Rate-Control and Multi-Channel Scheduling for Wireless Live Streaming with Stringent Deadlines

Shizhen Zhao and Xiaojun Lin

School of ECE, Purdue University, West Lafayette, IN, USA

Email: {zhao147, linx}@purdue.edu

Abstract—SVC-based live video-streaming in multi-channel wireless networks leads to a challenging joint rate-control and scheduling problem with stringent deadline constraints. Traditional utility-based approaches often did not explicitly account for deadlines. In this paper, we explicitly account for deadlines and study the problem of optimizing the total reward from packets meeting their deadlines in a modern 4G OFDM system. Motivated by a heuristic utility-based approach, we propose a class of threshold-based rate-control and wireless scheduling policies that can respect the deadline constraints and approach the optimal system reward asymptotically as the system size increases. We also propose a distributed realization of our threshold-based policies that can be easily implemented in practical scenarios. We substantiate the result via both analysis and simulation.

I. INTRODUCTION

Live-streaming is becoming one of the most popular applications in wireless networks. According to a recent industry whitepaper, video is expected to consume two-thirds of the overall mobile data traffic by 2016 [1]. Compared to other types of mobile data applications, supporting wireless live-streaming is much more challenging due to the time-sensitivity of the video packets and the time-varying nature of the wireless channel. If not controlled properly, a significant fraction of the live-streaming packets may not meet their deadlines when the channel condition is poor. The overall quality of the live-streaming service will then degrade significantly.

In the literature, SVC (scalable video coding) has been proposed as a way to improve live-streaming performance against channel uncertainty. An SVC video-encoder produces a layered video stream which contains a base layer and several enhancement layers for each video frame. When the channel condition is poor, the sender can reduce the data rate by transmitting the video at a lower layer. Such adaptivity can potentially improve the overall performance of live-streaming under time-varying wireless channels [2].

In real networks, a large number of live-streaming users may compete for the wireless channel simultaneously. The use of SVC in such an environment leads to an interesting and challenging problem of joint rate-control and scheduling with stringent deadline constraints. Note that if the video rate is not controlled, there have been a number of studies on the design of wireless scheduling mechanisms to optimize the system performance subject to deadline constraints [3][4][5][6][7]. However, with SVC the video rate may change according to the state of the network, which in turn will affect the dynamics of the scheduling mechanism. Thus, there is a tight close-loop coupling between rate-control and scheduling, whose impact on the delay performance is not captured in the above studies. On the other hand, joint rate control and scheduling *without deadline constraints* has been studied using an optimization approach (see, e.g., [8]). Specifically, one may model the rate-adaptivity using a rate-distortion [9] or utility [10] function, and then design joint control algorithms to optimize the total system utility. However, these studies do not account for the deadline requirement. As we will see in Section III, such an approach may lead to extremely poor performance for live-streaming services requiring stringent deadlines.

In this work, we directly account for deadlines in a joint rate-control and scheduling problem. Specifically, for SVC, we assign different rewards to packets from different layers according to the respective video quality. Further, we assign each packet a hard deadline when it must be received by the receiver. We then study the problem of how to jointly adjust the video rates and the scheduling decisions in order to maximize the total reward obtained from those packets meeting their deadlines. In addition, we note that for practical implementation, it is highly desirable that the solutions to this problem possess the following decentralized structure. Recall that rate-control for SVC naturally occurs at the user side[1], while multi-user scheduling occurs at a base-station. It would be highly desirable that there is only a minimal amount of information exchange between the users and the base station. Specifically, the base-station ideally does not know extensive application-level information from the users, and each user does not need to know extensive application-level information from other users. Our goal in this paper is to develop efficient joint rate-control and scheduling algorithms under stringent deadline constraints that possess this decentralized structure.

Designing such a decentralized joint rate-control and scheduling algorithm with deadline constraints is a challenging problem, and existing results in the literature are quite limited. Many of the existing studies focus on a single user. For example, Kang et al. [16] proposed a threshold based policy for video packet scheduling by a single user, but there is no study of optimality. If the statistics of the channel are known, one may formulate the single-user problem (with deadline) as an MDP (Markov Decision Problem) [13]. However, The MDP-based approach incurs prohibitively high complexity,

---

[1]For example, in DASH (dynamic adaptive streaming over http) [11], users will request a certain number of video layers based their channel condition

and thus is difficult to generalize to the multi-user case. Some recent works [7][15] studied the multi-user problem with deadline constraints. The analysis of [7] is based on the assumption that all packets are generated at the beginning of each period, and expire at the end of the same period. Such a period structure is difficult to maintain in a decentralized setting where packets from multiple sources arrive at the base station with varying delay. The algorithm proposed in [15] is a heuristic algorithm, which does not provide any performance guarantee. Further, the algorithm in [15] requires the base station to know substantial application-level knowledge of all the users (e.g., the rate distortion function, and the reward of each packet), which may be difficult to implement in a practical decentralized setting.

In this paper, we take a different approach from the prior works. We focus on the setting where there are a large number of channels. Such a multi-user multi-channel setting naturally arises for 4G OFDM systems [12]. Somewhat surprisingly, we can develop simple decentralized algorithms that are asymptotically optimal when the number of users and the number of channels are both very large. Specifically, we first motivate the structure of our proposed algorithm by considering another version of the problem without the deadline constraints. This problem can be solved using a standard utility-based approach [8], and the resulting solution (without deadline constraints) exhibits the following decentralized structure: the base station only needs to report some congestion signal to the users, and the user can determine the video rate (i.e., layers) by comparing the congestion signal to a set of thresholds. However, as we will show in Section III, the thresholds in the resulting solution are not chosen based on the deadline constraints, and hence it is difficult to tune this algorithm to achieve good performance when there are stringent deadline constraints. Motivated by the above observation, we then design a class of threshold-based algorithms that choose the thresholds more carefully to respect the hard deadline constraints. We show that when these thresholds satisfy certain conditions, this class of algorithms can asymptotically achieve the optimal time-average per-user reward (even with deadline constraints) as the number of users and the number of OFDM channels become large. Noting that it may still involve centralized knowledge to set thresholds that meet these conditions. We further develop a practical algorithm to adaptively adjust the thresholds based on the current state of network. Our practical algorithm possesses the desirable decentralized structure: the base station does not need to know video rates and utility functions of the users, and each user only need a small amount of feedback from the base station to dynamically adjust the thresholds. Hence, our algorithm can be easily implemented in practice. Finally, our simulation results confirm that the proposed algorithm also achieves superior performance in medium-size systems.

