Cognitive Radio Networks for Delay-Sensitive Applications: Games and Learning

Yuanzhang Xiao and Mihaela van der Schaar

Abstract We have witnessed an explosion in wireless video traffic in recent years. Video applications are bandwidth-intensive and delay-sensitive, and hence require efficient utilization of spectrum resources. Born to utilize wireless spectrum more efficiently, cognitive radio networks are promising candidates for deployment of wireless video applications. In this chapter, we introduce our recent advances in *foresighted resource allocation mechanisms* that enable multi-user wireless video applications over cognitive radio networks. The introduced resource allocation mechanisms are foresighted, in the sense that they optimize the *long-term* video quality of the wireless users. Due to the temporal coupling of delay-sensitive video applications, such foresighted mechanisms outperform mechanisms that maximize the short-term video quality. Moreover, the introduced resource allocation mechanisms allow wireless users to optimize while learning the unknown dynamics in the environment (e.g., incoming traffic, primary user activities). Finally, we introduce variations of the mechanisms that are suitable for networks with self-interested users. These mechanisms ensure efficient video resource allocation even when the users are self-interested and aim to maximize their individual video quality. The foresighted resource allocation mechanisms introduced in this chapter are built upon our theoretical advances in multi-user Markov decision processes, reinforcement learning, and dynamic mechanism design.

Yuanzhang Xiao
Department of Electrical Engineering and Computer Science, Northwestern University, e-mail: xyz.xiao@gmail.com

Mihaela van der Schaar
Oxford – Man Institute of Quantitative Finance (OMI) and the Department of Engineering Science, University of Oxford, e-mail: mihaela.vanderschaar@eng.ox.ac.uk

1 Introduction

Video applications, such as video streaming, video conferencing, remote teaching, surveillance, have become the major applications deployed over wireless networks. Video applications are bandwidth-intensive and delay-sensitive, and hence require efficient allocation of spectrum resources among the users, and efficient scheduling of each user's video packets based on its allocated resources.

One promising physical-layer technology to improve the spectrum efficiency is cognitive radio. In cognitive radio, secondary users (SUs) can utilize the idle spectrum when primary users (PUs) are inactive. Due to the potential of high spectrum efficiency, cognitive radio is a promising candidate for the physical-layer technology of video applications. Although cognitive radio is promising, there are several challenges in efficiently deploy video applications over cognitive radio networks.

First, video applications are delay-sensitive, namely the video packets have to be received before strict deadlines for successful decoding. Therefore, video applications require not only high spectrum efficiency, but also efficient scheduling of video packets.

In addition, there is interdependency across video packets, which makes the issue of delay sensitivity more challenging. Specifically, the successful decoding of some video packets may depend on the successful decoding of others. In other word, even if a video packet is received before its deadline, it may not be decoded if some packets that it depends on were not received by deadlines. This interdependency results in temporal coupling of packet scheduling decisions. Therefore, we need *foresighted* video packet scheduling that aims at maximizing the *long-term* video quality, instead of instantaneous video quality.

Moreover, the delay sensitivity and interdependency mentioned above require not only efficient packet scheduling, but also efficient allocation of spectrum resources. More specifically, we need *foresighted* resource allocation among the users in the network, in order to maximize the long-term video quality.

Furthermore, the unknown dynamic enviroment requires users to make resource allocation and packet scheduling decisions while *learning* the unknown dynamics. Here the unknown dynamics include incoming video traffic, the channel quality, and the PU activities.

Finally, the users in cognitive radio networks are autonomous, and may aim to maximize their own video quality (i.e., they are *self-interested*). It is more challenging to design efficient resource allocation mechanism for self-interested users, because they may misreport their information (e.g., their video traffic and channel quality). In this case, the resource allocation mechanism has to be *strategy-proof*.

In this chapter, we introduce our solutions to the aforementioned challenges in multi-user wireless video transmission over cognitive radio networks. Specifically, we introduce a framework to design foresighted resource allocation mechanisms and packet scheduling algorithms [1]–[7]. The introduced resource allocation mechanisms allow wireless users to optimize while learning the unknown dynamics in the environment (e.g., incoming traffic, channel qualilty, primary user activities). We also describe variations of the mechanisms that are suitable for self-interested users

[1]–[5]. These mechanisms ensure efficient video resource allocation even when the users are self-interested and aim to maximize their individual video quality. The foresighted resource allocation mechanisms introduced in this chapter are built upon our theoretical advances in multi-user Markov decision processes, reinforcement learning, and dynamic mechanism design.

The rest of this chapter is organized as follows. We give a literature review in Section 2. We introduce the system model in Section 3, and then formulate the design problem in Section 4. We describe the introduced solutions in Section 5. We also briefly describe the strategy-proof variations of the solutions in Section 6. Finally, we conclude the chapter in Section 7.

2 Related Work

2.1 Related Works on Video Transmission

A plethora of recent works [1]–[16] propose distinct solutions to optimize the video quality. Some works [8]–[14] assume that the users are *myopic*, namely they only maximize their *instantaneous* video quality over a given time interval without considering the impact of their actions on the *long-term* video quality. They cast the problem in a network utility maximization (NUM) framework to maximize the instantaneous joint video quality of all the users, and apply the NUM framework repeatedly when the channel conditions or video traffic characteristics change. However, since the users are optimizing their transmission decisions myopically, their long-term average performance is inferior to the performance achieved when the users are *foresighted* [15]. Some of the works considering the foresighted decision of users focus solely on a *single* foresighted user making sequential transmission decisions (e.g. packet scheduling, retransmissions etc.) [15][16]. However, these single user solutions do not discuss how to allocate resources among multiple users as well as how this allocation is impacted by and impacts the foresighted scheduling decisions of individual users. Static allocations of resources, which are often assumed in the works studying the foresighted decisions of a single user, have been shown to be suboptimal compared to the solutions that dynamically allocate resources among multiple users [6].

