\hat{g},\hat{m}}$
8:	Delete $\hat{g}$ from A
9:	End if
10:	End while

Once the LP relaxation is obtained, we solve it iteratively. During each iteration, we find the $l_{\hat{g},\hat{m}}$ which has the minimum value for $(\lceil l_{\hat{g},\hat{m}} \rceil - l_{\hat{g},\hat{m}})$ or $(l_{\hat{g},\hat{m}} - \lfloor l_{\hat{g},\hat{m}} \rfloor)$ among all fractional $l_{g,m}$'s, and round it up or down to the nearest integer. We next reformulate and solve a new LP with $l_{\hat{g},\hat{m}}$ fixed. This procedure repeats until all the $l_{g,m}$'s are fixed. The complete SF algorithm is given in Table I.

2) A Greedy Algorithm GRD1 : Although the SF algorithm can compute a near-optimal solution in polynomial time, it does not provide any guarantee on the optimality gap. In the following, we describe a greedy algorithm, termed GRD1, which exploits the inherent priority structure of layered video and MC schemes. The complete greedy algorithm is given in Table II, where $R = \sum_{g=1}^{G} R_g^e$ is the total rate of all the enhancement layers and $\vec{e}_i$ is a *unit vector* with "1" at the i-th location and "0" at all other locations.

In GRD1, all the $l_{g,m}$'s are initially set to 0. During each iteration, one tile is allocated to the $\hat{m}$-th sub-layer of video $\hat{g}$. In Step 4, $l_{\hat{m},\hat{g}}$ is chosen to be the one that achieves the largest increase in terms of the "normalized" utility (i.e., $[U(\vec{l}+\vec{e}_{g,m}) - U(\vec{l})]/[b_{g,m} + R/T_e]$) if it is assigned with an additional tile. Lines 6, 7, and 8 check if the assigned rate exceeds the rate of the coded enhancement layer.

Theorem 1: The greedy algorithm GRD1 shown in Table II has a complexity $O(MGT_e)$. It guarantees a solution that is within a factor of $(1 - e^{-1/2})$ of the global optimal solution.

Proof: (i) *Complexity*: In Step 4 in Table II, it takes $O(MG)$ to solve for $\vec{e}_{\hat{g},\hat{m}}$. Since each iteration assigns one tile to sub-layer $\hat{m}$ of group $\hat{g}$, it takes T_e iterations to allocate all the available tiles in a GOP window. Therefore, the overall

complexity of GRD1 is $O(MGT_e)$.

(ii) *Optimality Bound*: This proof is extended from a result first shown in [6]. We first show an interesting property of group utility $U_g(\vec{l})$, which will be used in the proof of optimality gap. For two vectors $\vec{l}_g^1$ and $\vec{l}_g^2$, we have

$$\begin{aligned}
&U_g(\vec{l}_g^1) - U_g(\vec{l}_g^2)\\
=&\sum_{k=1}^{M}(n_{g,k}-n_{g,k+1})\log\left(1+\frac{\sum_{m=1}^{k} b_{g,m}(l_{g,m}^1-l_{g,m}^2)}{Q_g^b+\sum_{m=1}^{k} b_{g,m}l_{g,m}^2}\right)\\
\le&\sum_{k=1}^{M}\sum_{m=1}^{k}(l_{g,m}^1-l_{g,m}^2)^+(n_{g,k}-n_{g,k+1})\times\\
&\qquad\log\left(1+\frac{b_{g,m}}{Q_g^b+\sum_{m=1}^{k} b_{g,m}l_{g,m}^2}\right)\\
\le&\sum_{k=1}^{M}\sum_{m=1}^{M}(l_{g,m}^1-l_{g,m}^2)^+(n_{g,k}-n_{g,k+1})\times\\
&\qquad\log\left(1+\frac{b_{g,m}}{Q_g^b+\sum_{m=1}^{k} b_{g,m}l_{g,m}^2}\right)\\
=&\sum_{m=1}^{M}(l_{g,m}^1-l_{g,m}^2)^+\left[U_g(\vec{l}_g^2+b_{g,m})-U(\vec{l}_g^2)\right], \qquad (14)
\end{aligned}$$

where $y^+ = \max\{0, y\}$. The first inequality is due to the concavity of logarithm functions.