II. Network Model

A. Model

We study the downlink video-streaming in a single cell in a modern 4G OFDM system. There are n_u wireless users, each of which wishes to receive a live stream from a video source in the backbone wireline network. Specifically, each user requests for video packets from its corresponding video source, and then the video source transmits the generated video packets to the user via the base station. We assume that the propagation delay from the source to the base station is upper bounded by σ, and each packet will expire in d time slots after it is generated at the source. The base station maintains a queue for each user, holding video packets that are waiting to be transmitted to the user. There are n_s frequency sub-carriers in the OFDM system, which we refer to as OFDM channels. In the rest of the paper, for ease of exposition, we assume that $n_u = n_s = n$. However, our methodology can be easily generalized to the case where $n_u \neq n_s$.

Different users may request different videos, and they are from different locations in the cell. To model such differences, we classify the n users into C classes. The total number of class-c users is $n_c = n\eta_c$, where $\sum_{c=1}^{C} \eta_c = 1$. Users from the same class have the same video statistics and channel statistics. Otherwise, they will be treated as different classes. We first model the video statistics. For a typical class-c user, the corresponding video source encodes the generated video into L_c layers using SVC, and then divides each layer into a sequence of packets. Packets from different layers are assigned different rewards since they contribute different video quality levels to the user. Specifically, we assume that the reward for a layer-l packet of a class-c user is $R_{c,l}$, and $R_{c,1} \geq R_{c,2} \geq ... \geq R_{c,L_c}$. Further, we assume that the average packet-generating rate for layer-l packets is $\lambda_{c,l}$. We then model the channel statistics. Our focus is the following *on-off* model with possible temporal correlation. Each class-c user has access to all the n OFDM channels. Each channel evolves as a two-state Markov chain with state-transition probability $q^{(c)} = [q_{0,0}^{(c)}, q_{0,1}^{(c)}; q_{1,0}^{(c)}, q_{1,1}^{(c)}]$ ("0" means *off*, "1" means *on*), independent of other channels. Whenever a channel is *on* for a class-c user, it can transmit r_c packets successfully from the base station to the user. Further, we assume that $q_{0,1}^{(c)}, q_{1,1}^{(c)} > 0$, and the channel states for different users are independent.

Our goal in this paper is to maximize the time-average per-user reward from those packets successfully received before their deadlines. We assume that at the beginning of time t, the base station knows the exact user-channel connectivity matrix $S(t) = [S_{i,j}(t)]_{n \times n}$, where $S_{i,j}(t) = 1$ if the j-th sub-channel is *on* for user i, and $S_{i,j}(t) = 0$ otherwise. A scheduler at the base station will then allocate channels to users in this time slot based on $S(t)$. We assume that each user can be allocated multiple channels, while each channel can only be allocated to one user. Based on the channel allocation, let $\gamma_{c,l}(t)$ be the number of layer-l packets received by class-c users at time t that meat the deadlines. Then, the total reward $J(n, t)$ obtained from packets received before their deadlines in the interval from time slot 0 to time slot $t - 1$ can be expressed as

$$J(n, t) = \sum_{t_0=0}^{t-1} \sum_{c=1}^{C} \sum_{l=1}^{L_c} \gamma_{c,l}(t_0) R_{c,l}.$$

We define the time-average per-user reward as

$$J(n) = \lim_{t \to \infty} \frac{1}{t} \frac{J(n,t)}{n}.$$

In this paper, we would like to design some joint rate-control and scheduling policies that maximize $J(n)$.

B. Desired Decentralized Structure of the Control Policies

The control of this system resides in both the user side and the base station side. At the beginning of each time slot t, a scheduler at the base station will allocate channels to users based on the user-channel connectivity matrix $S(t)$, and transmit packets to the corresponding users. Further, the base station also generates some control signal based on its queue-length information, and sends the control signal to all users through a broadcast control channel. We assume that the control signal can be immediately delivered to the users. After receiving the control signal, each user will then use the control signal to adjust the video rate by requesting different number of layers from the corresponding video sources.

In a practical system, the users and the base-station are distributed entities. Hence, for practical implementation, it is highly desirable that the control policies also possesses a decentralized structure. Specifically, the base station need not know much user-level information (e.g., packet rewards and utility function) to generate control signals and allocate channels; each user should request for a certain number of layers purely based on the control signal, independent of the behavior of other users.

III. MOTIVATION FOR THRESHOLD-BASED POLICIES

In this section, we first consider another version of the problem that ignores the deadline constraints for all the users. Then, our problem can be easily solved using a standard utility-based approach. Further, the solution exhibits a decentralized structure as we discuss earlier. However, since this solution does not account for deadlines, we refer to it as a "heuristic utility-based approach". Indeed, we will see soon that it is difficult to tune this heuristic approach to achieve good performance with deadline constraints.

We begin with an outer bound on the capacity region of our OFDM system. Let v_c denote the average packet injection rate for a class-c user, and let $\vec{v} = [v_1, v_2, ..., v_C]$. A rate vector $\vec{v}$ is said to be feasible if and only if there exists a scheduling policy such that the long-term average allocated rate for each class-c user is at least v_c. Define a set Ω as

$$\Omega = \left\{ \vec{v} : \sum_{c=1}^{C} \eta_c \frac{v_c}{r_c} \le 1 \right\}. \tag{1}$$

It is easy to check that Ω is an outer bound on the capacity region of our OFDM system. In other words, any feasible rate vector $\vec{v}$ must be in Ω. Indeed, since v_c/r_c is a lower bound on the average number of channels that a class-c user needs in each time-slot, and there are $\eta_c n$ class-c users and n channels, the outer bound Ω then follows. Later on, we will see that this

outer bound is asymptotically tight when n is large. For more details, please refer to our technical report [14].

We then propose a heuristic utility-based policy (ignoring deadline constraints) based on the above outer bound. We first construct the utility function $U_c(\cdot)$ for a class-c user. Recall that the video requested by a class-c user has L_c layers, and the packets from the l-th layer have per-packet reward $R_{c,l}$ and average generating rate $\lambda_{c,l}$. Recall that $R_{c,l}$ is monotone decreasing with respect to l. Hence, if the rate allocated to this user is v_c, to achieve the highest reward we should first serve the layer-1 packets, following the layer-2 packets, layer-3 packets, and so on, until a total rate of v_c is fully utilized or all the layers are served. Therefore, the utility function $U_c(\cdot)$ of a class-c user is a piecewise linear function, i.e.,

$$U_c(v_c) = \begin{cases} \sum_{l=1}^{L_c} R_{c,l} \left[v_c - \sum_{i=1}^{l-1} \lambda_{c,i} \right]^+, & \text{if } v_c \le \sum_{l=1}^{L_c} \lambda_{c,l}, \\ U_c(\sum_{l=1}^{L_c} \lambda_{c,l}), & \text{if } v_c > \sum_{l=1}^{L_c} \lambda_{c,l}, \end{cases} \tag{2}$$

where $[x]^+ = \max\{x, 0\}$.

