Decomposition Principles and Online Learning in Cross-Layer Optimization for Delay-Sensitive Applications

Fangwen Fu, *Student Member, IEEE*, and Mihaela van der Schaar, *Fellow, IEEE*

***Abstract*—In this paper, we propose a general cross-layer optimization framework for delay-sensitive applications over single wireless links in which we explicitly consider both the heterogeneous and dynamically changing characteristics (e.g., delay deadlines, dependencies, distortion impacts, etc.) of delay-sensitive applications and the underlying time-varying channel conditions. We first formulate this problem as a nonlinear constrained optimization by assuming complete knowledge of the application characteristics and the underlying channel conditions. This constrained cross-layer optimization is then decomposed into several subproblems, each corresponding to the cross-layer optimization for one DU. The proposed decomposition method explicitly considers how the cross-layer strategies selected for one DU will impact its neighboring DUs as well as the DUs that depend on it through the resource price (associated with the resource constraint) and neighboring impact factors (associated with the scheduling constraints). However, the attributes (e.g., distortion impact, delay deadline, etc.) of future DUs as well as the channel conditions are often unknown in the considered real-time applications. In this case, the cross-layer optimization is formulated as a constrained Markov decision process (MDP) in which the impact of current cross-layer actions on the future DUs can be characterized by a *state-value function*. We then develop a low-complexity cross-layer optimization algorithm using online learning for each DU transmission. This online optimization utilizes information only about the previous transmitted DUs and past experienced channel conditions, which can be easily implemented in real-time in order to cope with unknown source characteristics, channel dynamics and resource constraints. Our numerical results demonstrate the efficiency of the proposed online algorithm.**

***Index Terms*—Cross-layer optimization, delay-sensitive applications, online learning, online optimization, wireless multimedia transmission.**

I. Introduction

ONE of the key challenges associated with the robust and efficient transmission of delay-sensitive data (e.g., video conferencing and real-time video streaming) over wireless networks is the *dynamic* characteristics of both the wireless networks and delay-sensitive applications experienced by a wireless user (i.e., a pair of transmitter and receiver) [1]. To overcome this challenge, the wireless user needs to jointly optimize the various protocol parameters and algorithms available at each layer of the OSI stack in order to maximize its application's utility (e.g., video quality). This joint optimization of the transmission strategies at the various layers is referred to as *cross-layer optimization* [1], [2]. In this paper, we focus on the single-user cross-layer optimization for delay-sensitive data transmission over a single-hop wireless network (i.e., a single wireless link).

Manuscript received October 31, 2008; accepted September 17, 2009. First published October 20, 2009; current version published February 10, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Christine Guillemot.

The authors are with the Electrical Engineering Department, University of California Los Angeles (UCLA), Los Angeles, CA 90095 USA (e-mail: fwfu@ee.ucla.edu; mihaela@ee.ucla.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2009.2034938

A. Related Research

Cross-layer optimization has been extensively investigated in recent years in order to maximize the application's utility given the underlying time-varying and error-prone channel characteristics. The majority of cross-layer optimization solutions [3]–[15] for single-link communications model the time-varying network conditions (e.g., channel conditions at the physical layer, allocated time/frequency bands at the MAC layer, etc.) and/or application characteristics (e.g., packet arrivals, delay deadlines, distortion impact, etc.) as (controlled) stochastic processes and aim to sequentially determine the cross-layer actions over time to control this stochastic process such that the long-term utility is maximized. The most important advantage of such sequential approaches is that they allow the wireless user to consider the experienced source and network dynamics (which are affected by both the uncertainty in the environment and the actions chosen by the wireless user) and, based on the user' knowledge about these dynamics up to that moment, select its cross-layer transmission strategies to maximize their utility *over time*.

Current cross-layer solutions often involve only the layers below the application layer, which collectively aim to maximize QoS metrics such as throughput, packet loss rate, average or worst case delay etc., but without considering the specific characteristics and requirements of the applications. For example, in [3] and [5], the cross-layer optimization is performed in order to minimize the incurred average delay for applications under energy (or average power) constraints. In [4], the cross-layer optimization is performed with the aim of increasing the spectrum efficiency under the average delay and packet loss rate constraints. In both cases, the application packets are assumed to be homogeneous (i.e., having the same distortion impact and same delay deadlines). The hard delay deadlines of the packets (i.e., the time after which packets expire and thus becomes useless if received) are then considered in [6]–[11], where the optimal packet scheduling algorithm is developed for the transmission of a group of equal-importance packets, which minimizes the consumed energy while satisfying their delay deadlines. However, the above papers disregard key properties of delay-sensitive applications: the interdependencies among packets and their different distortion impacts.

To take into consideration the heterogeneous characteristics of the delay-sensitive data, the packet scheduling is often performed in order to maximize the application utility at the application (APP) layer. In [14], the video packets with various characteristics are scheduled considering a common delay deadline and an optimal solution (including optimal packet ordering and retransmission) is developed assuming that the underlying wireless channel is static. In [12], the delay-constrained data are scheduled over a constant wireless channel in order to minimize the remaining distortion of the applications (accordingly, maximizing the application utility). In [13], the optimal packet scheduling (corresponding to the rate allocation there) is developed for the embedded data transmission over noisy channels with constant packet loss rates. In [15], a directed acyclic graph (DAG) model is used to capture the media packet dependencies and, based on this, an optimal packet scheduling method is developed using dynamic programming [17]. However, the proposed solutions disregard the dynamics and error protection capabilities at the lower layers (e.g., MAC and physical layers).

Summarizing, a general cross-layer optimization framework which simultaneously considers both the heterogeneous and dynamically changing characteristics of delay-sensitive applications and the underlying time-varying network conditions is still missing. In this paper, we aim to develop a solution that addresses both of these challenges for the delay-sensitive applications such as multimedia transmission. In the developed cross-layer optimization framework, packet scheduling and transmission strategy adaptation will be jointly optimized in order to maximize the application utility. The packet scheduling is often performed in the APP layer to consider the heterogeneous characteristics of the delay-sensitive data. The transmission strategy is referred to the transmission parameter adaptation in the other layers beside the APP layer in order to adapt to the time-varying channel conditions. The transmission strategy can include, e.g., the average retransmission at the MAC layer [14], power allocation in the physical (PHY) layer.

B. Contribution of This Paper

Delay-sensitive multimedia data (e.g., video) is often encoded using prediction-based coding schemes which may introduce sophisticated dependencies among the data [25], [26] and then packetized into multiple data units (DU) for transmission. Each DU can be further divided into one or multiple packets when it is scheduled for transmission. We assume that the cross-layer decisions are performed for each DU. We consider both independently decodable DUs (i.e., they can be decoded independently without requiring the knowledge of other DUs) as well as interdependent DUs (i.e., in order to be decoded, each DU requires those DUs it depends on to be decoded beforehand and these dependencies are expressed as a DAG). We first formulate a nonlinear constrained optimization problem by assuming complete knowledge of the attributes[1] (including the time ready for transmission, delay deadlines, DU size and distortion impact, and DAG-based dependencies) of the application DUs and the underlying channel conditions. The formulations in [8]–[10], [14] are special cases of the framework proposed in this paper.