In contrast, our introduced solutions, based on [1]–[7], make *foresighted* resource allocation and packet scheduling decisions over *multiple* video users.

Table 1 Related works on video (the first two rows are works introduced in this chapter).

	Users	Foresighted	Learning	Strategy-Proof	Optimal
[1]–[5]	Multiple	Yes	Yes	Yes	No
[6][7]	Multiple	Yes	Yes	No	Yes
[8]–[14]	Multiple	No	No	No	No
[15][16]	Single	Yes	Yes	No	No

Table 1 summarizes the above discussions. Note that the optimality shown in the last column of Table 1 indicates whether the solution is optimal for the long-term network utility (i.e., the joint long-term video quality of all the users in the network).

2.2 Related Theoretical Frameworks

Single-user foresighted decision making in a dynamically changing environment has been studied and formulated as Markov Decision Process (MDP). Foresighted decision making in a dynamically changing environment can also be solved using the Lyapunov optimization framework [22]. However, the Lyapunov optimization framework is not able to make optimal decisions for video streaming since it disregards specific interdependency and distortion impact of video traffic [23].

Table 2 Related theoretical frameworks (the first row is works introduced in this chapter).

	Decision makers	Foresighted	Learning	Optimal
MU-MDP [6][7]	Multiple	Yes	Yes	Yes
MDP [15]	Single	Yes	Yes	No
Repeated NUM [14]	Multiple	No	No	No
Lyapunov Optimization [22]	Single	Yes	Yes	No

Table 2 summarizes the above discussions about existing theoretical frameworks.

3 General Model For Video Applications Over Cognitive Radio

We first present a general model for multi-user wireless video transmission in cognitive radio networks. Then we give an example of a commonly-used model as in [6][15][16] as an instantiation of our general model.

3.1 The General Model

We consider a cognitive radio network with a network manager indexed by 0, and a set $\mathscr{I}$ of I wireless video users, indexed by $i = 1, \ldots, I$. Time is slotted at $t = 0, 1, 2, \ldots$. In the rest of the paper, we will put the user index in the superscript and the time index in the subscript of variables. The multi-user wireless video transmission system is described by the following features.

3.1.1 States

Each user i has a finite state space S^i, from which a state s^i is realized and revealed to user i at the beginning of each time slot. The state s^i may consist of several components, such as the video traffic state and the channel state. An example of a simplified video traffic state can be the types of video frames (I, P, or B frame) available for transmission and the numbers of packets in each available video frame. Note that the video traffic in our model can come from video sequences that are either encoded in real-time, or offline and stored in the memory before the transmission. An example channel state can be the channel quality reported to the application layer by the lower layers. The network manager has a finite state space S^0 that describes the status of the resource in the network. An example resource state can be the available idle bandwidth.

3.1.2 Actions

At each state s^i, each user i chooses an action $a^i \in A^i(s^i)$. For example, an action can be how many packets within each available video frame should be transmitted. We allow the sets of actions taken under different states to be different, in order to incorporate the minimum video quality requirements that will be discussed.

3.1.3 Payoffs

Each user i has a payoff function $u^i : S^i \times A^i \to \mathbb{R}$. The payoff function u^i is concave in the action under any state. A typical payoff can be the distortion impact of the transmitted packets minus the cost of energy consumption in transmission.

3.1.4 State Transition

Each user i's state transition is Markovian, and can be denoted by $p^i(s^{i\prime}|s^i, a^i) \in \Delta(S^i)$, where $\Delta(S^i)$ is the probability distribution over the set of states.

3.1.5 Resource Constraints

Given the status of the resource (i.e. the network manager's state s^0), we can write the (linear) resource constraint as

$$f(s^0, a^1, \ldots, a^I) \triangleq f^0(s^0) + \sum_{i=1}^{I} f^i(s^0) \cdot a^i \leq 0,$$

where $f^i(s^0)$ is the coefficient under state s^0. When a^i is a vector, $f^i(s^0)$ is a vector of the appropriate length, and $f^i(s^0) \cdot a^i$ is the inner product of the two vectors.

A variety of multi-user wireless video transmission systems can be modeled as special cases of our general model. Next, we present a packet-level video transmission model as an example.

3.2 An Example Packet-Level Video Transmission Model

Packet-level video transmission models have been proposed in a variety of related works, including [16]–[6]. In the following, we briefly describe the model based on [16]–[6], and refer interested readers to [16]–[6] for more details.

We first consider a specific video user i, and hence drop the superscript before we describe the resource constraints. The video source data is encoded using an H.264 or MPEG video coder under a Group of Pictures (GOP) structure: the data is encoded into a series of GOPs, indexed by $g = 1, 2, \ldots$, where one GOP consists of N data units (DUs). Each DU $n \in \{1, \ldots, N\}$ in GOP g, denoted DU_n^g, is characterized by its size $l_n^g \in \mathbb{N}_+$ (i.e. the number of packets in it), distortion impact $q_n^g \in \mathbb{R}_+$, delay deadline $d_n^g \in \mathbb{N}_+$, and dependency on the other DUs in the same GOP. The dependency among the DUs in one GOP comes from encoding techniques such as motion estimation/compensation. In general, if DU_n^g depends on DU_m^g, we have $d_n^g \geq d_m^g$ and $q_n^g \leq q_m^g$, namely DU_m^g should be decoded before DU_n^g and has a higher distortion impact than DU_n^g [15]. Note that in the case of scalable video coding, there is no dependency among the DUs, and the following representation of the model can be greatly simplified. We will keep the dependency for generality in our exposition.

Among the above characteristics, the distortion impact q_n^g, delay deadline d_n^g, and the dependency are deterministic and fixed for the same DUs across different GOPs (e.g. $q_n^g = q_n^{g+1}$) [15][6]. As in [6], the sizes of all the DUs are independent random variables, and that the sizes of the nth DUs in different GOPs have the same distribution.