Our goal is to maximize the total utility under the constraint that the rate vector $\vec{v}$ is inside the outer bound Ω, i.e.,

$$\max_{\mathbf{v}} \sum_{c=1}^{C} n \eta_c U_c(v_c), \text{ subject to } \sum_{c=1}^{C} \eta_c \frac{v_c}{r_c} \le 1. \tag{3}$$

To solve (3), we associate a Lagrange multiplier $\mu \ge 0$ to the inequality constraint, and define the Lagrangian as:

$$\begin{aligned} L(\mathbf{v}, \mu) &= \sum_{c=1}^{C} n \eta_c U_c(v_c) - \mu \left(\sum_{c=1}^{C} n \eta_c \frac{v_c}{r_c} - n \right) \\ &= \sum_{c=1}^{C} n \eta_c \left(U_c(v_c) - \frac{\mu}{r_c} v_c \right) + n\mu \end{aligned} \tag{4}$$

To maximize $L(\mathbf{v}, \mu)$, each user can independently choose the rate $v_c(t)$ at time t, such that

$$v_c(t) = \arg\max_{v_c} U_c(v_c) - \frac{\mu}{r_c} v_c, \tag{5}$$

and the base station will update μ according to

$$\mu(t+1) = \left[\mu(t) + \alpha \left(\sum_{c=1}^{C} n \eta_c \frac{v_c(t)}{r_c} - n \right) \right]^+, \tag{6}$$

where α is a positive step size.

It is easy to see that the rate-control part (5) has a threshold structure. Indeed, based on the form of $U_c(\cdot)$ in (2), we can obtain the closed-form solution to (5), i.e.,

$$v_c(t) = \begin{cases} \sum_{l=1}^{L_c} \lambda_{c,l}, & \text{if } \mu \le R_{c,L_c} r_c, \\ \sum_{i=1}^{l} \lambda_{c,l}, & \text{if } R_{c,l+1} r_c < \mu \le R_{c,l} r_c, \\ 0, & \text{if } \mu > R_{c,1} r_c. \end{cases}$$

Thus, the user will request for layer l if and only if $\mu \le R_{c,l} r_c$.

We note that the above heuristic solution possesses the desirable decentralized structure that we discussed earlier. Specifically, each user only needs the control signal μ to adjust the video layer, and can do so independently of other users.

Further, the base-station updates the control signal μ based on (6), without the need to know the utility function of each user.

However, this heuristic utility-based solution may perform poorly under deadline constraints. Specifically, note that μ/α may be thought as an approximation to the weighted queue length at the base station, where each packet of class-c is taken as of length $1/r_c$. To achieve the optimal utility, the step size α needs to be small. Unfortunately, it implies that the queue length at the base station will be large, and the delay at the base station may violate the deadline constraint d. Hence, it is difficult to tune the parameter α of the solution to attain good performance subject to deadline constraints. In the next section, we will propose a class of threshold-based policies, which respect the stringent deadline constraints and achieves close-to-optimal reward.

IV. TP POLICIES FOR REWARD OPTIMIZATION

A. Simplification of Notations

For ease of exposition, we introduce the following concept of a super-packet, which will greatly simplify the description and analysis of the system. Recall that users of class-c can transmit r_c packets when an OFDM channel is *on*. Each super-packet can contain r_c class-c packets. Note that with this concept, users from all classes can transmit exactly one super-packet when an OFDM channel is *on*.

We define the normalized reward of a layer-l packet from class-c users as $R_{c,l}r_c$, which is also the reward of a super-packet if the super-packet only contains layer-l class-c packets. We then introduce the concept of "tiers" based on the normalized reward of all the layers from all classes. We say two layers are in the same tier, if they have the same normalized reward. Assume that the total number of different tiers is L. The layers with the largest normalized reward belong to the first tier, the layers with the second largest normalized reward belong to the second tier, and so on. For ease of exposition, we assume that the normalized rewards for all L tiers of packets are $R_1 > R_2 > ... > R_L$, respectively. Note that if there are multiple classes and multiple corresponding layers with the same value of normalized reward R_l, then a tier-l super-packet may contain a different number of real packets depending on which class they are from. However, regardless of the class, a tier-l super packet always has the same reward and can be served by one channel use. Hence, the notion of super-packets considerably simplify the notation and analysis that follow. Specifically, let $a_u^l(i)$ denote the number of tier-l super-packets[2] generated for user u at time i, and let $a_u^l(i,j) = \sum_{k=i}^{j} a_u^l(k)$ denote the total number of tier-l super-packets for user u from time i to j. The total rate of tier-l super-packets for all the users is $n\lambda_l$, where

$$\lambda_l = \sum_{c,i:R_{c,i}r_c=R_l} \eta_c \frac{\lambda_{c,i}}{r_c}.$$

[2]If the source of user u does not generate tier-l packets, we will have $a_u^l(i) = 0$.

Therefore, $E[\sum_{u=1}^{n} a_u^l(\cdot)] = n\lambda_l$. Let $A_u^l(i) = \sum_{k=1}^{l} a_u^k(i)$ be the cumulative number of generated super-packets from tier 1 to tier l for user u at time i, and $A_u^l(i,j) = \sum_{k=i}^{j} A_u^l(k)$ be the sum from time i to time j. Similarly, let $A^l(i) = \sum_{u=1}^{n_u} A_u^l(i)$ be the total number of generated super-packets from tier 1 to tier l for all users at time i, and let $A^l(i,j) = \sum_{k=i}^{j} A^l(k)$ be the sum from time i to j.

Further, we make the following assumptions in the asymptotic regime.

Assumption 1. *There exist $M < \infty$, such that the total number of generated super-packet for any time i and user u is upper bounded by M, i.e., $A_u^L(i) < M$.*

Assumption 2. *Given $l = 1, 2, ..., L$ and $\epsilon > 0$, there exists $N_\epsilon^l > 0$, and $I_l(\epsilon) > 0$ independent of n and t, such that for all $t_0, t > 0, n > N_\epsilon^l$,*

$$\mathbb{P}\left(\left|\frac{A^l(t_0, t_0+t)}{n} - t\sum_{k=1}^{l}\lambda_k\right| > \epsilon\right) < e^{-nI_l(\epsilon)}.$$

Assumption 3. *Given $l = 1, 2, ..., L$ and $\epsilon > 0$, there exists $\tilde{T}_\epsilon^l, \tilde{N}_\epsilon^l > 0$, and $\tilde{I}_l(\epsilon) > 0$ independent of n and t, such that for all $t_0, t > \tilde{T}_\epsilon^l$ and $n > \tilde{N}_\epsilon^l$,*

$$\mathbb{P}\left(\left|\frac{A^l(t_0, t_0+t)}{nt} - \sum_{k=1}^{l}\lambda_k\right| > \epsilon\right) < e^{-nt\tilde{I}_l(\epsilon)}.$$

Assumption 1 is mild. Assumption 2 and Assumption 3 are also very general. Assumption 2 essentially states that the probability that the total number of generated packets deviates from its mean decays exponentially to zero as the number of sources increases. For large t, Assumption 3 further states that the exponential decay rate increases linearly with t. In our model, Assumption 2 can be verified using Cramer's Theorem [3]. Assumption 3 has been verified for packet-generating processes that follow a two-state Markov chain [3], and the proof can also be generalized to more general Markov chains.