Next we prove the optimality bound. Let $\vec{l}_t$ be the output of GRD1 after t iterations. The utility gap between the optimal solution and the GRD1 solution is denoted as $F_t = U(\vec{l}^*) - U(\vec{l}_t)$. Let $\vec{e}_{\hat{g},\hat{m}}(t)$ be the argument found in Step 4 of GRD1 after t iterations. We have $\vec{l}_t = \vec{l}_{t-1} + \vec{e}_{\hat{g},\hat{m}}(t)$ and

$$\begin{aligned}
&F_{t-1} = U(\vec{l}^*) - U(\vec{l}_{t-1})\\
\le&\sum_g\sum_m(l_{g,m}^*-l_{g,m})^+[U(\vec{l}_{t-1}+\vec{e}_{g,m}(t))-U(\vec{l}_{t-1})]\\
\le&\sum_g\sum_m(l_{g,m}^*-l_{g,m})^+[U(\vec{l}_{t-1}+\vec{e}_{\hat{g},\hat{m}}(t))-\\
&\qquad U(\vec{l}_{t-1})]\frac{b_{g,m}+R/T_e}{b_{\hat{g},\hat{m}}(t)+R/T_e}\\
\le&\frac{U(\vec{l}_t)-U(\vec{l}_{t-1})}{b_{\hat{g},\hat{m}}(t)+R/T_e}\sum_g\sum_m[l_{g,m}^*(b_{g,m}+R/T_e)].
\end{aligned}$$

The first inequality is due to (14) and the second inequality follows Step 4 of GRD1. It follows (11) and (12) that $\sum_g\sum_m l_{g,m}^* \le T_e$ and $\sum_g\sum_m b_{g,m}l_{g,m}^* \le R$. We have

$$F_{t-1} \le (F_{t-1}-F_t)\frac{2R}{b_{\hat{g},\hat{m}}(t)+R/T_e}.$$

Solving for F_t, we have

$$F_t \le F_{t-1}\left[1-\frac{b_{\hat{g},\hat{m}}(t)+R/T_e}{2R}\right].$$

Suppose the *while* loop in Table II has been executed k times when the solution is obtained. It follows that

$$\begin{aligned}
F_k &\le F_{k-1}\left[1-\frac{b_{\hat{g},\hat{m}}(k)+R/T_e}{2R}\right]\\
&\le F_0\prod_{t=1}^{k}\left[1-\frac{b_{\hat{g},\hat{m}}(t)+R/T_e}{2R}\right]\\
&\le F_0\left\{1-\frac{1}{2kR}\sum_{t=1}^{k}[b_{\hat{g},\hat{m}}(t)+R/T_e]\right\}^k.
\end{aligned}$$

The *while* loop exits when one or both of two constraints are violated. If $\sum_g\sum_m l_{g,m} \le T_e$ is violated, there is no tile that can be used. Therefore $k \ge T_e$ and $\sum_{t=1}^{k} R/T_e \ge R$. If constraint "$A$ is not empty" is violated, all the videos have been allocated sufficient number of tiles and will be transmitted at full rate. We have $\sum_{t=1}^{k} b_{\hat{g},\hat{m}}(t) \le R$ in this case. It follows the above reasoning that

$$\begin{aligned}
F_k &\le F_0\left\{1-\frac{1}{2kR}\sum_{t=1}^{k}[b_{\hat{g},\hat{m}}(t)+R/T_e]\right\}^k\\
&\le F_0\left[1-1/(2k)\right]^k \le F_0e^{-1/2}.
\end{aligned}$$

Since $F_0 = U(\vec{l}^*)$, we have

$$U(\vec{l}_k) \ge (1-e^{-1/2})U(\vec{l}^*).$$

Therefore, we conclude that the GRD1 solution is bounded by $(1-e^{-1/2})U(\vec{l}^*)$ and $U(\vec{l}^*)$. ■

3) A Refined Greedy Algorithm GRD2 : GRD1 computes $l_{g,m}$'s based on an estimate of network status $\vec{S}(t)$ in the next T_{GOP} time slots. Depending on channel parameters, the computed $l_{g,m}$'s may not be accurate, especially when T_{GOP} is large. In the following, we present a refined greedy algorithm, termed GRD2, which adjusts the $l_{g,m}$'s based on more accurate estimation of the channel status.