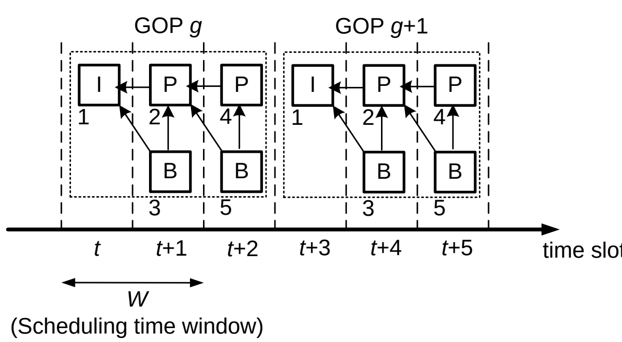

Fig. 1 Illustration of GOP (group of pictures), DU (data unit), and the context. Since the scheduling time window is $W = 2$, the contexts in different time slots are $C_t = \{DU_1^g, DU_2^g, DU_3^g\}$, $C_{t+1} = \{DU_2^g, DU_3^g, DU_4^g, DU_5^g\}$, $C_{t+2} = \{DU_4^g, DU_5^g, DU_1^{g+1}\}$, $C_{t+3} = \{DU_1^{g+1}, DU_2^{g+1}, DU_3^{g+1}\}$, and so on.

3.2.1 States

Each user's state consists of the traffic state T_t and the channel state h_t. We describe the traffic state T_t first. At time slot t, as in [16][15][6], we assume that the wireless user will only consider for transmission the DUs with delay deadlines in the range of $[t, t+W)$, where W is referred to as the scheduling time window (STW). Following the model in [15][6], at time slot t, we introduce *context* to represent the set of DUs that are considered for transmission, i.e., whose delay deadlines are within the range of $[t, t+W)$. We denote the context by $C_t = \left\{ DU_j^g | d_j^g \in [t, t+W) \right\}$. Since the GOP structure is fixed, the transition from context C_t to C_{t+1} is deterministic. An illustration of the context is given in Fig. 1.

Given the current context C_t, we let $x_{t,DU}$ denote the number of packets in the buffer associated with a DU in C_t. We denote the buffer state of the DUs in C_t by $x_t = \{x_{t,DU} | DU \in C_t\}$. The traffic state T_t at time slot t is then $T_t = (C_t, x_t)$, where the context C_t represents which DUs are available for transmission, and the buffer state x_t represents how many packets each available DU has left in the buffer .

Next we describe the channel state h_t. At each time slot t, the wireless user experiences a channel condition $h_t \in \mathscr{H}$, where $\mathscr{H}$ is the finite set of possible channel conditions. We assume that the wireless channel is slow-fading (i.e. remains the same in one time slot), and that the channel condition h_t can be modeled as a finite-state Markov chain [25].

In summary, the state of a user at each time slot t is $s_t = (C_t, x_t, h_t)$, which includes the current context, buffer state and channel state.

3.2.2 (Packet Scheduling) Actions

At each time slot t, the user decides how many packets should be transmitted from each DU in the current context. The decision is represented by $a_t(s_t) = \{y_{t,DU} | DU \in C_t, y_{t,DU} \in [0, x_{t,DU}]\}$, where $y_{t,DU}$ is the amount of packets transmitted from the DU.

3.2.3 Payoffs

As in [15], we consider the following *instantaneous payoff* at each time slot t:[1]

$$u(s_t, a_t) = \sum_{DU \in C_t} q_{DU} y_{t,DU} - \beta \cdot \rho\left(h_t, \|a_t\|_1\right), \tag{1}$$

where the first term $\sum_{DU \in C_t} q_{DU} y_{t,DU}$ is the *instantaneous video quality*, namely the distortion reduction obtained by transmitting the packets from the DUs in the current context, and the second term $\beta \cdot \rho\left(h_t, \|a_t\|_1\right)$ represents the disutility of the

[1] The payoff function can be easily extended within our framework to include additional features in the model. For example, when there are packet loss, we can modify the first term to be the expected distortion reduction given the packet loss rate, or modify the second term to consider the additional energy consumption associated with packet retransmission.

energy consumption by transmitting the packets. Since the packet scheduling action a_t is a vector with nonnegative components, we have $\|a_t\|_1 = \sum_{DU\in C_t} y_{t,DU}$, namely $\|a_t\|_1$ is the total number of transmitted packets. As in [15], the energy consumption function $\rho(h,\|a\|_1)$ is assumed to be convex in the total number of transmitted packets $\|a\|_1$ given the channel condition h. An example of such a function can be $\rho(h,\|a\|_1) = \sigma^2(e^{2\|a\|_1 b}-1)/h$, where b is the number of bits in one packet [24]. The payoff function is a tradeoff between the distortion reduction and the energy consumption, where the relative importance of energy consumption compared to distortion reduction is characterized by the tradeoff parameter $\beta > 0$. In the simulation, we will set different values for β to illustrate the tradeoff between the distortion reduction and energy consumption.

3.2.4 The Resource Constraint

Suppose that the users access the channels in a FDMA (frequency-division multiple access) manner. The total bandwidth B is shared by the users. We assume that each user i uses adaptive modulation and coding (AMC) based on its channel condition. In other words, each user i chooses a data rate r_t^i under the channel state h_t^i. Note that the rate selection is done by the physical layer and is not a decision variable in our framework. Then as in [?][8], we have the following resource constraint:

$$\sum_{i=1}^{I} \frac{\|a_t^i\|_1 b}{r_t^i(h_t^i)} \leq B, \tag{2}$$

where $\frac{\|a_t^i\|_1\cdot b}{r_t^i(h_t^i)}$ is the bandwidth needed for transmitting the amount $\|a_t^i\|_1\cdot b$ of bits given the data rate $r_t^i(h_t^i)$.