B. Upper Bound of $J(n)$

Recall that our goal is to maximize the time-average per-user reward $J(n)$. Therefore, we first provide an upper-bound using the newly-defined "tier" structure.

Theorem 1. *Let l^* be the tier satisfying $\sum_{i=1}^{l^*} \lambda_i < 1 \leq \sum_{i=1}^{l^*+1} \lambda_i$. Then, the upper bound of $J(n)$ under any policies is given by*

$$\bar{J} \triangleq \sum_{i=1}^{l^*} \lambda_l R_l + \left(1 - \sum_{i=1}^{l^*} \lambda_l\right) R_{l^*+1}.$$

Essentially, the upper bound $\bar{J}$ is obtained by assuming that each OFDM channel can always serve one super-packet at each time slot. Specifically, we consider a large time period $[0, \tau-1]$, in which at most $n\tau$ super-packets can be served. We then use $n\tau$ to serve the real packets in a decreasing order of their normalized reward. Based on the definition of l^*, all tiers from 1 to l^* can be served, while only part of packets from tier $l^* + 1$ can be served. Summing the total reward obtained

from all of these tiers, we eventually obtain $\bar{J}$. The detailed proof is provided in our technical report [14].

We would like to point out that $\bar{J}$ is also closely related to the optimization problem defined in (3). Recall that (3) maximizes the total utility (reward) among all users under the constraint that the rate vector $\vec{v} \in \Omega$, and the outer bound Ω is essentially obtained using the same assumption that each OFDM channel can always serve one super-packet at each time slot. Therefore, the optimal solution of (3) must be equal to $n\bar{J}$. Here, with the help of the "tier" concept, we can further give a closed form of $\bar{J}$.

C. TP Policies and Optimality

We now define a policy class TP (Threshold-based Policies). This class of TP policies is motivated by the heuristic utility-based policy proposed in Section III. Specifically, in the heuristic policy, each user adjusts the layers by comparing μ to a set of thresholds $R_{c,l}r_c$. However, these thresholds are not chosen according to the deadline constraints, and hence it is difficult to tune the heuristic policy to achieve good performance subject to deadline constraints. We will see soon that most of these thresholds are non-critical. We focus on adjusting the more critical thresholds. Thus, we can obtain near-optimal reward even with stringent deadline constraints.

TP contains all the joint rate-control and scheduling policies satisfying the following requirements. A typical policy in TP consists of two parts.

1) The first part is a threshold-based rate-control policy. Let $Q(t)$ be the total number of super-packets at the base station at the beginning of each time slot t. The base station will provide $Q(t)$ as a feedback to all the users at the beginning of each time slot. Assume that there are L thresholds $1 \leq D_L \leq D_{L-1} \leq ... \leq D_1 \leq d$. (How these thresholds should be set will be clear shortly.) Based on $Q(t)$, the users will request for packets from the corresponding sources as follows: The packets from tier l will be requested if and only if $D_l n > Q(t)$. (Note that if this condition is met, all layers below tier-l are also requested.) If $Q(t) \geq D_1 n$, the users will stop requesting any packets.

2) The second part is a scheduling policy at the base station. At the beginning of each time slot t, we first drop all the expired packets. Then, based on different scenarios, we use different strategies to serve packets:

   - If there exists a channel allocation scheme that could serve all the n super-packets with the largest waiting times[3] (if the number of packets at the base station is less than n, we just consider all the packets), we will serve these n super-packets. In this case, we say there is a *successful schedule* at time slot t.
   - If the above scenario does not occur, we can use an arbitrary policy to serve packets.

---

[3]The waiting time of a super-packet is the largest waiting time of packets from this super-packet.

By comparing the heuristic utility-based approach in Section III and the TP policy, we can see some striking similarities. Specifically, the value of $\mu(t)$ corresponds to $\alpha Q(t)$. Hence, if we choose the thresholds D_l to be $\frac{R_l}{\alpha n}$, then the TP policy would have been very similar to the heuristic policy. However, as we discussed earlier, in order to achieve optimal utility, we would like α to be small. Then, we will have difficulty to ensure that all thresholds D_l are between $[1, d]$ (as stated in the TP policy)! Fortunately, our main result below shows that not all thresholds are that critical. The most important thresholds are those around the tier l^*+1 that leads to the upper bound $\bar{J}$. Next, we will show that as long as certain conditions are met for these thresholds, any policy in TP asymptotically achieves the upper-bound $\bar{J}$ of the time-average per-user reward, which is formally stated in Theorem 2.

Theorem 2. *Suppose that the thresholds satisfy the following four constraints: (1). $D_{l^*+2} \geq 1$; (2). $d - D_1 - \sigma - 2\sigma M + 1 > 0$; (3). $D_{l^*} - D_{l^*+1} - 2\sigma M > 0$; (4). $D_{l^*+1} - D_{l^*+2} - 2\sigma M > 0$. Then, the long-term average per-user reward $J(n)$ under any policy in TP asymptotically approaches the upper bound $\bar{J}$ with a positive decay-rate, i.e.,*

$$\liminf_{n \to \infty} -\frac{1}{n} \log \frac{\bar{J} - J(n)}{\bar{J}} > 0.$$

In Theorem 2, there are 4 constraints to ensure that a typical policy in TP is asymptotically optimal. From these constraints, we can see that there are only 4 critical thresholds (D_1, D_{l^*}, D_{l^*+1} and D_{l^*+2}), and three critical gaps ($d - D_1$, $D_{l^*} - D_{l^*+1}$ and $D_{l^*+1} - D_{l^*+2}$). Therefore, to implement a TP policy, we only need to worry about these three gaps. One simply strategy is to set $D_l = 1$ for all $l \geq l^* + 2$, $D_l = \frac{2(d-\sigma)}{3}$ for all $l \leq l^*$, and $D_{l^*+1} = \frac{D_{l^*}+D_{l^*+2}}{2}$. Compare to the heuristic policy, as we discussed above the gap between the corresponding thresholds D_{l^*} and D_{l^*+1} would have been $\frac{R_{l^*}-R_{l^*+1}}{\alpha n}$. Therefore, it would have been difficult to keep all the thresholds between $[1, d]$ when the step size α is small. In theorem 2, by focusing only on D_{l^*} and D_{l^*+1}, effectively a smaller value of α is now used, while at the same time all thresholds are between $[1, d]$. This intuition is the main reason why the TP policy can attain near-optimal utility subject to deadline constraints.