Interestingly, the formulated nonlinear constrained cross-layer optimization can be decomposed into several subproblems and two master problems. One master problem corresponds to the Lagrange multiplier (i.e., price of the resource) update associated with the considered resource constraint imposed at the lower layer (e.g., energy constraint); and the other master problem corresponds to the update of the Lagrange multipliers [called neighboring impact factors (NIFs)] associated with the DU scheduling constraints between neighboring DUs.[2] Each subproblem represents the cross-layer optimization for one DU given the resource price and NIFs of its neighboring DUs. As we will show in this paper, the proposed decomposition illustrates how the cross-layer strategies for one DU impact its neighboring DUs and the DUs it connects with in the DAG, and finally, induces the online cross-layer optimization which is described next.

In delay-sensitive real-time applications, the wireless user is often not allowed or cannot know the attributes of future DUs and corresponding channel conditions. In other words, it only knows the attributes of previous DUs, and past experienced network conditions and transmission results. The message exchange mechanism developed based on the decomposition of the nonlinear optimization is infeasible since it requires exact information about future DUs. However, when the distribution of the attributes and channel conditions of DUs fulfil the Markov property [23], the cross-layer optimization can be reformulated as a constrained MDP [30]. Then, the impact of the cross-layer action of the current DU on the future unknown DUs are characterized by a state-value function which quantifies the impact of the current DU's cross-layer action on the future DUs' distortion. Using the obtained decomposition principles developed for the cross-layer optimization with complete knowledge, we develop a low-complexity algorithm which only utilizes the available (causal) information to solve the online cross-layer optimization for each DU, update the resource price and learn the state-value function.

The rest of the paper is organized as follows. Section II formulates the cross-layer optimization problem for the independently decodable DUs as a nonlinear constrained optimization assuming the knowledge of the characteristics of the supported application and underlying channel conditions, and decomposes the optimization problem and presents the necessary message exchanges between layers and between neighboring DUs. Section III further formulates the cross-layer optimization for interdependent DUs as a nonlinear constrained

[1]This is the case, for instance, when the delay-sensitive data was preencoded and hinting files were created before transmission time [24]. However, in the real-time encoding case, these attributes are known just in time when the packets are deposited in the streaming buffer, which will be considered in Section IV.

[2]These are consecutive packets generated by the source codec in the encoding/decoding order.

optimization and presents the decomposed cross-layer optimization algorithm based on the decomposition principles developed in Section II-B. Section IV presents an online cross-layer optimization for each DU transmission. Section V shows some numerical results, followed by the conclusions in Section VI.

II. Cross-Layer Optimization for Independently Decodable DUs

In this paper, we consider the problem that a wireless user streams delay-sensitive data over a time-varying single wireless link. In this section, we consider that the DUs are independently decodable and will discuss the cross-layer optimization for the interdependent DUs in Section III.

A. Formulation

Specifically, the wireless user has $M \in \mathbb{N}$ DUs with individual delay constraints and different distortion impacts. Each DU has the following attributes:

- *Size*: The size of DU $i \in \mathbb{N}$ is denoted as l_i (measured in bits).
- *Distortion impact*: DU i has a distortion impact q_i, which is the amount by which the distortion will be reduced if the DU is decoded at the destination.
- *Arrival time*: The arrival time is the time at which the DU is ready for transmission. The arrival time for DU i is denoted by t_i. If the delay-sensitive data is preencoded, then each DU is available for transmission at $t_i = 0$. If the delay-sensitive data is encoded in real time, the arrival time is the time when the DU is packetized and injected into the postencoding buffer.
- *Delay deadline*: The delay deadline is the time by which the data unit must be decoded. If the DU is not received at the destination by the delay deadline, it will be discarded and it will be considered useless.[3] The delay deadline is denoted by d_i and $t_i < d_i$, since the DU needs to be transmitted before its expiration.

Hence, DU i is associated with an attribute tuple $\psi_i = \{q_i, l_i, t_i, d_i\}$. In this section and the subsequent section, we assume that the attributes are known *a priori* for all DUs. In Section IV, we will discuss the case in which the attributes of all the future DUs are unknown to the wireless user, as is the case in real-time encoding and transmission scenarios. In this paper, we consider that the DUs are transmitted in the First In First Out (FIFO) fashion (i.e., the same as the encoding/decoding order).

During the transmission, DU i is delivered over the duration from time x_i to time y_i ($y_i \geq x_i$), where x_i represents the starting transmission time (STX) and y_i represents the ending transmission time (ETX). The choice of x_i and y_i represents the scheduling action of DU i, which is determined in the application layer. The scheduling action is to determine the STX x_i and the ETX y_i, and is denoted by (x_i, y_i) satisfying the condition of $t_i \leq x_i \leq y_i \leq d_i$. When the DU is scheduled for transmission during $[x_i, y_i]$, the wireless user experiences the average channel condition [channel gain or signal-to-noise ratio (SNR)] $c_i \in \mathbb{R}+$. For simplicity, we assume that the average channel condition is independent of the scheduled time (x_i, y_i), which can be the case when the wireless channel is slowly fading. The wireless user can then deploy the transmission action $a_i \in \mathcal{A}$ based on the experienced channel condition. The set $\mathcal{A}$ represents the possible transmission actions that the wireless user can choose and is assumed to be convex. One example is provided below. The consumed energy incurred by the transmission is denoted by $w_i(x_i, y_i, a_i)$. The distortion reduction due to the transmission is given by $q_i(1 - p_i(x_i, y_i, a_i))$, where $p_i(x_i, y_i, a_i)$ can be the probability that DU i is lost as in [15] or the distortion decaying function[4] due to partial data of DU i being received as in [18]. We can also interpret $Q_i(x_i, y_i, a_i) = q_i p_i(x_i, y_i, a_i)$[5] as the remaining distortion after the transmission. It is worth to note that $w_i(x_i, y_i, a_i)$ and $p_i(x_i, y_i, a_i)$ may also depend on the size l_i of DU i and the underlying channel condition c_i. Since both l_i and c_i are constant during the transmission of DU i, we omit them in the arguments of $w_i(x_i, y_i, a_i)$ and $p_i(x_i, y_i, a_i)$.

1) Example: The transmission action[6] is the amount of bits that can be successfully transmitted and $\mathcal{A} = [0, l_i]$. $p_i(x_i, y_i, a_i)$ is the distortion decaying function and is computed in [18] as $p_i(x_i, y_i, a_i) = 2^{-\theta_i a_i}$ where $\theta_i > 0$. By transmitting a_i bits of data in DU i, the incurred transmission energy is given as in [8]

$$w_i(x_i, y_i, a_i) = \frac{N_0}{c_i}\left(2^{(2a_i/(y_i - x_i)B)} - 1\right)(y_i - x_i)$$

where N_0 denotes the thermal noise, B is the bandwidth of the wireless link, and c_i represents the channel gain. ■

In addition, we assume that the functions $p_i(x_i, y_i, a_i)$ and $w_i(x_i, y_i, a_i)$ depend on x_i, y_i only through the difference $\tau_i = y_i - x_i$ and satisfy the following conditions:

C1 (Monotonicity): $p_i(x_i, y_i, a_i)$ is a nonincreasing function of the difference $\tau_i = y_i - x_i$ and the transmission action a_i.