In this model, the network manager's state s^0 is then the collection of channel states, namely $s^0 = (h^1,\dots,h^I)$. The information about the channel states is fed back from the users to the network manager. We can write the constraint compactly as the linear constraint $f(s_t^0,a_t^1,\dots,a_t^N) \leq 0$ with $f^i(s_t^0) = \frac{b}{r_t^i(h_t^i)}, i = 1,\dots,I$ and $f^0(s_t^0) = -B$.

4 The Design Problem

Each user makes decisions based on its state s_t. Hence, each user i's strategy can be defined as a mapping $\pi^i : S^i \to \cup_{s^i} A^i(s^i)$, where $A^i(s^i)$ is the set of actions available under state s^i. We allow the set of available actions to depend on the state, in order to capture the minimum video quality guarantee. For example, at any time, user i has a minimum distortion impact reduction requirement D^i, formulated as

$$A^i(s_t^i) = \left\{a_t^i : \sum_{DU\in C_t^i} q_{DU}\cdot y_{t,DU}^i \geq D^i\right\}.$$

The users aim to maximize their expected long-term payoff. Given its initial state s_0^i, each user i's strategy π^i induce a probability distribution over the sequences of states $s_1^i, s_2^i, \ldots$, and hence a probability distribution over the sequences of instantaneous payoffs $u_0^i, u_1^i, \ldots$. Taking expectation with respect to the sequences of payoffs, we have user i's *long-term payoff* given the initial state as

$$U^i(\pi^i|s_0^i) = \mathbb{E}\left\{(1-\delta)\sum_{t=0}^{\infty}\left(\delta^t \cdot u_t^i\right)\right\}, \tag{3}$$

where $\delta \in [0,1)$ is the discount factor.

The design problem can be formulated as

$$\begin{array}{ll} \max_{\pi^1,\ldots,\pi^I} & \sum_{s_0^1,\ldots,s_0^I} \sum_{i=1}^{I} U^i(\pi^i|s_0^i) \\ s.t. & \text{minimum video quality guarantee :} \\ & \pi^i(s^i) \in A^i(s^i),\ \forall i, s^i, \\ & \text{resource constraint :} \\ & f(s^0, \pi^1(s^1), \ldots, \pi^I(s^I)) \leq 0,\ \forall s^0. \end{array} \tag{4}$$

Note that the design problem (4) is a weakly-coupled MU-MDP as defined by [26]. It is a MU-MDP because there are multiple users making foresighted decisions. The MU-MDP is coupled, because the users influence each other through the resource constraints (namely the choice of one user's action depends on the choices of the other users). However, it is weakly-coupled, because the coupling is through the resource constraints only, and because one user's *instantaneous* payoff $u^i(s^i, a^i)$ is not affected by the other users' actions a^j. It is this weak coupling that enables us to decompose the multi-user problem into multiple single-user problems through prices. Such a decomposition of weakly-coupled MU-MDPs has been studied in a general setting [26] and in wireless video transmission [6], both adopting a dual decomposition approach based on *uniform* price (i.e. the same Lagrangian multiplier for the resource constraints under all the states).

Note also that we sum up the network utility $\sum_{i=1}^{I} U^i(\pi^i|s_0^i)$ under all the initial states $(s_0^1, \ldots, s_0^I)$. This can be interpreted as the expected network utility when the initial state is uniformly distributed. The optimal stationary strategy profile that maximizes this expected network utility will also maximize the network utility given any initial state.

The design problem (4) is very challenging, and has never been solved optimally. To better understand this, let us assume that a central controller would exist which knows the complete information of the system (i.e. the states, the state transitions, the payoff functions) at each time step. Then, this central controller can solve the above problem (4) as a centralized single-user MDP (e.g. using well-known Value Iteration or Policy Iteration methods) and obtain the solution to the design problem $\boldsymbol{\pi}^\star$ and the optimal value function $U^\star$. However, the multi-user wireless video system we discussed is inherently informationally-decentralized and there is no entity in the network that possesses the complete information. Moreover, the computational complexity of solving (4) by a single entity is prohibitively high. Hence, our goal is to develop an optimal decentralized algorithm that converges to the optimal solution.

5 Optimal Foresighted Video Transmission

In this section, we show how to determine the optimal foresighted video transmission policies. We propose an algorithm that allows each entity to make decisions based on its local information and the limited information exchange between the BS and the users. Specifically, in each time slot, the BS sends resource prices to each user, and the users send their total numbers of packets to transmit to the BS. The BS keeps updating the resource prices based on the resource usage by the users, and obtains the optimal resource prices based on which the users' optimal individual decisions achieve the optimal network utility.

5.1 Decoupling of The Users' Decision Problems

Each user aims to maximize its own long-term payoff $U^i(\pi^i|s_0^i)$ subject to the constraints. Specifically, given the other users' strategies $\boldsymbol{\pi}^{-i} = (\pi^1, \ldots, \pi^{i-1}, \pi^{i+1}, \ldots, \pi^I)$ and states $\boldsymbol{s}^{-i} = (s^1, \ldots, s^{i-1}, s^{i+1}, \ldots, s^I)$, each user i solve the following long-term payoff maximization problem:

$$\begin{aligned} \pi^i = \arg\max_{\tilde{\pi}^i} \; & U_i(\tilde{\pi}^i|s_0^i) \\ s.t. \quad & \tilde{\pi}^i(s^i) \in A^i(s^i), \; \forall s^i, \\ & f(s^0, \tilde{\pi}^i(s^i), \boldsymbol{\pi}^{-i}(\boldsymbol{s}^{-i})) \leq 0. \end{aligned} \tag{5}$$

Assuming that the user knows all the information (i.e. the other users' strategies $\boldsymbol{\pi}^{-i}$ and states $\boldsymbol{s}^{-i}$), user i's optimal value function should satisfy the following:

$$\begin{aligned} V(s^i) = & \max_{a^i \in A^i(s^i)} \; (1-\delta)u^i(s^i, a^i) + \delta \sum_{s^{i\prime}} p^i(s^{i\prime}|s^i, a^i) V(s^{i\prime}) \\ & s.t. \quad f(s^0, a^i, \boldsymbol{\pi}^{-i}(\boldsymbol{s}^{-i})) \leq 0. \end{aligned} \tag{6}$$

Note that the above equations would be the Bellman equation, if user i knew the other users' strategies $\boldsymbol{\pi}^{-i}$ and states $\boldsymbol{s}^{-i}$ and the BS's state s^0 (i.e. the channel states of all the users). However, such information is never known to a particular user. Without such information, one user cannot solve the decision problem above because the resource constraint contains unknown variables. Hence, we need to separate the influence of the other users' decisions from each user's decision problem.