We also note that the conditions in Theorem 2 require the gap between the critical thresholds to be larger when the propagation delay σ is large. Intuitively, a larger σ implies that the closed-loop control is more sluggish. Thus, we would expect that it would be more difficult to attain the optimal reward. Indeed, if σ is too large, it may not be possible to satisfy the conditions in Theorem 2 given the deadline constraints d. Hence, these conditions are of critical importance in practice.

D. Proof of Theorem 2

1) A Key Preliminary Lemma: Recall that the upper bound $\bar{J}$ is derived based on the assumption that each OFDM channel can transmit one super-packet at each time slot. The following lemma suggests that when the number of users and channels

are large, our TP policy can indeed serve n super-packets with probability close to 1 at each time slot.

Lemma 3. *Let $\omega_i(t), i = 1, 2, ..., n$ be n non-negative integers satisfying $\max_{1 \le i \le n} \omega_i(t) \le H$ and $\sum_{i=1}^n \omega_i(t) \le n$, where H is a fixed positive constants. Let $\mathcal{A}(t)$ be the event that there exists a channel allocation scheme at time t, such that at least $\omega_i(t)$ OFDM channels are allocated to user i. Then, there exists a polynomial function $f(n)$ of n and a constant N_0, such that for all $n > N_0$, the following holds*

$$\mathbb{P}(\mathcal{A}(t)|S(t-1)) \ge 1 - f(n)e^{-nI},$$

where $I = \frac{1}{H} \log \frac{1}{1-q} > 0$ and $q = \min_{c=1,...,C} \{q_{0,1}^{(c)}, q_{1,1}^{(c)}\} > 0$.

In Lemma 3, $\mathcal{A}(t)$ is an event that is mainly determined by the channel state $S(t)$ at time t. Consider the j-th OFDM channel, even though the probability that it is *on* for one user is less than 1, when there are many users we can find a user i such that $S_{i,j}(t) = 1$ with high probability. In other words, there exists significant diversity in this system. Lemma 3 makes this intuition rigorous by showing that we can in fact find a channel allocation to send any n super-packets for all users with high probability, regardless of the past channel state. We note that Lemma 3 is similar to Lemma 6 in [3]. The key difference, however, is that we only require $\sum_{i=1}^n \omega_i \le n$ instead of $\sum_{i=1}^n \omega_i \le n - H$, at the cost of a smaller decay rate I, and we allow temporally correlated channels. The detailed proof is given in our technical report [14].

Recall the system model where we assume that the number of generated super-packets at each time slot for each user is upper bounded by M. Further, each super-packet will expire in at most d time slots. Therefore, if we let $H = dM$, then in each time slot, each user could have no more than H super-packets at the base station. Thus, by Lemma 3, the OFDM system will be able to serve any n super-packets with high probability. In other words, any policy in TP will see a successful schedule at each time slot with probability close to 1. This insight also implies that the outer bound Ω in (1) is tight when the system size is large.

2) Sketch of the Proof of Theorem 2: To prove Theorem 2, we need to find a lower bound of $J(n)$ for an arbitrary policy π_{TP} in TP. Recall that not all layers of packets will be injected to the network. We use $\hat{J}(n)$ to denote the average reward from all the injected packets at each time slot. Further, not all injected packets can be served before expiration. We use p_d to denote the probability that a typical injected packet is dropped due to expiration. Note that the system reward is obtained from those packets meeting their deadlines. Therefore, $J(n) = \frac{1}{n}\hat{J}(n)(1 - p_d)$.

The main difficulty is to find a lower bound on $\hat{J}(n)$. Note that the injection of packets is controlled by $Q(t)$ and the set of thresholds $D_1, D_2, ..., D_L$. If we know the probability distribution of $Q(t)$, then we can compute $\hat{J}(n)$. Specifically, let $q_0 = \mathbb{P}(Q(t) < nD_{l^*+2})$, $q_1 = \mathbb{P}(nD_{l^*+2} \le Q(t) < nD_{l^*+1})$, $q_2 = \mathbb{P}(nD_{l^*+1} \le Q(t) < nD_{l^*})$ and $q_3 = \mathbb{P}(Q(t) \ge nD_{l^*})$. Obviously, $\sum_{i=0}^3 q_i = 1$. Furthermore,

if $D_{l^*+2} \ge 1$ (constraint 1 in Theorem 2), we also have

$$q_0 \sum_{l=l^*+2}^{L} \lambda_l + (q_0 + q_1)n\lambda_{l^*+1} + n\sum_{l=1}^{l^*} \lambda_l$$
$$\ge n(1 - q_0)(1 - f(n)e^{-nI}). \qquad (7)$$

Eqn. (7) is obtained from the fact that not all injected packets will get served eventually. The left hand side of Eqn. (7) is an upper bound on the average number of super-packets injected in each time slot, while the right hand side of Eqn. (7) is a lower bound on the average number of super-packets served in each time slot. More details are provided in our technical report [14].

Recall that if the event $\{Q(t) < nD_{l^*+1}\}$ occurs, all packets up to layer $l^* + 1$ will be injected into the network. If the event $\{nD_{l^*+1} \le Q(t) < nD_{l^*}\}$ occurs, all packets up to layer l^* will be injected into the network. Then, we can obtain a lower bound for $\hat{J}(n)$ based on q_0, q_1, q_2 and q_3, i.e.,

$$\begin{aligned}
\hat{J}(n) &\ge (q_0 + q_1)n\sum_{l=1}^{l^*+1} \lambda_l R_l + q_2 n\sum_{l=1}^{l^*} \lambda_l R_l + q_3 \cdot 0 \\
&= (1 - q_3)n\sum_{l=1}^{l^*} \lambda_l R_l + (q_0 + q_1)n\lambda_{l^*+1}R_{l^*+1} \\
&\ge (1 - q_3)n\sum_{l=1}^{l^*} \lambda_l R_l + \left(n(1 - q_0)(1 - f(n)e^{-nI})\right. \\
&\qquad \left. - q_0 n\sum_{l=l^*+2}^{L} \lambda_l - n\sum_{l=1}^{l^*} \lambda_l\right)R_{l^*+1}. \qquad (8)
\end{aligned}$$

The last inequality of Eqn. (8) is obtained using Eqn. (7). Recall that $J(n) = \frac{1}{n}\hat{J}(n)(1 - p_d)$. Comparing (8) with $\bar{J}$ in Theorem 1, we can see that if q_0, q_3 and p_d converge exponentially fast to 0 as $n \to \infty$, then the result of Theorem 2 follows. (More detailed steps are available in our technical report [14].) Such an exponential decay for q_0, q_3 and p_d are shown in the following lemma.