C2 (Convexity): $p_i(x_i, y_i, a_i)$ and $w_i(x_i, y_i, a_i)$ are convex functions with respect to the joint variables $\tau_i = y_i - x_i$ and a_i.

Condition C1 means that the expected distortion will be reduced by increasing the difference $y_i - x_i$, since this results in a longer transmission time which increases the chance DU i will be successfully transmitted. In condition C2, the convexities of p_i and w_i are assumed to simplify the analysis. It is easy to show

[3]In real multimedia applications, the discard data can be concealed using previous received data. The error concealment algorithm can be easily incorporated into our proposed cross-layer optimization framework. In this paper, we do not consider such concealment algorithms at the decoder side.

[4]The distortion decaying function represents the fraction of the distortion remained after the (partial) data are successfully transmitted. For example, when the source is encoded in a scalable way, the distortion function is given by $D = Ke^{-\theta R}$ when R bits has been received [18]. In this case, the distortion decaying function is given as $p_i(x_i, y_i, a_i) = e^{-\theta_i R_i(x_i, y_i, a_i)}$ and $q_i = K$.

[5]We consider here that the distortion of the independently decodable DUs is not affected by other DUs, as in [20].

[6]This transmission action can be easily converted into the power allocation in the PHY in this example.

that $p_i(x_i, y_i, a_i)$ and $w_i(x_i, y_i, a_i)$[7] in the aforementioned example satisfy conditions C1 and C2.

Based on the description above, the cross-layer optimization for the delay-sensitive application over the time-varying wireless link is to find the optimal scheduling action (i.e., determining the STX x_i and ETX y_i for each DU) at the application layer and, under the scheduled time, the optimal transmission action a_i at the lower layer. The goal of the cross-layer optimization is to minimize the expected average remaining distortion experienced by the delay-sensitive application which is equivalent to maximizing the expected distortion reduction. This cross-layer optimization is also constrained on the total transmission energy at the PHY layer. Then, the cross-layer optimization problem with complete knowledge (referred to as CK-CLO) can be formulated as shown in the top equation at the bottom of the page, where the individual constraints $x_i \le y_i, x_i \ge t_i, y_i \le d_i$, $a_i \in \mathcal{A}$ are imposed for each DU which is independent of other DUs; the constraint $x_{i+1} \ge y_i$ indicates that DU $i+1$ has to be transmitted after DU i is transmitted (i.e., FIFO), and the last constraint in the CK-CLO problem indicates that the average consumed energy should not be larger than the budget W. It is easy to show that CK-CLO is a convex optimization problem because $Q_i(x_i, y_i, a_i)$ and $w_i(x_i, y_i, a_i)$ are the convex functions and the constraints in CK-CLO are also convex.

B. Decomposition for Cross-Layer Optimization

In this section, we discuss how the cross-layer optimization in the CK-CLO problem can be decomposed using duality theory [16]. This decomposition is important for developing optimal cross-layer solutions since it clearly shows how the packet scheduling action at the APP layer and transmission action at the lower layer can be jointly adapted for each DU. This decomposition further provides the necessary foundation to develop the online cross-layer optimization which is discussed in Section IV.

1) Lagrange Dual Problem: We first relax the constraints in the CK-CLO problem by introducing the Lagrange multiplier $\lambda \ge 0$ associated with the energy constraint and Lagrange multiplier vector $\mu = [\mu_1, \ldots, \mu_{M-1}]^T \ge 0$, whose elements are associated with the constraint $x_{i+1} \ge y_i, \forall i$. The corresponding Lagrange function is given as

$$L(x, y, a, \lambda, \mu) = \frac{1}{M}\sum_{i=1}^{M} Q_i(x_i, y_i, a_i) + \lambda\left(\frac{1}{M}\sum_{i=1}^{M} w_i(x_i, y_i, a_i) - W\right) + \sum_{i=1}^{M-1} \mu_i(y_i - x_{i+1}) \tag{1}$$

where $x = [x_1, \ldots, x_M]$, $y = [y_1, \ldots, y_M]$, and $a = [a_1, \ldots, a_M]$.

Then, the Lagrange dual function is given by (2) at the bottom of the page. The dual function shown in (2) corresponds to the cross-layer optimization under the individual constraints, given the Lagrange multipliers λ and μ. The dual problem (referred to as CK-DCLO) is then given by

$$\max_{\lambda \ge 0, \mu \ge 0} g(\lambda, \mu) \quad \text{(CK-DCLO)}$$

where $\mu \ge 0$ denotes the component-wise inequality. The dual problem aims to find the optimal Lagrange multipliers under which we can solve the optimization in the Lagrange function shown in (2). It can be shown [16] that, when the cross-layer optimization problem shown in CK-CLO is convex optimization, the optimal cross-layer action obtained from the Lagrange dual function with the optimal Lagrange multipliers is also the optimal solution to CK-CLO. In other words, the dual gap between CK-CLO and CK-DCLO is zero, which is shown in Section V-B. The optimal Lagrange multipliers can be obtained using the subgradient method as shown next.

The subgradients of the dual function at (λ, μ) are given [16] by

$$h_\lambda = \left(\frac{1}{M}\sum_{i=1}^{M} w_i\left(x_i^{\lambda,\mu}, y_i^{\lambda,\mu}, a_i^{\lambda,\mu}\right) - W\right)$$

[7]The convexity of $w_i(x_i, y_i, a_i)$ can be proved by showing that the Hessian matrix of $w_i(x_i, y_i, a_i)$ is semi-definite.

$$\begin{aligned} \min_{t_i \le x_i \le y_i \le d_i, a_i \in \mathcal{A}, i=1,\ldots,M} \quad & \frac{1}{M}\sum_{i=1}^{M} Q_i(x_i, y_i, a_i) \\ \text{s.t.} \quad & x_{i+1} \ge y_i, \ \forall\, i \in \{1, \ldots, M-1\} \\ & \frac{1}{M}\sum_{i=1}^{M} w_i(x_i, y_i, a_i) \le W. \qquad \text{(CK-CLO)} \end{aligned}$$

$$g(\lambda, \mu) = \min_{t_i \le x_i \le y_i \le d_i, a_i \in \mathcal{A}, i=1,\ldots,M} \left\{ \frac{1}{M}\sum_{i=1}^{M} Q_i(x_i, y_i, a_i) + \lambda\left(\frac{1}{M}\sum_{i=1}^{M} w_i(x_i, y_i, a_i) - W\right) + \sum_{i=1}^{M-1} \mu_i(y_i - x_{i+1}) \right\}. \tag{2}$$

Initialize $\lambda^0, \boldsymbol{\mu}^0, \ \lambda^1, \boldsymbol{\mu}^1, \ \varepsilon, \ k=1$
While $(|\lambda^k - \lambda^{k-1}| + \|\boldsymbol{\mu}^k - \boldsymbol{\mu}^{k-1}\| > \varepsilon$ or $k=1)$
  For $i = 1, \cdots, M$
    Solving DUCLO for DU i;
  End
  Compute $\lambda^{k+1}, \boldsymbol{\mu}^{k+1}$ as in Eqs. (3) and (4).
  $k \leftarrow k+1$
End

Algorithm 1: Algorithm for solving the CK-CLO problem.

with respect to the variable λ and $h_{\mu_i} = \left(y_i^{\lambda,\mu} - x_{i+1}^{\lambda,\mu}\right)$ with respect to the variable μ_i, where $x_i^{\lambda,\mu}$, $y_i^{\lambda,\mu}$, $a_i^{\lambda,\mu}$ is the optimal cross-layer solution in the dual function in (2) corresponding to the Lagrange multipliers λ, μ. The CK-DCLO problem can then be iteratively solved using the subgradients to update the Lagrange multipliers as follows.