GRD2 is executed at the beginning of every time slot within a GOP window. It estimates the number of available tiles $T_e(t)$ in the next T_{est} successive time slots, where $1 \le T_{est} \le T_{GOP}$ is a design parameter depending on the coherence time of the channels. Such estimates are more accurate than that in GRD1 since they are based on recently received ACKs and more recent sensing results.

Specifically, we estimate $T_e(t)$ using the belief vector $\vec{a}(t)$ in time slot t. Recall that $a_n(t)$'s are computed based on the channel model, feedback, sensing results, and sensing errors, as given in (3), (4), and (5). For the next time slot, $a_n(t+1)$ can be estimated as $\hat{a}_n(t+1) = \lambda_n a_n(t) + \mu_n[1-a_n(t)] = (\lambda_n-\mu_n)a_n(t)+\mu_n$. Recursively, we can derive $\hat{a}_n(t+\tau)$ for the next τ time slots.

$$\hat{a}_n(t+\tau) = (\lambda_n-\mu_n)^\tau a_n(t) + \mu_n\frac{1-(\lambda_n-\mu_n)^\tau}{1-(\lambda_n-\mu_n)}. \qquad (15)$$

Recall that at the beginning part of a GOP window, all the base layers will be first transmitted. We start the estimation after all the base layers have been successfully received (possibly with retransmissions). The estimated number of available

TABLE III

THE REFINED GREEDY ALGORITHM (GRD2) FOR EACH TIME SLOT

Initialize $l_{g,m} = 0$ for all g and m
Initialize $A = \{1, 2, \cdots, G\}$
Initialize $N_{ack}(0) = 0$
Estimate $T_e(1)$ based on the Markov Chain channel model
Use GRD1 to find all $l_{g,m}$'s based on $T_e(1)$
While $t = 2$ to T_{GOP}
  Estimate $T_e(t)$
  If $(T_e(t) + N_{ack}(t-1) < T_e(t-1) + N_{ack}(t-2))$
    While $(\sum_{g=1}^{G}\sum_{m=1}^{M} l_{g,m} > T_e(t) + N_{ack}(t-2))$
      Find $l_{\hat{g},\hat{m}}$ which can be reduced by 1:
        $\vec{e}_{\hat{g},\hat{m}} = \arg\min_{\forall g,m\in\{m',\cdots,M\}} \left\{ \frac{U(\vec{l}) - U(\vec{l} - \vec{e}_{g,m})}{b_{g,m} + R/T_e} \right\}$
      $\vec{l} = \vec{l} - \vec{e}_{\hat{g},\hat{m}}$
      If $(\hat{g} \notin A)$
        Add $\hat{g}$ to A
      End if
    End while
  End if
  If $(T_e(t) + N_{ack}(t-1) > T_e(t-1) + N_{ack}(t-2))$
    While $(\sum_{g=1}^{G}\sum_{m=1}^{M} l_{g,m} \le T_e(t) + N_{ack}(t-1)$ and A is not empty)
      Find $l_{\hat{g},\hat{m}}$ which can be increased by 1
        $\vec{e}_{\hat{g},\hat{m}} = \arg\max_{g\in A, m\in\{m',\cdots,M\}} \left\{ \frac{U(\vec{l} + \vec{e}_{g,m}) - U(\vec{l})}{b_{g,m} + R/T_e} \right\}$
      $\vec{l} = \vec{l} + \vec{e}_{\hat{g},\hat{m}}$
      If $\sum_m b_{\hat{g},m} l_{\hat{g},m} > \bar{R}_g^e$
        $\vec{l} = \vec{l} - \vec{e}_{\hat{g},\hat{m}}$
        Delete $\hat{g}$ from A
      End if
    End while
  End if
  Update $N_{ack}(t-1)$
End while

tiles in the following T_{est} time slots can be computed as:

$$T_e(t) = \sum_{n=1}^{N}\sum_{\tau=0}^{t_{min}} \hat{a}_n(t+\tau). \tag{16}$$

where $\hat{a}_n(t+0) = a_n(t)$ and $t_{min} = \min\{T_{est} - 1, T_{GOP} - (t \bmod T_{GOP})\}$. Although $T_e(t)$ may not be an integer, this fact does not affect the outcome of GRD2.