One way to decouple the interaction among the users is to remove the resource constraint and add it as a penalty to the objective function. Denote the Lagrangian multiplier (i.e. the “price”) associated with the constraint under state s^0 as $\lambda^0(s^0)$. Then the penalty at state s^0 is

$$-\lambda^0(s^0)\cdot f(s^0,a^1,\dots,a^I) = -\lambda^0(s^0)\cdot \textstyle\sum_{i=1}^I f^i(s^0)\cdot a^i.$$

Since the term $-\lambda^0(s^0)\cdot\sum_{j\neq i} f^j(s^0)\cdot a^j$ is a constant for user i, we only need to add $-\lambda^0(s^0)\cdot f^i(s^0)\cdot a^i$ to each user i's objective function. We define $\lambda^i(s^0)\triangleq\lambda^0(s^0)\cdot f^i(s^0)$. Then we can rewrite user i's decision problem as

$$\begin{aligned}\tilde{V}^{\lambda^i(s^0)}(s^i) = \max_{a^i\in A^i(s^i)} &(1-\delta)\left[u^i(s^i,a^i)-\lambda^i(s^0)\cdot a^i\right]\\ &+\delta\cdot\textstyle\sum_{s_i'}\left[p^i(s^{i\prime}|s^i,a^i)\tilde{V}^{\lambda^i(s^0)}(s^{i\prime})\right]. \end{aligned} \tag{7}$$

By contrasting (7) with (6), we can see that given the price λ^i, each user can make decisions based only on its local information since the resource constraint is eliminated. Note, importantly, that the above decision problem (7) for each user i is different from that in [6] with uniform price. This can be seen from the term $\lambda^i(s^0)\cdot a^i$ in (7), where the price $\lambda^i(s^0)$ is user-specific and depends on the state, while the uniform price in [6] is a constant λ. The decision problem (7) is also different from the subproblem resulting from dual decomposition in NUM, because it is a foresighted optimization problem that aims to maximize the long-term payoff. This requires a different method to calculate the optimal Lagrangian multiplier $\lambda^i(s^0)$ than that in NUM.

5.2 Optimal Decentralized Video Transmission Strategy

For the general model described in Section 3-A, we propose an algorithm used by the BS to iteratively update the prices and by the users to update their optimal strategies. The algorithm will converge to the optimal prices and the optimal strategy profile that achieves the minimum total system payoff $U^\star$. The algorithm is described in Table 3.

Theorem 1. *The algorithm in Table 3 converges to the optimal strategy profile, namely*

$$\lim_{t,k\to\infty}\left|\textstyle\sum_{s_t^1,\dots,s_t^I}\sum_{i=1}^I U_i(\pi^{i,\lambda_k^i}|s_t^i)-U^\star\right| = 0\,.$$

Proof. See the appendix in [7].

We illustrate the the BS's and users' updates and their information exchange in one time slot in Fig. 2. At the beginning of each time slot t, the BS and the users exchange information to compute the optimal resource price and the optimal actions to take. Specifically, in each iteration k, the BS updates the resource price λ_k^0. Then based on the user-specific resource price λ_k^i, each user i solves for the optimal individual strategy π^{i,λ_k^i}, and sends the BS its resource request $f^i \cdot \pi^{i,\lambda_k^i}(s_t^i)$. Then the BS updates the prices based on the users' resource requests using the stochastic subgradient method, which can be performed easily. The difference from the dual decomposition in NUM is that each user's decision problem in our work is a foresighted optimization problem aiming to maximize the long-term, instead of instantaneous, payoff. Our algorithm is also different from the algorithm in [6] in that we have different prices under different states.

From Fig. 2, we can clearly see what information (namely resource prices λ_k^0 and resource requests $f^i \cdot \pi^{i,\lambda_k^i}(s_t^i)$) is exchanged. The amount of information exchange is small ($O(I)$), compared to the amount of information required by each user to solve the decision problem (6) directly ($\prod_{j\neq i} |S_i|$ states plus the strategies $\boldsymbol{\pi}_{-i}$). In other words, the algorithm enables the entities to exchange a small amount ($O(I)$) of information and reach the optimal video transmission strategy that achieves the same performance as when each entity knows the complete information (i.e. the states and the strategies of all the entities) about the system.

Table 3 Distributed algorithm to compute the optimal strategy at time t.