Price Updating: See (3) at the bottom of the page and *NIF Updating*:

$$\mu_i^{k+1} = \left(\mu_i^k + \beta_i^k \left(y_i^{\lambda^k,\mu^k} - x_{i+1}^{\lambda^k,\mu^k}\right)\right)^+, \tag{4}$$

where $z^+ = \max\{z, 0\}$ and α^k and β_i^k are the update step size and satisfy the following conditions: $\sum_{k=1}^{\infty} \alpha^k = \infty$, $\sum_{k=1}^{\infty} \left(\alpha^k\right)^2 < \infty$ and $\sum_{k=1}^{\infty} \beta_i^k = \infty$, $\sum_{k=1}^{\infty} \left(\beta_i^k\right)^2 < \infty$[8] The proof of convergence is given in [16].

From the subgradient method, we note that the Lagrange multiplier λ is updated based on the consumed energy and available budget, which is interpreted as the "price" of the resource and it is determined at the lower layer, while the Lagrange multiplier vector μ is updated based on the scheduling time of the neighboring DUs, which is interpreted as the neighboring impact factors and is determined at the APP layer.

2) Decomposition for Lagrange Dual Function: Given the Lagrange multipliers λ and μ, the dual function shown in (2) is separable and can be decomposed into M DUCLO problems:

DUCLO problem $i \in \{1, \ldots, M\}$:

$$\min_{x_i \le y_i, x_i \ge t_i, y_i \le d_i, a_i \in \mathcal{A}} \frac{1}{M} Q_i(x_i, y_i, a_i) + \frac{\lambda}{M} w_i(x_i, y_i, a_i) - \mu_{i-1} x_i + \mu_i y_i \tag{5}$$

where $\mu_0 = 0$ and $\mu_M = 0$. Given the Lagrange multipliers λ and μ, each DUCLO problem is independently optimized. From (5), we note that all the DUCLO problems share the same Lagrange multiplier λ, since the budget constraint at the lower layer is imposed on all the DUs. We also note that DUCLO problem i shares the same Lagrange multiplier μ_{i-1} with DUCLO problem $i-1$ and μ_i with DUCLO problem $i+1$. Compared to the traditional myopic algorithm in which each DU is transmitted greedily without considering its impact on neighboring DUs as in [14], the DUCLO problems presented here automatically take into account the impact of the scheduling for the current DU on its neighbors. The impact between the independently decodable DUs takes place only through the Lagrange multipliers λ and μ.

Since $Q_i(x_i, y_i, a_i)$ and $w_i(x_i, y_i, a_i)$ are the convex function of $\tau_i = y_i - x_i$ and a_i, the DUCLO in (5) can be solved using the well-developed convex optimization methods [29]. It is easy to show that if $t_i \ge y_{i-1}$, then $x_i = t_i$ which means that DU $i-1$ is transmitted before DU i is available for transmission. If $t_i < y_{i-1}$, then $x_i = y_{i-1}$ which means that DU i is available for transmission before DU $i-1$'s transmission is stopped and immediately starts the transmission after DU $i-1$'s transmission is stopped. Hence, $x_i = \max\{y_{i-1}, t_i\}$. This observation will be used to develop the online optimization in Section IV.

In summary, the algorithm for solving the CK-CLO problem is illustrated in Algorithm 1.

III. Cross-Layer Optimization for Interdependent DUs

In this section, we consider the cross-layer optimization for interdependent DUs. Besides the attributes of each DU discussed in Section II-A, the interdependencies between DUs can be expressed using a DAG. One example for video frames is given in Fig. 1. (More examples can be found in [15].) Each node of the graph represents one DU and each edge of the graph directed from DU i to DU i' represents the dependence of DU i on DU i'. This dependency means that the distortion impact of DU i depends on the amount of successfully received data in DU i'. We can further define the partial relationship between two DUs which may not be directly connected, for which we write $i' \prec i$ if DU i' is an ancestor of DU i or equivalently DU

[8]These conditions are required to enforce the convergence of the subgradient method. The choice of α^k and β_i^k trades off the speed of convergence and performance obtained. One example is $\alpha^k = \beta_i^k = 1/k$.

$$\lambda^{k+1} = \left(\lambda^k + \alpha^k \left(\frac{1}{M}\sum_{i=1}^{M} w_i\left(x_i^{\lambda^k,\mu^k}, y_i^{\lambda^k,\mu^k}, a_i^{\lambda^k,\mu^k}\right) - W\right)\right)^+ \tag{3}$$

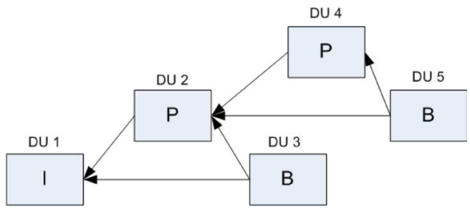

Fig. 1. DAG example with IBPBP video compressed frames.

i is a descendant of DU i' in the DAG. We further assume that if $i' \prec i$, then $t_{i'} \leq t_i$, which means that DU i' is encoded and available for transmission earlier than DU i. This assumption is reasonable since most of the current prediction-based coding schemes [25], [26] for the delay-sensitive applications actually satisfy this assumption. The relationship $i' \prec i$ means that the distortion (or error) is propagated from DU i' to DU i. Then, the average remaining distortion of DU i can be computed as

$$Q_i\left(x_i, y_i, a_i, \{x_k, y_k, a_k\}_{k \prec i}\right) = q_i - q_i\left(\left(1 - p_i\left(x_i, y_i, a_i\right)\right) \prod_{k \prec i}\left(1 - p_k\left(x_k, y_k, a_k\right)\right)\right) \quad (6)$$

where $\{x_k, y_k, a_k\}_{k \prec i}$ represents all the cross-layer actions of the DUs that DU i depends on, and $\prod_{k \prec i}(1 - p_k(x_k, y_k, a_k))$ is interpreted as the error propagation factor representing the impact of the cross-layer actions of all the DUs that DU i depend on, similar to the case in [15].

The primary problem of the cross-layer optimization for the interdependent DUs is the same as in the CK-CLO problem by replacing $Q_i(x_i, y_i, a_i)$ with $Q_i\left(x_i, y_i, a_i, \{x_k, y_k, a_k\}_{k \prec i}\right)$ in (6). The difference from the CK-CLO problem is that $Q_i\left(x_i, y_i, a_i, \{x_k, y_k, a_k\}_{k \prec i}\right)$ depends on the cross-layer actions of its ancestors and $Q_i\left(x_i, y_i, a_i, \{x_k, y_k, a_k\}_{k \prec i}\right)$ may not be a convex function of all the cross-layer actions $(x_k, y_k, a_k)\ \forall\ k \preceq i$, although $p_k(x_k, y_k, a_k)$ is a convex function of (x_k, y_k, a_k). However, we note that, given $(x_k, y_k, a_k) \forall k \prec i$ (i.e., $\prod_{k \prec i}(1 - p_k(x_k, y_k, a_k))$ is constant), $Q_i\left(x_i, y_i, a_i, \{x_k, y_k, a_k\}_{k \prec i}\right)$ is a convex function of (x_i, y_i, a_i). We will use this property to develop a dual solution for the original nonconvex problem and we will quantify the duality gap in the simulation section.