Lemma 4. *There exists N such that for all $n \ge N$,*

$$p_d \le f_1(n)e^{-nI_1}, \quad q_3 \le f_2(n)e^{-nI_2}, \quad q_0 \le f_3(n)e^{-nI_3},$$

where $I_1 = (d - D_1 - \sigma - 2\sigma M + 1)I$, $I_2 = \min_{c \ge 0}\{cI + I_{l^*}(D_{l^*} - D_{l^*+1} - 2\sigma M - c)\}$, $I_3 = \min\{I_1, I_{l^*+1}(D_{l^*+1} - D_{l^*+2} - 2\sigma M)\}$, and $f_1(n), f_2(n), f_3(n)$ are polynomials of n. Note that $I_1, I_2, I_3 > 0$ if constraints 2,3,4 in Theorem 2 are satisfied.

The intuition behind Lemma 4 is as follows. Note that a TP policy provides a closed-loop control for the entire system. According to Lemma 3, we know that the total service rate in the unit of super-packets is very close to n. If the number of backlogged super-packets $Q(t)$ is greater than $D_{l^*+1}n$, then the total injecting rate is $n\sum_{l=1}^{l^*} \lambda_l < n$. Therefore, the backlog has a tendency to decrease. Similarly, if $Q(t)$ is smaller than $D_{l^*+1}n$, the backlog will have a tendency to increase. As a result, $Q(t)$ will stay around $D_{l^*+1}n$. Thus,

the probability q_0 of underflow ($Q(t) < nD_{l^*+2}$) and the probability q_3 of overflow ($Q(t) \geq nD_{l^*}$) should both be very small. As for p_d, if a packet is dropped eventually, then at the time that this packet arrives at the base-station, the queue size Q needs to be very large, which deviates far from the equilibrium point $D_{l^*+1}n$. As we explain earlier, such an event occurs with very low probability too.

We note that the results in Lemma 4 share some similarity to the many-source many-channel asymptotes in [3]. However, the analysis in [3] is for an open-loop system, where the statistics of the packet-injection process are given. In contrast, Lemma 4 is for a closed-loop system where the packet-injection is also controlled by the queue length. Hence, the statistics of the packet-injection process is unknown beforehand. Due to this reason, the proof is also significantly different from that in [3]. For the detailed proof, please refer to our technical report [14].

V. AN ADAPTIVE IMPLEMENTATION OF THE TP POLICIES

We have shown that the TP policies achieve close-to-optimal time-average per-user reward under certain conditions on the thresholds. However, it remains a challenging problem to find the thresholds to satisfy these conditions. First, it requires the base station to know l^*, which depends on the normalized rewards for all the layers from all classes of users. Second, such a value l^* may change when a user moves to another location (in which case the channel rate will change) or when users join or leave the system dynamically. Therefore, there is a pressing need to develop decentralized and adaptive ways to set these thresholds.

In this section, we will develop such decentralized and adaptive implementation of the TP policies, which we call ATP (Adaptive Threshold-based Policy). The main idea of ATP is as follows. Suppose that the normalized rewards R_l of all tiers l are laid down on a direct line, shown as the top line in Fig. 1. Further, recall that the queue length $Q(t)$ at the base station in the unit of super-packets lied between $[0, nd]$, which is the bottom line in Fig. 1. Suppose that now the base station focuses on the interval $[R(1-\beta), R(1+\beta)]$ on the top line, and use a linear mapping $D(\cdot)$ that maps the interval to the interval $[0, d]$. If R_{l^*}, R_{l^*+1} and R_{l^*+2} all belong to the interval $[R(1-\beta), R(1+\beta)]$, then their values of $D(R_{l^*}), D(R_{l^*+1})$ and $D(R_{l^*+2})$ will correspond to the thresholds D_{l^*}, D_{l^*+1} and D_{l^*+2} of the TP policy. Like the heuristic utility-based approach in Section III, each user can simply compare $nD(R_{c,l}r_c)$ with $Q(t)$, which precisely implements these thresholds for TP. Further, if β is sufficiently small, then the gaps between these thresholds will be large, which may then satisfy the conditions in Theorem 2. A decentralized implementation of the TP policy thus results. However, the difficulty lies in that the base station does not know the right value for R and β before hand. Hence, we propose to adaptively adjust these values based on the observation of the queue length.

Specifically, the ATP policy contains two components: Algorithm 1 for base-station-side control and Algorithm 2 for

---

(1) Initialization: set constants $b_{\text{right}} = \frac{2(d-\sigma)}{3}, b_{\text{middle}} = \frac{1+b_{\text{right}}}{2}$; set control parameters $R = R^{(0)}, \beta = 1, \vec{Q}_{\text{history}}(1:4\sigma) = 0, \bar{Q}_0 = 0, p = 0$.

(2) Updating control parameters: obtain the total number of super-packets $Q(t)$ at the beginning of time slot t. Let

$$\bar{Q}_t = 0.9\bar{Q}_{t-1} + 0.1Q(t)$$

be the exponentially weight average queue length, and set

$$\vec{Q}_{\text{history}} = [\vec{Q}_{\text{history}}(2:4\sigma), \bar{Q}_t].$$

Update p according to

$$p = 0.9p + 0.1\mathbb{1}_{\{n\frac{4+b_{\text{middle}}}{5} \leq Q(t) \leq n\frac{b_{\text{middle}}+4b_{\text{right}}}{5}\}},$$

where $\mathbb{1}_{\{\cdot\}}$ is an indicator function. Use linear regression on $\vec{Q}_{\text{history}}$ to estimate the slope s of the smoothed queue length. Then update R and β as follows:

if $\bar{Q}_t < n\frac{1+b_{middle}}{2}$ **then**
 | update $R = R/\rho_1$ if $s < 0$;
else if $\bar{Q}_t > n\frac{b_{right}+b_{middle}}{2}$ **then**
 | update $R = R \times \rho_1$ if $s > 0$;
else
 | **if** $s > 0.1$ **then** $R = R + \rho_2$;
 | **else if** $s < -0.1$ **then** $R = R - \rho_2$;
 | **else**
 | | **if** $p > 0.99$ **then** $\beta = \max\{\beta - \rho_3, 0.1\}$;
 | | **if** $p < 0.9$ **then** $\beta = \min\{\beta \times \rho_4, 1\}$;
 | **end**
end

Send R, β and $\frac{Q(t)}{n}$ to all the users through a control channel.

(3) Scheduling: obtain the channel states by some channel estimation strategy. Use the second part of a TP policy to allocate channels and serve packets.

Algorithm 1: ATP policy-base station side.

---

(1) Initialization: set constants $b_{\text{right}} = \frac{2(d-\sigma)}{3}, b_{\text{middle}} = \frac{1+b_{\text{right}}}{2}$. Estimate its maximum transmission rate r_c in one sub-channel.