Input: Performance loss tolerance ε
Initialization: Set $k=0$, $\lambda_k^0 = 0$.
Each user i observes s_t^i, the BS observes s_t^0
repeat
Each user i solves the decoupled decision problem (7) to obtain π^{i,λ_k^i}
Each user i submits its resource request $f^i(s_t^0)\cdot\pi^{i,\lambda_k^i}(s_t^i)$
The BS updates the prices (stochastic subgradient update):
$\lambda_{k+1}^i(s_t^0) = \lambda_k^i(s_t^0) + \frac{1}{k+1} f(s_t^0, \pi^{1,\lambda_k^1}(s_t^1), \cdots, \pi^{I,\lambda_k^I}(s_t^I))$
until $\|\lambda_{k+1}^i(s_t^0) - \lambda_k^i(s_t^0)\| \leq \varepsilon$
Output: optimal price λ_k^0, optimal strategies $\{\pi^{i,\lambda_k^i}\}_{i=1}^I$

5.3 *Optimal Packet Scheduling*

In the previous subsection, we propose an algorithm of optimal foresighted resource allocation and packet scheduling for the general video transmission model described in Section 3-A. In the algorithm, each user's packet scheduling decision is obtained by solving the Bellman equation (7) (see Table 3). The Bellman equation (7) can be solved by a variety of standard techniques such as value iteration. However, the computational complexity of directly applying value iteration may be high, because each user's state contains the information of all DUs and thus each user's state space can be very large. In the following, we show that for the specific model described in Section 3-B, we can greatly simplify the packet scheduling decision problem. The key simplification comes from the decomposition of each user's packet scheduling problem with multiple DUs into multiple packet scheduling problems with single DU.

In this way, we can greatly reduce the number of states in each single-DU packet scheduling problem, such that the total complexity of packet scheduling grows linearly, instead of exponentially without decomposition, with the number of DUs.

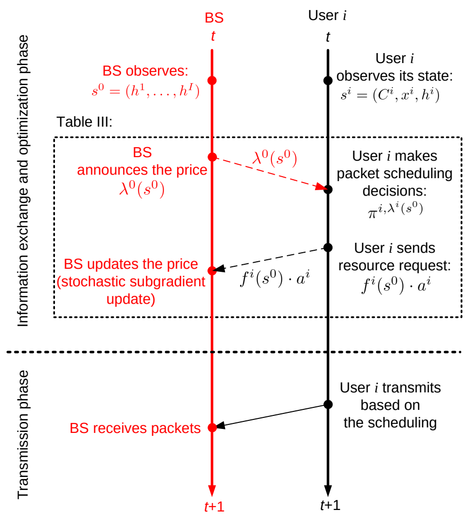

Fig. 2 Illustration of the interaction between the BS and user i (i.e. their decision making and information exchange) in one period.

The decomposition closely follows the decomposition of multiple-DU packet scheduling problems proposed in [15]. The only difference is that the decision problem (7) in our work has an additional term $-\lambda^i(s^0) \cdot a^i$ due to the price, while such a term does not exist in [15] because the *single*-user packet scheduling problem is considered in [15].

Lemma 1 (Structural Result). *Suppose $DU_1 \in C_t$ and $DU_2 \in C_t$. If DU_2 depends on DU_1, we should schedule the packets of DU_1 before scheduling the packets of DU_2.*

Although Lemma 1 is straightforward, it greatly simplifies the scheduling problem because we can now take advantage of the partial ordering of the DUs. However, this still does not solve the scheduling decision for the DUs that are not dependent on each other. Next, we provides the algorithm of optimal packet scheduling in Table 4. The algorithm decomposes the multiple-DU packet scheduling problem into a sequence of single-DU packet scheduling problems, and determines how many packets to transmit for each DU sequentially. This greatly reduces the total computational complexity (which is linear in the number of DUs) compared to solving the multiple-DU packet scheduling problem directly (in which the number of states grows exponentially withe number of DUs). The algorithm is similar to [15, Algorithm 2]. The only difference is the term $-\lambda^i(s^0) \cdot a^i$.

5.4 Learning Unknown Dynamics

In practice, each entity may not know the dynamics of its own states (i.e., its own state transition probabilities) or even the set of its own states. When the state dynamics are not known a priori, each entity cannot solve their decision problems using the distributed algorithm in Table 3. In this case, we can adapt the online learning algorithm based on post-decision state (PDS) in [15], which was originally proposed for single-user wireless video transmission, to the considered deployment scenario.

The main idea of the PDS-based online learning is to learn the post-decision value function, instead of the value function. Each user i's post-decision value function is defined as $\tilde{U}^i(\tilde{x}^i, \tilde{h}^i)$, where $(\tilde{x}^i, \tilde{h}^i)$ is the post-decision state. The difference from the normal state is that the PDS $(\tilde{x}^i, \tilde{h}^i)$ describes the status of the system *after* the scheduling action is made but before the DUs in the next period arrive. Hence, the relationship between the PDS and the normal state is

$$\tilde{x}^i = x^i - a^i,\ \tilde{h}^i = h^i.$$

Then the post-decision value function can be expressed in terms of the value function as follows:

$$\tilde{U}^i(\tilde{x}^i, \tilde{h}^i) = \sum_{x^{i\prime}, h^{i\prime}} p^i(x^{i\prime}, h^{i\prime} | \tilde{x}^i + a^i, \tilde{h}^i) \cdot \tilde{V}^i(x^{i\prime}, \tilde{h}^i).$$

In PDS-based online learning, the normal value function and the post-decision value function are updated in the following way:

$$\begin{aligned} V^i_{k+1}(x^i_k, h^i_k) &= \max_{a^i}\ (1-\delta) \cdot u^i(x^i_k, h^i_k, a^i) \\ &\quad + \ \delta \cdot U^i_k(x^i_k + (a^i - l^i_k), h^i_k), \\ U^i_{k+1}(x^i_k, h^i_k) &= (1 - \frac{1}{k}) U^i_k(x^i_k, h^i_k) \\ &\quad + \frac{1}{k} \cdot V^i_k(x^i_k - (a^i - l^i_k), h^i_k). \end{aligned}$$

We can see that the above updates do not require any knowledge about the state dynamics. In particular, we propose the decomposed optimal packet scheduling with PDS-based learning in Table 5. Note that the difference between the learning algorithm in Table 5 with the algorithm assuming statistic knowledge in Table 4 is that we use the post-decision state value function instead of the normal value function. It is proved in [15] that the PDS-based online learning will converge to the optimal value function. Hence, the distributed packet scheduling and resource allocation solution in Table 3 can be modified by letting each user perform the packet scheduling using the PDS-based learning in Table 5.