The derivative of the dual problem is the same as the one in Section II-B. By replacing $Q_i(x_i, y_i, a_i)$ with $Q_i\left(x_i, y_i, a_i, \{x_k, y_k, a_k\}_{k \prec i}\right)$ in (6), the Lagrange dual function shown in (2) becomes (7), shown at the bottom of the page.

Due to the interdependency, this dual function cannot be simply decomposed into the independent DUCLO problems as shown in (5). However, the dual function can be computed DU by DU assuming the cross-layer actions of other DUs is given, as shown in [15]. Specifically, given the Lagrange multipliers λ, μ, the objective function in (7) is denoted as $G((x_1, y_1, a_1), \ldots, (x_M, y_M, a_M), \lambda, \mu)$. When the cross-layer actions of all DUs except DU i are fixed, the DUCLO for DU i is given by (8) at the bottom of the page where [see (9) at the bottom of the next page], and θ_i represents the remaining part in (7), which does not depend on the cross-layer action (x_i, y_i, a_i). Note that, since we fix the cross-layer actions of all other DUs, we write Q_i' as a function of only (x_i, y_i, a_i). It is easy to show that the optimization over the cross-layer action of DU i in (8) is a convex optimization, which can be solved using the well-developed convex optimization methods [29].

As discussed in [15], $Q_i'(x_i, y_i, a_i)$ can be interpreted as the sensitivity to (or impact of) the imperfect transmission of DU i, i.e., the amount by which the expected distortion will increase if the data of DU i is fully received, given the cross-layer actions of other DUs. It is clear that the DUCLO for DU i is solved only by fixing the cross-layer actions of other DUs, unlike the solutions for the independently decodable DUs which do not require the knowledge of other DUs.

A local optimal cross-layer action to the optimization in (7) can be obtained using the block coordinate

$$g(\lambda, \mu) = \min_{x_i \leq y_i, x_i \geq t_i, y_i \leq d_i, a_i \in \mathcal{A}, i=1,\ldots,M} \left\{ \begin{array}{l} \frac{1}{M} \sum_{i=1}^{M} \left(q_i - q_i\left(1 - p_i\left(x_i, y_i, a_i\right)\right) \prod_{k \prec i}\left(1 - p_k\left(x_k, y_k, a_k\right)\right)\right) \\ +\lambda\left(\frac{1}{M} \sum_{i=1}^{M} w_i\left(x_i, y_i, a_i\right) - W\right) + \sum_{i=1}^{M-1} \mu_i\left(y_i - x_{i+1}\right) \end{array} \right\}. \quad (7)$$

$$\begin{aligned} &\min_{x_i \leq y_i, x_i \geq t_i, y_i \leq d_i, a_i \in \mathcal{A}} G\left(\left(x_1, y_1, a_1\right), \ldots, \left(x_i, y_i, a_i\right), \ldots, \left(x_M, y_M, a_M\right), \lambda, \mu\right) \\ &= \min_{x_i \leq y_i, x_i \geq t_i, y_i \leq d_i, a_i \in \mathcal{A}} \left(\frac{1}{M} Q_i'\left(x_i, y_i, a_i\right) + \frac{\lambda}{M} w_i\left(x_i, y_i, a_i\right) - \mu_{i-1} x_i + \mu_i y_i\right) + \theta_i \end{aligned} \quad (8)$$

Initialize: $W_{left} = W$.

For $i = 1,\cdots,M$

if $i = 1$ $x_i^p = x_i^d, y_i^p = y_i^d, a_i^p = a_i^d$

Else $x_i^p = \max\{x_i^d, y_{i-1}^d\}, y_i^p = \min\{y_i^d, y_{i-1}^d\}, a_i^p = a_i^d$;

End

$W_{left} \leftarrow W_{left} - w_i(x_i^p, y_i^p, a_i^p)$;

If $W_{left} \leq 0$ $a_i^p = 0$; **End**

End

Algorithm 2: Algorithm for deriving the feasible primary cross-layer solution form the dual solution.

descent method [16], as described next. Given the current optimizer $((x_1^n, y_1^n, a_1^n), \ldots, (x_M^n, y_M^n, a_M^n))$ at iteration n, the optimizer at iteration $n+1$, $((x_1^{n+1}, y_1^{n+1}, a_1^{n+1}), \ldots, (x_M^{n+1}, y_M^{n+1}, a_M^{n+1}))$ is generated according to the iteration

$$\begin{aligned}(x_i^{n+1}, y_i^{n+1}, a_i^{n+1}) &= \arg\min_{x_i \leq y_i, x_i \geq t_i, y_i \leq d_i, a_i \in \mathcal{A}} G\Big((x_1^{n+1}, y_1^{n+1}, a_1^{n+1}), \ldots, (x_{i-1}^{n+1}, y_{i-1}^{n+1}, a_{i-1}^{n+1})\\ &\quad (x_i, y_i, a_i), (x_{i+1}^n, y_{i+1}^n, a_{i+1}^n), \ldots\\ &\quad (x_M^n, y_M^n, a_M^n), \lambda, \mu\Big). \end{aligned} \tag{10}$$

At each iteration, the objective function is decreased compared to that of the previous iteration and the objective function is lower bounded (greater than zero). Hence, this block coordinate descent method converges to the locally optimal solution to the optimization in (7), given the Lagrange multipliers λ and μ.

We note that, for this nonconvex cross-layer optimization, the dual solution developed above may not satisfy the desired constraints: $x_{i+1} \geq y_i, \forall\, i \in \{1, \ldots, M-1\}$ and $(1/M)\sum_{i=1}^M w_i(x_i, y_i, a_i) \leq W$. However, we can simply derive a feasible solution to the original cross-layer optimization from the optimal dual solution.

Assuming that the cross-layer actions associated with the optimal dual solution λ^*, μ^* are $(x_i^d, y_i^d, a_i^d), \forall\, i \in \{1, \ldots, M\}$. Then, (x_i^d, y_i^d, a_i^d) satisfies the individual constraints: $x_i^d \leq y_i^d$, $x_i^d \geq t_i$, $y_i^d \leq d_i$, $a_i^d \in \mathcal{A}$. Algorithm 2 provides a method to generate the feasible primary cross-layer solution (x_i^p, y_i^p, a_i^p), $\forall\, i \in \{1, \ldots, M\}$.

IV. Online Cross-Layer Optimization With Incomplete Knowledge

The cross-layer optimization formulated in Sections II and III assumes complete *a priori* knowledge of the DUs' attributes and the channel conditions. However, in real-time applications, this knowledge is available only right before the DUs are transmitted. Furthermore, the cross-layer optimization algorithms based on the decomposition principles presented in Sections II-B and III require multiple iterations (as shown in Sections V-B and C) to converge, which may be difficult to implement for real-time applications. To deal with the real-time transmission scenario, we propose a low-complexity online cross-layer optimization algorithm motivated by the decomposition principles developed in Sections II-B and III.