We then adjust the $l_{g,m}$'s based on $T_e(t)$ and $N_{ack}(t)$, the number of ACKs received by the base station in time slot t. Specifically, if $T_e(t) + N_{ack}(t-1) > T_e(t-1) + N_{ack}(t-2)$, there are more tiles that can be allocated and we can increase some of the $l_{g,m}$'s. On the other hand, if $T_e(t) + N_{ack}(t-1) < T_e(t-1) + N_{ack}(t-2)$, we have to reduce some of the $l_{g,m}$'s. Due to layered videos, when we increase the number of allocated tiles, we only need to consider $l_{g,m}$ for $m = m', m'+1, \cdots, M$, where $MC_{m'}$ is the highest MC scheme used in the previous time slot. Similarly, when we reduce the number of allocated tiles, we only need to consider $l_{g,m}$ for $m = m', m'+1, \cdots, M$.

The refined greedy algorithm is given in Table III. For time slot t, the complexity of GRD2 is $O(MGK)$, where $K = |N_{ack}(t-1) - N_{ack}(t-2) + T_e(t) - T_e(t-1)|$. Since $K << T_e$, the complexity of GRD2 is much lower than GRD1, making it suitable to be executed in each time slot.

TABLE IV

ALGORITHM FOR TILE SCHEDULING IN A TIME SLOT

Initialize m_g to the lowest MC that has not been ACKed for all g
Initialize i_g to the first packet that has not been ACKed for all g
Sort $\{c_n(t)\}$ in decreasing order. Let the sorted channel list be indexed by j.
While ($j = 1$ to N)
  Find the group having the maximum increase in $U(g)$:
    $\hat{g} = \arg\max_{\forall g} \text{Inc}(g, m_g, i_g)$
  Allocate the tile on channel j to group $\hat{g}$
  Update $m_{\hat{g}}$ and $i_{\hat{g}}$
End while

C. Tile Scheduling in a Time Slot

In each time slot t, we need to schedule the remaining tiles for transmission on the N channels. We define $\text{Inc}(g, m, i)$ to be the increase in the group utility function $U(g)$ after the i-th tile in the sub-layer using MC_m is successfully decoded. It can be shown that

$$\text{Inc}(g,m,i) = \sum_{k=m}^{M} (n_{g,k} - n_{g,k+1}) \times \log\left[1 + \frac{\beta_g b_{g,m}}{Q_g^b + \beta_g \sum_{u=1}^{m-1} b_{g,u} l_{g,u} + (i-1)\beta_g b_{g,m}}\right].$$

$\text{Inc}(g, m, i)$ can be interpreted as the *reward* after the tile is successfully received. Letting $c_n(t)$ be the probability that the tile is successfully received, we have $c_n(t) = p_n^{tr}(t) a_n(t)$. Our objective of tile scheduling is to maximize the expected reward, i.e.,

$$\text{maximize:} \quad \text{E}[\text{Reward}(\vec{\xi})] = \sum_{n=1}^{N} c_n(t) \cdot \text{Inc}(\xi_n), \tag{17}$$

where ξ_n is the tile allocation for channel n, i.e., representing the three-tuple $\{g, m, i\}$. We present the tile scheduling (TS) algorithm in Table IV, which solves the above optimization problem. The complexity of TS is $O(N \log N)$. We have the following theorem for TS.

Theorem 2: E[Reward] is maximized if $\text{Inc}(\xi_i) > \text{Inc}(\xi_j)$ when $c_i(t) > c_j(t)$ for all i and j.

Proof: Suppose there exists a pair of i and j where $\text{Inc}(\xi_i) > \text{Inc}(\xi_j)$ and $c_i(t) < c_j(t)$. We can further increase E[Reward] by switching the tile assignment, i.e., assign channel i to ξ_j and channel j to ξ_i. With this new assignment, the net increase in E[Reward] is

$$\begin{aligned} & c_j(t)\text{Inc}(\xi_i) + c_i(t)\text{Inc}(\xi_j) - c_i(t)\text{Inc}(\xi_i) - c_j(t)\text{Inc}(\xi_j) \\ = \;& [c_j(t) - c_i(t)][\text{Inc}(\xi_i) - \text{Inc}(\xi_j)] > 0. \end{aligned}$$

Therefore E[Reward] is maximized when the $\{\text{Inc}(\xi_i)\}$ and $\{c_i(t)\}$ are in the same order. ■

The greedy TS algorithm in Table IV computes the optimal solution that maximizes the expected reward in (17).