Table 4 The optimal packet scheduling algorithm.

Input: Directed acyclic graph given the current context: $DAG(C_t)$

Initialization: Set $DAG_1 = DAG(C_t)$.

For $k = 1, \ldots, |C_t|$

$$DU_k = \arg \max_{DU \in root(DAG_k)} \max_{0 \le y \le x_{t,DU}} (1-\delta)\left[q_{DU} \cdot y - \lambda_k^i(s^0) \cdot y\right] + \delta \cdot \sum_{s^{i\prime}} \left[p^i(s^{i\prime}|s^i, a_{f,t}) \tilde{V}^{i,\lambda^{i,(k)}(s^0)}(s^{i\prime})\right]$$

$$y^*_{t,DU_k} = \arg \max_{0 \le y \le x_{t,DU_k}} (1-\delta)\left[q_{DU_k} \cdot y - \lambda_k^i(s^0) \cdot y\right] + \delta \cdot \sum_{s^{i\prime}} \left[p^i(s^{i\prime}|s^i, a_{f,t}) \tilde{V}^{i,\lambda^{i,(k)}(s^0)}(s^{i\prime})\right]$$

$$DAG_{k+1} = DAG_k \setminus \{DU_k\}$$

End For

Table 5 The optimal decomposed packet scheduling algorithm with PDS-based learning.

Input: Directed acyclic graph given the current context: $DAG(C_t)$

Initialization: Set $DAG_1 = DAG(C_t)$.

For $k = 1, \ldots, |C_t|$

$$DU_k = \arg \max_{DU \in root(DAG_k)} \max_{0 \le y \le x_{t,DU}} (1-\delta)\left[q_{DU} \cdot y - \lambda_k^i(s^0) \cdot y\right] + \delta \cdot U_{DU}(C_t, x_{t,DU} - y, h_t)$$

$$y^*_{t,DU_k} = \arg\max_{0 \le y \le x_{t,DU}} (1-\delta)\left[q_{DU} \cdot y - \lambda_k^i(s^0) \cdot y\right] + \delta \cdot U_{DU}(C_t, x_{t,DU} - y, h_t)$$

$$DAG_{k+1} = DAG_k \setminus \{DU_k\}$$

End For

6 Strategy-Proof Resource Allocation Mechanisms

When the users are self-interested, they may not follow the solutions introduced in Section 5. In particular, they may not be truthful in the message exchange with the network manager. There are several ways of designing strategy-proof resource allocation mechanisms, based on pricing [3] and mechanism design [1]–[5]. In this section, we describe a representative framework based on auctions [4][5].

The auction-based resource allocation mechanism is illustrated in Fig. 3. The basic procedure at each time slot is described as follows:

1. The network manager announces the total amount of available resources (e.g., state s^0).
2. The SUs submit bids of how much resources they are willing to get.
3. Based on SUs' bids, the network manager determines the resource allocation and the payments required from SUs.
4. Based on the allocated resources, the SUs schedule their video packets.

As we can see, most elements (e.g., states, rewards) in auction-based mechanisms are the same as in the general model in Section 3. Here we list some key features of the auction-based mechanism.

- In the auction-based mechanism, the SUs' actions consist of two types of actions, internal actions and external actions. The internal actions, denoted by b_i^t in Fig. 3,

are the packet scheduling actions described in Section 3. The external actions are unique in the auction-based mechanism. Specifically, the external action is the amount of resources each SU wants to get (i.e., their bids).
- In the auction-based mechanism, the network manager directly allocates the resources to the SUs and announces the payments required from the SUs. This is different from the mechanism in Section 3, where the network manager announces the prices and the SUs determine the resource allocation based on the prices.

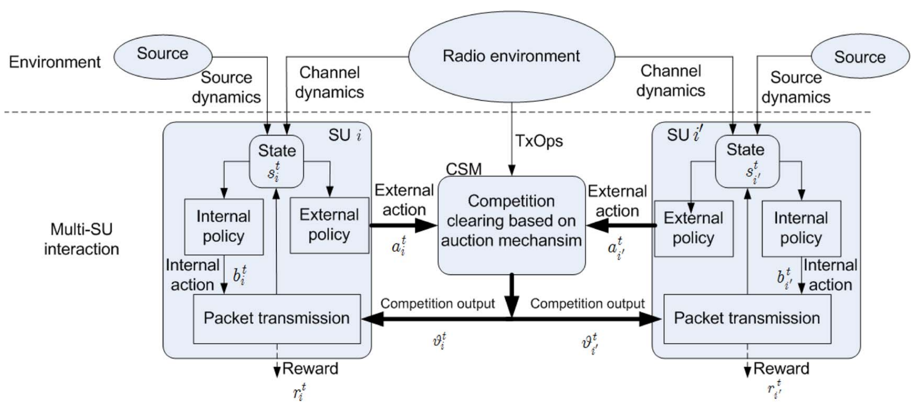

Fig. 3 Illustration of strategy-proof resource allocation mechanism based on auctions.

We refer interested readers to [4][5] for detailed descriptions and theoretical results of the auction-based mechanisms.

7 Conclusion

In this chapter, we introduce the optimal foresighted resource allocation and packet scheduling for multi-user wireless video transmission over cognitive radio networks. The introduced solution achieves the optimal long-term video quality subject to each user's minimum video quality guarantee, by dynamically allocating resources among the users and dynamically scheduling the users' packets while taking into account the dynamics of the video traffic and channel states. We develop a low-complexity algorithm that can be implemented by the network manager and the users in an informationally-decentralized manner and converges to the optimal solution. We also introduce strategy-proof variations of our solutions for self-interested users.