A. Online Optimization Using Learning for Independent DUs

In this section, we consider the case in which the DUs can be independently decoded and that the attributes and channel conditions dynamically change over time. The random versions of the arrival time, delay deadline, DU size, distortion impact and channel condition are denoted by T_i, D_i, L_i, $\mathcal{Q}_i$, C_i, respectively. We assume that both the interarrival interval (i.e., $T_{i+1} - T_i$) and the life time (i.e., $D_i - T_i$) of the DUs are i.i.d. The other attributes of each DU and the experienced channel condition are also i.i.d. random variables independent of other DUs. We further assume that the user has an infinite number of DUs to transmit. Let $\Omega_i = \{T_i, D_i, L_i, \mathcal{Q}_i, C_i\}$. Then, the cross-layer optimization with complete knowledge presented in the CK-CLO problem becomes a cross-layer optimization with incomplete knowledge (referred to as ICK-CLO) as shown in the top equation at the bottom of the next page, where $\Delta_i = \{(x_i, y_i, a_i) \,|\, T_i \leq x_i \leq y_i \leq D_i, x_i \geq y_{i-1}, a_i \in \mathcal{A}\}$ is

$$\begin{aligned} Q_i'(x_i, y_i, a_i) &= \frac{1}{M} q_i p_i(x_i, y_i, a_i) \prod_{k \prec i} (1 - p_k(x_k, y_k, a_k)) \\ &\quad - (1 - p_i(x_i, y_i, a_i)) \left(\sum_{i' \succ i} q_{i'} (1 - p_{i'}(x_{i'}, y_{i'}, a_{i'})) \prod_{\substack{k \prec i' \\ k \neq i}} (1 - p_k(x_k, y_k, a_k)) \right) \end{aligned} \tag{9}$$

the set of feasible cross-layer actions for DU i, which depends on y_{i-1} and Ω_i. We note that the decision on the cross-layer action (x_i, y_i, a_i) is performed after knowing all the cross-layer actions (x_k, y_k, a_k) of DUs k with $k < i$ and the realization of $\Omega_i = \{T_i, D_i, L_i, \mathcal{Q}_i, C_i\}$. It is easy to show that the optimization in the ICK-CLO problem is the same as the CK-CLO problem (i.e., if Ω_i is deterministic, the expectation operations disappear and the minimization operations can be taken out and put in the front of limitation) except that the ICK-CLO problem minimizes the expected average distortion for the infinite number of DUs over the expected average energy constraint. However, the solution to the ICK-CLO problem is quite different from the solution to the CK-CLO problem. The ICK-CLO problem can be formulated as a constrained MDP [30] problem, which is formally presented below.

1) Constrained MDP Formulation: From the assumption presented at the beginning of Section IV-A, we note that $T_{i+1} - T_i$, $D_i - T_i$, C_i and other attribute of DU i are i.i.d. random variables. Hence, for the independently decodable DUs, if we know the value of T_i, the attributes and channel conditions of all the future DUs (including DU i) are independent of the attributes and channel conditions of previous DUs. From the observation in Section II-B-II), we know that the x_i satisfies $x_i = \max(y_{i-1}, t_i)$, which is further demonstrated in Fig. 2. Hence, DU $i-1$ will impact the cross-layer action selection of DU i only through ETX y_{i-1}. In other words, DU $i-1$ brings forward or postpones the transmission of DU i by determining its ETX y_{i-1}. If we define a state for DU i as $s_i = \max(y_{i-1} - t_i, 0)$, then the impact from previous DUs is fully characterized by this state. Knowing the state s_i, the cross-layer optimization of DU i is independent of the previous DUs. This observation motivates us to model the cross-layer optimization for the time-varying DUs as a constrained MDP [30] in which the state transition from state s_i to state s_{i+1} is

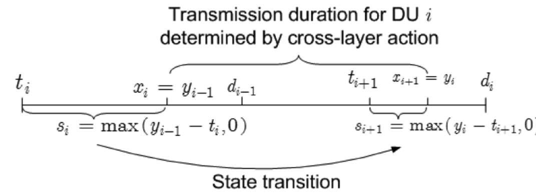

Fig. 2. State of DU i and state transition from DU i to DU $i+1$.

determined only by the ETX y_i of DU i and the time t_{i+1} DU $i+1$ is ready for transmission, i.e., $s_{i+1} = \max(y_i - t_{i+1}, 0)$. The action in this MDP formulation is the STX x_i, ETX y_i, and the action a_i.

Similar to the dual problem presented in Section II-B, the constrained MDP can also be solved via the dual solution [30]. The dual problem (referred to as ICK-DCLO) corresponding to the ICK-CLO problem is given by the following optimization:

$$\max_{\lambda \geq 0} g(\lambda) \quad \text{(ICK-DCLO)}$$

where $g(\lambda)$ is computed by the following optimization [see (11) at the bottom of the page], where $\Phi_i(x_i, y_i, a_i) = Q_i(x_i, y_i, a_i) + w_i(x_i, y_i, a_i)$ and the Lagrange multiplier λ is associated with the expected average resource constraint, which is the same as the one in (1). Once the optimization in (11) is solved, the Lagrange multiplier is then updated as follows: see (12) at the bottom of the next page where $\left(x_i^\lambda, y_i^\lambda, a_i^\lambda\right)$ is the optimal cross-layer action corresponding to the Lagrange multiplier λ.

Hence, in the following, we focus on the optimization in (11). Based on the discussion at the beginning of this section, we know that the dual function in (11) corresponds to the unconstrained MDP which can be solved using dynamic programming [17]. Specifically, given the resource price λ, the optimal policy

$$\lim_{N\to\infty} \frac{1}{N} \left\{ \underset{\Omega_1}{E} \left[\begin{array}{c} \min\limits_{(x_1,y_1,a_1)\in\Delta_1} Q_1(x_1,y_1,a_1) + \cdots \\ + \underset{\Omega_{N-1}}{E} \left[\begin{array}{c} \min\limits_{(x_{N-1},y_{N-1},a_{N-1})\in\Delta_{N-1}} Q_{N-1}(x_{N-1},y_{N-1},a_{N-1}) + \\ \underset{\Omega_N}{E} \left[\min\limits_{(x_N,y_N,a_N)\in\Delta_N} Q_N(x_N,y_N,a_N) \right] \end{array} \right] \end{array} \right] \right\}$$

$$s.t. \lim_{N\to\infty} \frac{1}{N} \left\{ \underset{\Omega_1}{E} \left[\begin{array}{c} w_1(x_1,y_1,a_1) + \cdots + \\ \underset{\Omega_{N-1}}{E} \left[\begin{array}{c} w_{N-1}(x_{N-1},y_{N-1},a_{N-1}) + \\ \underset{\Omega_N}{E} \left[w_N(x_N,y_N,a_N) \right] \end{array} \right] \end{array} \right] \right\} \leq W \qquad \text{(ICK-CLO)}$$

$$g(\lambda) = \lim_{N\to\infty} \frac{1}{N} \left\{ \underset{\Omega_1}{E} \left[\begin{array}{c} \min\limits_{(x_1,y_1,a_1)\in\Delta_1} \Phi_1(x_1,y_1,a_1) + \cdots \\ + \underset{\Omega_{N-1}}{E} \left[\begin{array}{c} \min\limits_{(x_{N-1},y_{N-1},a_{N-1})\in\Delta_{N-1}} \Phi_{N-1}(x_{N-1},y_{N-1},a_{N-1}) + \\ \underset{\Omega_N}{E} \left[\min\limits_{(x_N,y_N,a_N)\in\Delta_N} \Phi_N(x_N,y_N,a_N) \right] \end{array} \right] \end{array} \right] \right\} - \lambda W \qquad (11)$$