IV. SIMULATION RESULTS

We evaluate the performance of the proposed CR video multicast framework using a customized simulator implemented with a combination of C and MATLAB. Specifically, the LPs

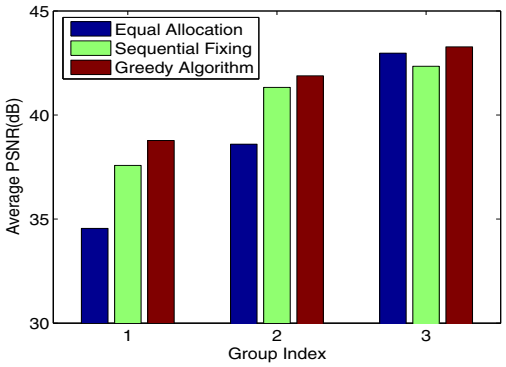

Fig. 5. Average PSNR of all multicast users.

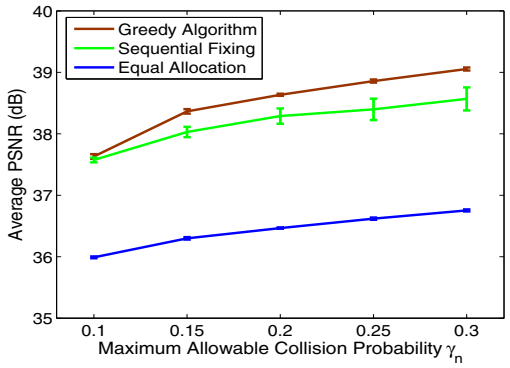

Fig. 6. Average PSNR of all users versus γ_n (with 95% confidence intervals).

are solved using the MATLAB Optimization Toolbox. For the results reported in this section, we have $N = 12$ channels (unless otherwise specified). The channel parameters λ_n and μ_n are set between $(0, 1)$. The maximum allowed collision probability γ_n is set to 0.2 for all n unless otherwise specified.

The CR base station multicasts three Common Intermediate Format (CIF, 352×288) video sequences to three multicast groups, i.e., *Bus* to group 1, *Foreman* to group 2, and *Mother & Daughter* to group 3. The $n_{1,m}$'s are [42, 40, 36, 30, 22, 12] (i.e., 42 users can decode MC_1 signal, 40 users can decode MC_2 signal, and so forth); the $n_{2,m}$'s are [51, 46, 40, 32, 23, 12] and the $n_{3,m}$'s are [49, 44, 40, 32, 24, 13].[2] The number of bits carried in one tile using the MC schemes are 1 kb/s, 1.5 kb/s, 2 kb/s, 3 kb/s, 5.3 kb/s, and 6 kb/s, respectively. We choose T_{GOP}=150 and $T_{est} = 10$, the sensing interval $K = 3$, the probability of false alarm $\epsilon_n = 0.3$ and the probability of miss detection $\delta_n = 0.25$ for all n, unless otherwise specified.

In every simulation, we compare three schemes: (i) a simple heuristic scheme that equally allocates tiles to each group (Equal Allocation); (ii) A scheme based on SF (Sequential Fixing), and (iii) a scheme based on the greedy algorithms (Greedy Algorithm). These schemes have increasing complexity in the order of Equal Allocation, Greedy Algorithm, and Sequential Fixing. They differ on how to solve Problem OPT-Part, while the same tile scheduling algorithm and opportunistic spectrum access scheme are used in all the schemes. Each point in the figures is the average of 10 simulation runs, with 95% confidence intervals plotted.

In Fig. 5 we plot the average PSNR among all users in each multicast group. for all the groups, Greedy Algorithm achieves the best performance, with up to 4.2 dB improvements over Equal Allocation and up to 0.6 dB improvements over Sequential Fixing.

In Fig. 6, we examine the impact of the maximum allowed collision probability γ_n on received video quality. We increase γ_n from 0.1 to 0.3, and plot the average PSNR values among all the users. Intuitively, a higher γ_n allows a higher transmission probability $p_n^{tr}(t)$ for the base station (see (7)), thus allowing the base station to grab more spectrum opportunities and improve the overall video quality. This is illustrated in the figure where all the three curves increase as γ_n gets larger. We also observe that the curves for Sequential Fixing and Equal Allocation are roughly parallel to each other, while the Greedy Algorithm curve has a steeper slope. This implies that Greedy Algorithm is more efficient in exploiting the additional available spectrum allowed by an increased γ_n.