(2) Constructing threshold mapping function $D(\cdot)$: obtain R, β and $\frac{Q(t)}{n}$ from the base station. Use R and β to construct the function $D(\cdot)$ as follows:

$$D(x) = \begin{cases} 1, & \text{if } x < R(1-\beta), \\ b_{\text{right}}, & \text{if } x > R(1+\beta), \\ (x-R)\frac{b_{\text{right}}-1}{2\beta R} + b_{\text{middle}}, & \text{else.} \end{cases} \quad (9)$$

(3) Request all video layers satisfying $D(R_{c,l}r_c) > \frac{Q(t)}{n}$ from the corresponding video source.

Algorithm 2: ATP policy-user side.

user-side control. The user-side control is simpler. Each user will generate the same piecewise-linear mapping function $D(\cdot)$ (see Eqn. (9)) based on the control parameters R and β given by the base station. By comparing $D(R_{c,l}r_c)$ with $Q(t)/n$, each user can decide the layers that it wishes to request.

The base-station-side control is more complex[4]. It needs to adaptively estimate the parameters R and β based purely on its queue evolution statistics. Our goal is to set R close to R_{l^*+1}, i.e., the normalized reward of tier-$(l^* + 1)$, and to set β to be sufficiently small, so that all the thresholds mapped by $D(\cdot)$ are set as desired by Theorem 2, i.e., $D_l = 1$ for all $l \geq l^*+2$, $D_l = \frac{2(d-\sigma)}{3}$ for all $l \leq l^*$, and $D_{l^*+1} = \frac{D_{l^*}+D_{l^*+2}}{2}$. Note that if R is estimated correctly, $Q(t)$ will be stabilized around $nb_{\text{middle}} = n\frac{1+b_{\text{right}}}{2}$. To adaptively find the right R, we let the base station keep track of the exponentially weighted average (EWA) queue $\vec{Q}_{\text{history}}$. We then compute the slope s at which the EWA queue $\vec{Q}_{\text{history}}$ increase or decrease. If the current EWA queue $\bar{Q}_t$ is too small compared to nb_{middle}, we need s to be positive. Hence, if $s < 0$, we will decrease R so that more tiers of packets can be injected to the network, and thus make $s > 0$. Similarly, if $\bar{Q}_t$ is too large, we need s to be negative. Hence, if $s > 0$, we will increase R so that less tiers of packets can be injected to the network, and thus make $s < 0$. if $\bar{Q}_t$ is close to nb_{middle}, we need s to be approximately 0. Then, if s is too small, we will decrease R to increase s; if s is too large, we will increase R to decrease s. After correctly estimating R, we should estimate β. If β is set to be too large, then the three critical gaps in Theorem 2 would become smaller, and hence the queue $Q(t)$ will be constrained in a small interval. If β is set to be too small, then $Q(t)$ may oscillate over a large interval. Thus, to adaptively set the value of β, we construct a window $[n\frac{4+b_{\text{middle}}}{5}, n\frac{b_{\text{middle}}+4b_{\text{right}}}{5}]$ around nb_{middle}. If the probability p that $Q(t)$ is in this window is too large, we will decrease β; if p is too small, we will increase β. Based on the above control, the values of R and β will eventually approach to the desired values. Next, we will study the performance of our ATP policy via simulation.
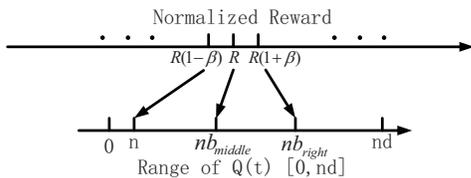

Fig. 1. Illustration of the mapping from normalized rewards to thresholds.

VI. SIMULATION

In this section, we are interested in the following policy in TP, which is called TP-DWM (Delay Weighted Matching) policy. In TP-DWM, each super-packet is assigned a weight, which is equal to the largest waiting time of the packets inside this super-packet. Then, TP-DWM tries to assign channels to super-packets such that the total weight is maximized.

[4]$R_0, \rho_1, \rho_2, \rho_3, \rho_4$ are a set of configuration parameters.

TP-DWM belongs to TP because TP-DWM will definitely serve super-packets with the largest waiting times if it sees a successful schedule at time t [3]. In the following, we will study the performance of the ATP-DWM, which is the adaptive implementation of TP-DWM as discussed in Section V.
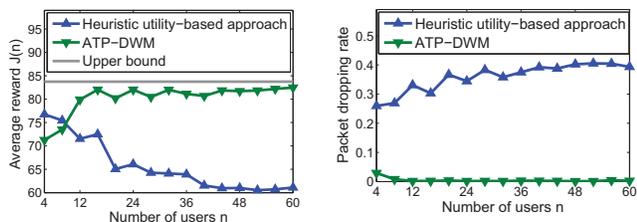
A. ATP-DWM v.s. the Utility-based Approach

We compare the time-average per-user reward and the packet dropping rate between the ATP-DWM policy and the heuristic utility-based approach. We use a Markov chain to generate videos. Specifically, we assume that each video has busy periods and idle periods, and the state of each period (busy or idle) evolves as a Markov chain with state transition probability $[0.8, 0.2; 0.4, 0.6]$. For each layer of packets, we assume that the number of packets generated in each period follows a Poisson distribution, and the packet-generating rate in busy periods is twice as much as that in idle periods. All videos are generated using the same approach described above. However, different videos can have different number of layers, different rewards and different packet-generating rates. We also use a Markov chain to generate OFDM channels. We model the channel for each channel-user pair by a two-state Markov chain with state-transition probability $q = [q_{0,0}, q_{0,1}; q_{1,0}, q_{1,1}]$. However, the state-transition probability q and the transmission rate r when the channel is *on* can be different for different users.

More precisely, we assume that there are two classes of users. The videos for the first class of users have 4 layers, with rewards $\{100, 50, 30, 20\}$ (from layer 1 to layer 4) and packet generating rates $\{0.7, 0.5, 0.5, 0.4\}$ (again from layer 1 to layer 4). The OFDM channels for the first class of users have the state-transition probability $q^{(1)} = [0.6, 0.4; 0.3, 0.7]$, and the transmission rate $r_1 = 1$. The videos for the second class of users have 3 layers, with rewards $\{60, 40, 10\}$ (from layer 1 to layer 3) and packet generating rates $\{0.8, 0.5, 0.3\}$ (again from layer 1 to layer 3). The OFDM channels for the second class of users have the state-transition probability $q^{(2)} = [0.5, 0.5; 0.2, 0.8]$, and the transmission rate $r_2 = 2$. There are n users, and the total numbers of class 1 and class 2 users are $n_1 = n_2 = \frac{n}{2}$. We assume that the propagation delay between the video source and the base station is 2, and all the packets will expire in $d = 13$ time slots after it is generated at the video source. Based on Theorem 1, it is easy to compute that the upper bound of the time-average per-user reward is 83.75. We compute the time-average per-user reward $J(n)$ through simulation under both the ATP-DWM polciy (the parameters of ATP-DWM are set as $R_0 = 10, \rho_1 = 1.01, \rho_2 = 1, \rho_3 = 0.01, \rho_4 = 2$) and the utility-based approach (the step size is $\alpha = \frac{1}{n}$). We vary n from 4 to 60. From Fig. 2, we can see that the ATP-DWM policy outperforms the utility-based approach in terms of both the average reward and the packet dropping rate.