(i.e., the optimal cross-layer action at each state) for the optimization in (11) satisfies the dynamic programming equation [17], which is given by (13) at the bottom of the page where $V(s)$ represents the state-value function at state s and the difference $V(s) - V(0)$ represents the total impact that the previous DU impose on all the future DUs by delaying the transmission of the next DU by s seconds; t is the time the current DU is ready for transmission; and β is the optimal average cost, which is the value computed in (11). It is easy to show [31] that $V(s)$ is a nondecreasing convex function of s because the larger the state s, the larger the delay in transmission of the future DUs, and therefore the larger the distortion.

A well-known relative value iteration algorithm (RVIA) [17] exists for solving the dynamic programming equation in (13), which is given by (14) at the bottom of the page where $V_n(\cdot)$ is the state-value function obtained at the iteration n.

In the CK-CLO problem, the solution is obtained assuming complete knowledge about the DUs' attributes and the experienced channel conditions. Hence, in the DUCLO for the CK-CLO problem, the impact on the neighboring DUs is fully characterized by the scalar numbers μ_{i-1} and μ_i. The cross-layer action selection for each DU is based on the assumption that the cross-layer actions for neighboring DUs (previous and future DUs) are fixed. However, in the ICK-CLO problem, the cross-layer action selection for each DU is based on the assumption that the cross-layer actions for the previous DUs are fixed (i.e., the sate s is fixed) and the future DUs (and the cross-layer actions for them) are unknown. The impact from the previous DUs is characterized by the state s and the impact on the future DUs is characterized by the state value function $V(s)$.

2) Online Cross-Layer Optimization Using Learning: Although the ICK-CLO is solved using the dual solution in (12) and (14), it requires to know the distributions of the attributes of DUs and the underlying channel conditions which are often difficult to accurately characterize. Instead, in this section, we develop an online learning to update the state-value function $V(s)$ in (14) and the resource price λ in (12) without knowing the distributions *a priori*. Assume that, before the cross-layer optimization for DU i, the estimated state-value function and resource price are denoted by $V_i(s)$ and λ_i. Then the cross-layer optimization for DU i is given by

$$\begin{aligned}\min_{x_i,y_i,a_i}\quad & Q_i(x_i,y_i,a_i)+\lambda_i w_i(x_i,y_i,a_i)\\ & +V_i(\max(y_i-t_{i+1},0))\\ \text{s.t.}\quad & x_i=s_i+t_i, y_i\le d_i, a_i\in\mathcal{A}\end{aligned} \tag{15}$$

which can be solved similar to the DUCLO in Section II-B since this optimization is convex. The remaining question is how we can choose the right price of resource λ_i and estimate the state-value function $V_i(s)$.

We notice that $V_i(s)$ is a function of the continuous state s and hence, it cannot be directly updated at each visited state as the reinforcement learning with the discrete state space [27]. To overcome this obstacle, we use a function approximation method similar to the work in [19] to approximate the state-value function by a finite number of parameters. Then, instead of updating the state-value function at each state, we update the finite parameters of the state-value function. Specifically, the state-value function $V(s)$ is approximated by a linear combination of a set of feature functions:

$$V(s)\approx\begin{cases}\sum_{k=1}^{K} r^k v^k(s) & \text{if } s\ge 0\\ 0 & \text{o.w.}\end{cases} \tag{16}$$

where $r=\left[r^1,\ldots,r^K\right]'\ge 0$ is the parameter vector; $v(s)=\left[v^1(s),\ldots,v^K(s)\right]'$ is a vector function with each element being a scalar convex feature function of s [19]; and K is the number of feature functions used to represent the impact function. The larger the value K is, the more accurate this approximation may be. However, the large K requires more memory to store the parameter vector. We enforce the feature functions to

$$\lambda^{k+1}=\left\{\lambda^k+\alpha^k\left(\lim_{N\to\infty}\frac{1}{N}\left\{\underset{\Omega_1}{E}\left[\begin{matrix}w_1\left(x_1^\lambda,y_1^\lambda,a_1^\lambda\right)+\cdots+\\ \underset{\Omega_{N-1}}{E}\left[\begin{matrix}w_{N-1}\left(x_{N-1}^\lambda,y_{N-1}^\lambda,a_{N-1}^\lambda\right)+\\ \underset{\Omega_N}{E}\left[w_N\left(x_N^\lambda,y_N^\lambda,a_N^\lambda\right)\right]\end{matrix}\right]\end{matrix}\right]\right\}-W\right)\right\}^{+}. \tag{12}$$

$$V(s)=\underset{D,L,Q,C,T}{E}\left\{\min_{x=s+t,y<D,a\in\mathcal{A}}\left[Q(x,y,a)+\lambda w(x,y,a)+V(\max(y-T,0))\right]\right\}-\beta \tag{13}$$

$$V_{n+1}(s)=\underset{D,L,Q,C,T}{E}\left\{\max_{x=s+t,y<D,a\in\mathcal{A}}\left[Q(x,y,a)+\lambda w(x,y,a)+V_n(\max(y-T,0))\right]\right\}-V_n(0) \tag{14}$$

Initialize $\lambda_1, r_1 = 0, \ s_1 = 0, \ i = 1$

For each DU i

Observe the attributes and network condition of DU i and the time t_{i+1} at which DU $i+1$ is ready for transmission;

Solving the DUCLO given in Eq. (15);

Update $s_{i+1} = \max(y_i - t_{i+1}, 0)$, λ_{i+1} as in Eq. (18) and r_{i+1} as in Eq. (17);

$i \leftarrow i+1$

End

Algorithm 3: Proposed online optimization using learning.

be convex in order to ensure that the approximated state value function is still convex with respect to the state s. The feature functions should be linearly independent. In general, the state-value function $V(s)$ may not be in the space spanned by these feature functions. For simplicity, in this paper, we choose $v(s) = \left[s^1, \ldots, (s^K/K!)\right]'$ as the feature functions[9]. Similar to the time difference learning in [19], the parameter vector $r = \left[r^1, \ldots, r^K\right]'$ is then updated as follows: see (17) at the bottom of the page where γ_i satisfies $\sum_{i=1}^{\infty} \gamma_i = \infty, \sum_{i=1}^{\infty} (\gamma_i)^2 < \infty$.