[2]We also simulated the case of dynamic channels. Similar results are obtained but omitted due to lack of space.

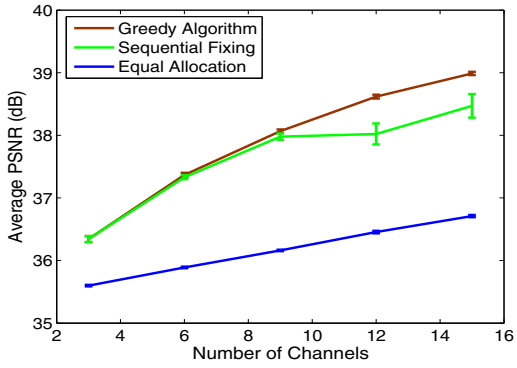

Fig. 7. Average PSNR of all users versus N (with 95% confidence intervals).

In Fig. 7, we examine the impact of number of channels N on the multicast video quality. We increase N from 3 to 15, in step size of 3, and plot the average PSNR values of all multicast users. As expected, the more channels, the more spectrum opportunities for the CR networks, and the better the video quality. Again, we observe that the Greedy Algorithm curve has the steepest slope, implying it is more efficient in exploiting the increased spectrum for video transmissions.

Finally, we examine the impact of sensing errors on the multicast video quality in Fig. 8. We test five sets of $\{\epsilon_n, \delta_n\}$ values as follows: [0.10, 0.38],[0.30, 0.25],[0.5, 0.17],[0.70, 0.10] and [0.9, 0.04] [10], and plot the average PSNR values of all users. It is quite interesting to see that the video quality is not very sensitive to sensing errors. Even as ϵ_n is increased 9 times from 10% to 90%, there is only 0.58 dB reduction (or a 1.5% normalized reduction) in average PSNR when Greedy Algorithm is used. The same can be observed for the other two curves. We conjecture that this is due to the opportunistic spectrum access approach adopted in all the three schemes.

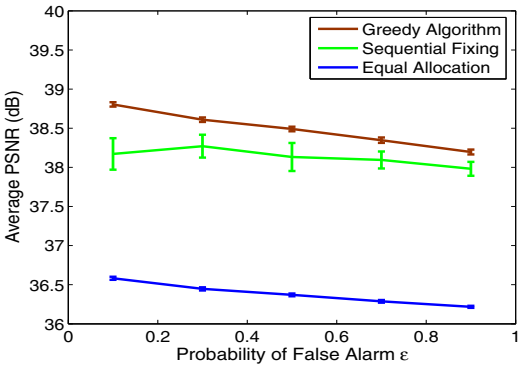

Fig. 8. Average PSNR of all users for various $\{\epsilon_n, \delta_n\}$ values (with 95% confidence intervals).

V. Related Work

As observed in [1], [2], the mainstream CR research has been focused on spectrum sensing and dynamic spectrum access (i.e., the MAC and PHY) issues. Fore example, the approach of iteratively sensing a selected subset of available channels has been adopted in the design of CR MAC protocols (e.g., see [11]). The important trade-off between the two types of sensing errors is addressed in depth in [10].

The equally important QoS issue has been considered only in a few papers [16]–[19], where the focus is on the so-called "network-centric" metrics such as maximum throughput. In [20], a game-theoretic framework is described for resource allocation for multimedia transmissions in spectrum agile wireless networks. In this interesting work, each wireless station plays a resource management game, which is coordinated by a network moderator. A mechanism-based resource management scheme determines the amount of transmission time to be allocated to various users on different frequency bands such that certain global system metrics are optimized.

Video multicast, as one of the most important multimedia services, has attracted considerable efforts from the research community. Layered video multicast has been studied in the context of mobile ad hoc networks (e.g., see [4], [5]) and infrastructure-based wireless networks (e.g., see [6], [7]). The main difference between this and the prior studies is that unlike in the prior work where the spectrum is exclusively used by the video sessions, we have to consider the presence and protection of primary users, which makes the problem more interesting and challenging.

VI. Conclusion

In this paper, we addressed the problem of video multicast in CR networks. The problem formulation took video quality and proportional fairness as objectives, while considering important cross-layer design factors such as considering scalable video coding, video rate control, spectrum sensing, opportunistic spectrum access, primary user protection, scheduling, retransmission and modulation. We proposed efficient optimization and scheduling algorithms for highly competitive solutions, and proved the complexity and optimality bound of the proposed greedy algorithm. Our simulation results demonstrate the efficacy of the proposed approach.