References

1. A. Fattahi, F. Fu, M. van der Schaar, and F. Paganini, “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.
2. F. Fu and M. van der Schaar, “Noncollaborative resource management for wireless multimedia applications using mechanism design,” *IEEE Trans. Multimedia*, vol. 9, no. 4, pp. 851–868, Jun. 2007.
3. F. Fu, T. M. Stoenescu, and M. van der Schaar, “A pricing mechanism for resource allocation in wireless multimedia applications,” *IEEE Journal of Selected Topics in Signal Process., Special Issue on Network-Aware Multimedia Process. and Communications*, vol. 1, no. 2, pp. 264–279, Aug. 2007.
4. F. Fu and M. van der Schaar, “Learning to compete for resources in wireless stochastic games,” *IEEE Trans. Veh. Tech.*, vol. 58, no. 4, pp. 1904–1919, May 2009.
5. M. van der Schaar and F. Fu, “Spectrum access games and strategic learning in cognitive radio networks for delay-critical applications,” *Proc. of IEEE, Special issue on Cognitive Radio*, vol. 97, no. 4, pp. 720–740, Apr. 2009.
6. F. Fu and M. van der Schaar, “A systematic framework for dynamically optimizing multi-user video transmission,” *IEEE J. Sel. Areas Commun.*, vol. 28, no. 3, pp. 308–320, Apr. 2010.
7. Y. Xiao and M. van der Schaar, “Optimal foresighted multi-user wireless video,” *IEEE Journal of Selected Topics in Signal Process., Special issue on Visual Signal Processing for Wireless Networks*, vol. 9, no. 1, pp. 89–101, Feb. 2015.
8. Y.-J. Yu, P.-C. Hsiu, and A.-C. Pang, “Energy-efficient video multicast in 4G wireless systems,” *IEEE Trans. Mobile Comput.*, vol. 11, no. 10, pp. 1508–1522, Oct. 2012.
9. M. van der Schaar, Y. Andreopoulos, and Z. Hu, “Optimized scalable video streaming over IEEE 802.11 a/e HCCA wireless networks under delay constraints,” *IEEE Trans. Mobile Comput.*, vol. 5, no. 6, pp. 755–768, Jun. 2006.
10. G-M. Su, Z. Han, M. Wu, and K.J.R. Liu, “Joint uplink and downlink optimization for real-time multiuser video streaming over WLANs,” *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 2, pp. 280–294, Aug. 2007.
11. X. Zhang and Q. Du, “Cross-layer modeling for QoS-driven multimedia multicast/broadcast over fading channels in mobile wireless networks,” *IEEE Commun. Mag.*, vol. 45, no. 8, pp. 62–70, Aug. 2007.
12. J. Huang, Z. Li, M. Chiang, and A.K. Katsaggelos, “Joint source adaptation and resource allocation for multi-user wireless video streaming,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 5, pp. 582–595, May 2008.
13. E. Maani, P. Pahalawatta, R. Berry, T. N. Pappas and A. K. Katsaggelos, “Resource allocation for downlink multiuser video transmission over wireless lossy networks,” *IEEE Trans. on Image Process.*, vol. 17, no. 9, 1663–1671, Sep. 2008.
14. X. Ji, J. Huang, M. Chiang, G. Lafruit, and F. Catthoor, “Scheduling and resource allocation for SVC streaming over OFDM downlink systems,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 10, pp. 1549–1555, Oct. 2009.
15. F. Fu and M. van der Schaar, “Structural solutions for dynamic scheduling in wireless multimedia transmission,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 5, pp. 727–739, May 2012.
16. P. Chou and Z. Miao, “Rate-distortion optimized streaming of packetized media,” *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 390–404, Apr. 2006.
17. S. Khan, M. Sgroi, Y. Peng, E. Steinbach, W. Kellerer, “Application-driven cross layer optimization for video streaming over wireless networks,” IEEE Commun. Mag. , pp. 122–130, Jan. 2006.
18. C. De Vleeschouwer and P. Frossard, “Dependent packet transmission policies in rate-distortion optimized media scheduling,” *IEEE Trans. Multimedia*, vol. 9, no. 6, pp. 1241–1258, Oct. 2007.

19. Z. Li, F. Zhai and A. Katsaggelos, "Joint video summarization and transmission adaptation for energy-efficient wireless video streaming," *EURASIP J. on Adv. in Signal Processing, special issue on Wireless Video*, 2008.
20. N. Tizon and B. Pesquet-Popescu, "Scalable and media aware adaptive video streaming over wireless networks," *EURASIP Journal on Advances in Signal Processing*, Volume 2008 (2008), Article ID 218046, 11 pages.
21. H. Wang and A. Ortega, "Rate-distortion optimized scheduling for redundant video representations," *IEEE Trans. Image Process.*, vol. 18, no. 2, pp. 225–240, Feb. 2009.
22. W. Chen, M. J. Neely, and U. Mitra, "Energy-efficient transmission with individual packet delay constraints," *IEEE Trans. Inform. Theory*, vol. 54, no. 5, pp. 2090–2109, May 2008.
23. F. Fu and M. van der Schaar, "Structure-aware stochasticcontrol for transmission scheduling," *IEEE Trans. Veh. Tech.* vol. 61, no. 9, pp. 3931–3945, Nov. 2012.
24. D. Bertsekas and R. Gallager, *Data Networks*. Upper Saddle River, NJ: Prentice-Hall, 1987.
25. Q. Zhang and S. A. Kassam, "Finite-state Markov model for Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 47, no. 11, pp. 1688–1692, Nov. 1999.
26. J. Hawkins, "A Lagrangian decomposition approach to weakly coupled dynamic optimization problems and its applications," PhD Dissertation, MIT, Cambridge, MA, 2003.
27. *IEEE Standard for Local and Metropolitan Area Networks, Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer*, IEEE Computer Society, April. 16, 2012, online at: http://standards.ieee.org/getieee802/download/802.15.4e-2012.pdf
28. H. Kushner and G. Yin, Stochastic Approximation and Recursive Algorithms and Applications. Springer, 2003.