B. Impact of User Dynamics on ATP-DWM

We study how the ATP-DWM policy adapt to user dynamics, e.g., user-arrival, user-departure, and user-moving. We

(a) Average reward under different policies vs. the upper bound $\bar{J}$.

(b) Packet drop rate under different policies.

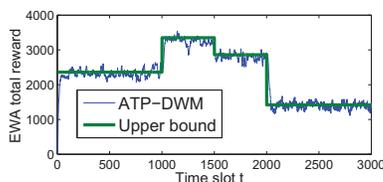Fig. 2.   PTP-DWM v.s. the Utility-based Approach.

Fig. 3.   Performance of ATP-DWM under user dynamics.

simulate 20 class-1 users and 20 class-2 users for 3000 time slots in an OFDM system with 40 channels. The video and channel statistics of each class of users are the same as that in Section VI-A. All the class-1 users are present in the system at time 0, and will leave the system at time 2000. All the class-2 users will join the system at time 1000, and leave the system at time 3000. All class-2 users will move at time 1500. As a result, their channel rate r_2 will change from 2 to 1. We obtain the total reward of all users[5] at each time t via simulation, and keep track of the exponentially weighted average (EWA) of the total reward as a function of time using

EWA total reward(t) (10)

$= 0.9 \times$ EWA total reward$(t-1) + 0.1 \times$ total reward(t).

In Fig. 3, we compare the EWA total reward with the corresponding upper bound (obtained by multiplying the upper bound $\bar{J}$ in Theorem 1 by the total number of on-going users). We can see that the ATP-DWM policy can quickly adapt to the right decisions to achieve a close-to-optimal reward[6]. Thus, the ATP-DWM policy is adaptive to user dynamics.

VII. CONCLUSION

We study the joint rate-control and scheduling problem with hard deadline constraints in a modern 4G OFDM system. Motivated by a heuristic utility-based algorithm, we design a class of threshold-based algorithms that can asymptotically achieve the optimal time-average per-user reward as the number of users and the number of OFDM channels become large,

provided that the thresholds satisfy certain conditions. Noting that it may still involve centralized knowledge to set thresholds that meet these conditions. We further develop a practical algorithm, called ATP, to adaptively adjust the thresholds based on the current state of network. Our practical algorithm possess the desirable decentralized structure: the base station does not need to know video rates, packets rewards, and utility functions of the users, and each user only need a small amount of feedback $(R, \beta, Q(t)/n)$ from the base station to dynamically adjust the thresholds. Finally, our simulation results show that the ATP algorithm not only outperforms the heuristic algorithm, but also adapt quickly to system dynamics.

REFERENCES

[1] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2011-2016," *Cisco*, Feb. 14, 2012. [online]. Available: http://www. cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html

[2] H. Sun, A. Vetro, J. Xin, "An overview of scalable video streaming," *Wireless Communications and Mobile Computing*, vol. 7, no. 2, 2007.

[3] M. Sharma and X. Lin, "OFDM Downlink Scheduling for Delay-Optimality: Many-Channel Many-Source Asymptotics with General Arrival Processes," in *2011 Information Theory and Applications Workshop*, UC San Diego, CA, February 2011.

[4] V. J. Venkataramanan and X. Lin, "Low-Complexity Scheduling Algorithm for Sum-Queue Minimization in Wireless Convergecast," in *IEEE INFOCOM*, Shanghai, China, April 2011.

[5] S. Zhao and X. Lin, "On the Design of Scheduling Algorithms for End-to-End Backlog Minimization in Multi-hop Wireless Networks," in *IEEE INFOCOM*, Orlando, Florida, USA, March 2012.

[6] X. Li, C.-C. Wang, and X. Lin, "Throughput and Delay Analysis on Un-coded and Coded Wireless Broadcast with Hard Deadline Constraints," in *IEEE INFOCOM Mini-Conference*, San Diego, CA, March 2010.

[7] I-Hong Hou and P.R. Kumar, "Queueing Systems with Hard Delay Constraints: A Framework for Real-Time Communication over Unreliable Wireless Channels," in *Queueing Systems: Theory and Applications (QUESTA)*, vol. 71, no. 1-2, pp. 151 Ù 177, Jun. 2012.

[8] X. Lin and N. B. Shroff, "Joint Rate Control and Scheduling in Multihop Wireless Networks," in *43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas, December 2004.

[9] G. J. Sullivan and T. Wiegand, "Rate-Distortion Optimization for Video Compression", in *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 7490, Nov. 1998.

[10] C. E. Luna, L. P. Kondi, and A. K. Katsaggelos, "Maximizing user utility in video streaming applications," in *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 2, pp. 148, Feb. 2003.

[11] T. Stockhammer, "Dynamic adaptive streaming over HTTP–standards and design principles," in *MMSys'11*, pp. 133-144, 2011.

[12] J. Zyren and W. McCoy, "Overview of the 3GPP long term evolution physical layer," Freescale Semiconductor, Inc., white paper, 2007.

[13] C. Chen, R. W. Heath Jr., A. C. Bovik and G. de Veciana, "Adaptive Policies for Real-Time Video Transmission: A Markov Decision Process Framework." in *IEEE international Conference on Image Processing*, September 2011.

[14] S. Zhao and X. Lin, "Joint Layer Control and Scheduling for Wireless Live Streaming with Stringent Deadline Constraints," *Technical Report*, Purdue University, 2013. [Online]. Available: http://web.ics.purdue.edu/%7ezhao147/.

[15] X. Ji, J. Huang, M. Chiang, G. Lafruit and F. Catthoor, "Scheduling and Resource Allocation for SVC Streaming Over OFDM Downlink Systems," in *IEEE Trans. Circuits Syst. Video Techn*, vol. 19, no. 10, pp. 1549-1555, 2009.

[16] S. H. Kang and A. Zakhor, "Packet Scheduling Algorithm for Wireless Video Streaming," in *Packet Video 2002*, Pittsburgh, April 2002.

[5]Here we do not use the time-average per-user reward, because the total number of users is changing.

[6]It is possible that the rewards obtained at some time slots are higher than the upper bound. Therefore, it is also possible for the EWA total reward obtained using Eqn. (10) to be greater than the upper bound. In fact, if we average the total reward over a long time period, the time-average total reward would be smaller than the upper bound.