Acknowledgments

Shiwen Mao's research is supported in part by the US National Science Foundation under Grant ECCS-0802113 and through the Wireless Internet Center for Advanced Technology (WICAT) at Auburn University.

References

[1] I. Akyildiz, W. Lee, M. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Netw. J.*, vol. 50, no. 9, pp. 2127–2159, Sept. 2006.

[2] Q. Zhao and B. Sadler, "A survey of dynamic spectrum access," *IEEE Signal Process. Mag.*, vol. 24, no. 3, pp. 79–89, May 2007.

[3] H. Radha, M. van der Schaar, and Y. Chen, "The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP," *IEEE Trans. Multimedia*, vol. 3, no. 1, pp. 53–68, Mar. 2001.

[4] S. Mao, X. Cheng, Y. Hou, and H. Sherali, "Multiple description video multicast in wireless ad hoc networks," *ACM/Kluwer Mobile Netw. Appl. J.*, vol. 11, no. 1, pp. 63–73, January 2006.

[5] W. Wei and A. Zakhor, "Multiple tree video multicast over wireless ad hoc networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 1, pp. 2–15, Jan. 2007.

[6] S. Deb, S. Jaiswal, and K. Nagaraj, "Real-time video multicast in WiMAX networks," in *Proc. IEEE INFOCOM'08*, Phoenix, AZ, Apr. 2008, pp. 1579–1587.

[7] M. van der Schaar, S. Krishnamachari, S. Choi, and X. Xu, "Adaptive cross-layer protection strategies for robust scalable video transmission over 802.11 WLANs," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 10, pp. 1752–1763, Dec. 2003.

[8] Y. Hou, Y. Shi, and H. Sherali, "Optimal spectrum sharing for multi-hop software defined radio networks," in *Proc. IEEE INFOCOM'07*, Anchorage, AK, May 2007, pp. 1–9.

[9] S. Kompella, S. Mao, Y. Hou, and H. Sherali, "On path selection and rate allocation for video in wireless mesh networks," *IEEE/ACM Trans. Netw.*, in press.

[10] Y. Chen, Q. Zhao, and A. Swami, "Joint design and separation principle for opportunistic spectrum access in the presence of sensing errors," *IEEE Trans. Inf. Theory*, vol. 54, no. 5, pp. 2053–2071, May 2008.

[11] Q. Zhao, S. Geirhofer, L. Tong, and B. Sadler, "Opportunistic spectrum access via periodic channel sensing," *IEEE Trans. Signal Process.*, vol. 36, no. 2, pp. 785–796, Feb. 2008.

[12] J. Nonnenmacher and E. Biersack, "Optimal multicast feedback," in *Proc. IEEE INFOCOM'98*, San Francisco, CA, Mar./Apr. 1998, pp. 964–971.

[13] U. Berthold and F. Jondral, "Guidelines for designing OFDM overlay systems," in *Proc. IEEE DySPAN'05*, Baltimore, MD, Nov. 2005, pp. 626–629.

[14] H. Tang, "Some physical layer issues of wide-band cognitive radio systems," in *Proc. IEEE DySPAN'05*, Baltimore, MD, Nov. 2005, pp. 151–159.

[15] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *J. Operational Research Society*, vol. 49, no. 3, pp. 237–252, Mar. 1998.

[16] T. Weingart, D. Sicker, and D. Grunwald, "A statistical method for reconfiguration of cognitive radios," *IEEE Wireless Commun. Mag.*, vol. 14, no. 4, pp. 34–40, Aug. 2007.

[17] H. Su and X. Zhang, "Cross-layer based opportunistic MAC protocols for QoS provisioning over cognitive radio wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 118–129, Jan. 2008.

[18] S. Jafar and S. Srinivasa, "Capacity limits of cognitive radio with distributed and dynamic spectral activity," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 3, pp. 529–537, Apr. 2007.

[19] C.-T. Chou, S. S. N., H. Kim, and K. Shin, "What and how much to gain by spectral agility," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 3, pp. 576–588, Apr. 2007.

[20] A. Fattahi, F. Fu, M. van der Schaar, and F. Paganni, "Mechanism-based resource allocation for multimedia transmission over spectrum agile wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 3, pp. 601–612, Apr. 2007.