Similar to the price update in Section II-B, the online update for λ is given as follows:

$$\lambda_{i+1} = \left(\lambda_i + \kappa_i \left(\frac{1}{i}\sum_{j=1}^{i} w_j - W\right)\right)^+ \tag{18}$$

where κ_i satisfies $\sum_{j=1}^{\infty} \kappa_j = \infty$, $\sum_{j=1}^{\infty} (\kappa_j)^2 < \infty$, $\lim_{j\to\infty}(\kappa_j/\gamma_j) = 0$. The update for λ is based on the average consumed energy up to DU i. If the average consumed energy is greater than the budget W, the resource price will increase in order to decrease the energy consumption for next DU transmission, and vice versa.

We should note that, in this proposed learning algorithm, the cross-layer action of each DU is optimized based on the estimated state-value function and resource price after the previous DU transmission. Then the state-value function is updated based on the current optimized result. Hence, this learning algorithm does not explore the entire cross-layer action space like the Q-learning algorithm [27] and may only converge to the local solution. However, in the simulation section, we will show that it can achieve the similar performance to the CK-CLO with $M = 10$, which means that the proposed online learning algorithm can forecast the impact of current cross-layer action on the future DUs by updating the state-value function.

[9]How to select the optimal feature functions is part of our future research.

The convergence of the resource price and state-value function (to the local optimal points) can be developed based on the function approximation [19] and the two time-scale stochastic approximation [22], [32]. The key idea behind the convergence proof is characterized as follows: in (17) and (18), the updates of the state-value function $V(s)$ and the resource price λ are performed using different step sizes. The step sizes satisfy $\lim_{j\to\infty}(\kappa_j/\gamma_j) = 0$, which means that the update rate of the state-value function is faster than that of the resource price. In other words, for each resource price, the state-value function $V(s)$ will approximately converge to the optimal value corresponding to the current resource price since it is updated at the faster time scale. On the other hand, from the perspective of the state-value function, the resource price appears to be almost constant. This two time-scale update ensures that the state-value function and resource price converge. The algorithm for the proposed online optimization using learning is illustrated in Algorithm 3.

B. Online Optimization for Interdependent DUs

In this section, we consider the online cross-layer optimization for the interdependent DUs as discussed in Section III. In order to take into account the dependencies between DUs, we assume that the DAG of all DUs is known *a priori*. This assumption is reasonable since, for instance, the GOP structure in video streaming is often fixed. When optimizing the cross-layer action (x_i, y_i, a_i) of DU i, the cross-layer actions and transmission results $p_k(x_k^*, y_k^*, a_k^*)$ of DUs with index $k < i$ have been determined. Then, the sensitivity $Q_i'(x_i, y_i, a_i)$ of DU i is computed, based on the current knowledge, as follows: see (19) at the bottom of the next page where $\tilde{q}_{i'}$ is the estimated distortion impact of DU i and $\tilde{p}_{i'}(x_{i'}, y_{i'}, a_{i'})$ for DU i is simply set to be 0 which means that we assume that the future DU i' can

$$r_{i+1} = (1-\gamma_i) r_i + \frac{\gamma_i \left\{ \max\limits_{x_i = s_i, y_i < d_i, a_i \in \mathcal{A}} \left[Q_i(x_i, y_i, a_i) + \lambda w_i(x_i, y_i, a_i) + V_i(\max(y_i - t_{i+1}, 0)) \right] - V_i(0) \right\}}{\{K v(s_i)\}} \tag{17}$$

be successfully received. Similar to the online cross-layer optimization for independent DUs given in Section IV-A, the online optimization for the interdependent DUs is given as follows:

$$\begin{aligned} \min_{x_i, y_i, a_i} \quad & Q_i'(x_i, y_i, a_i) + \lambda w_i(x_i, y_i, a_i) \\ & + V_i(\max(y_i - t_{i+1}, 0)) \\ \text{s.t.} \quad & x_i = s_i + t_i, y_i \leq d_i, a_i \in \mathcal{A}. \end{aligned} \tag{20}$$

The update of the parameter vector r and the resource price λ is the same as in (17) and (18).

V. Numerical Results

In this section, we present our numerical results to evaluate the proposed decomposition method and the online algorithm.

A. Models for Distortion Impact and Energy Cost Functions

In this example, we consider the proposed cross-layer optimization solution to determine the optimal scheduling and energy allocation for DUs with various attributes at the application layer transmitted over a time-varying channel at the PHY layer, as shown in the Example in Section II-A.

In this example, the distortion impact q_i is the realization of a uniformly distributed random variable in the range of $[50, 150]$. The DU size l_i is assumed to be constant and equals 10 000 bits. The varying DU size is considered in Section V-F for video streaming. The arrival interval $t_i - t_{i-1}$ is the realization of an exponentially distributed random variable with the mean of 50 ms. The DU lifetime $d_i - t_i$ is 50 ms. The parameter θ_i equals 0.5. We will verify the efficiency of the proposed methods using the model developed in this section in Sections V-B–E. We will further consider a more realistic scenario with video streaming in Section V-F.

B. Dual and Primal Solutions and Duality Gap for Independent DUs

Fig. 3(a) shows the duality gap between the dual solutions and primal solutions over 110 iterations in a setting with independent DUs. It is shown that the duality gap goes to zero after around 100 iterations, which demonstrates that the subgradient algorithm developed in Section II-B converges to the optimal total expected distortion given by the primal solutions. Fig. 3(b) further shows that the primal and dual solutions are equivalent. However, the subgradient method requires around 100 iterations to converge to the optimal solutions, which may be hard to implement in the real-time applications (e.g., video streaming) since it requires a lot of computation. Hence, in Section IV, we have developed an online algorithm which can significantly reduce the complexity of the cross-layer optimization (i.e., one iteration) and only use the current available information. The simulation results for the online algorithms are presented in Section V-D.

C. Dual and Primal Solutions and Duality Gap for the Interdependent DUs

Fig. 4(a) shows the duality gap between the dual solutions and primal solutions for the interdependent DUs with $M = 10$. Although the cross-layer optimization problem for the interdependent DUs is not a convex optimization, it is shown here that the duality gap in this example goes to zero after around 230 iterations, which demonstrates that the subgradient algorithm developed in Section II-B also converges in the cross-layer optimization for interdependent DUs. The subgradient algorithm for the interdependent DUs requires two types of iterations: one is the outer iteration which updates the price of the resource λ and NIFs μ and the other one is the inner iteration which is to find the optimal cross-layer action for each DU given λ and μ as shown in (10). Fig. 4(b) shows the required number of inner iterations per outer iteration using the cross-layer actions obtained in the previous outer iteration as the starting point in the current outer iteration. It is clear that 2–6 inner iterations are required for each outer iteration to converge to the optimal cross-layer actions given λ and μ. Hence, the subgradient method requires a total of 651 inner iterations, which is unacceptable for the real-time applications (e.g., video streaming). As discussed in Section V-B, this motivates us to develop an online algorithm which was presented in Section IV. The simulation results for the online algorithm are presented in Section V-E.

D. Online Cross-Layer Optimization for Independent DUs

In this simulation, we consider three cross-layer optimization algorithms for the scenario with independent DUs. The first one is the online cross-layer optimization for each DU proposed in Section IV. The second performs the cross-layer optimization every $M = 10$ DUs by assuming complete knowledge of these

$$\begin{aligned} & Q_i'(x_i, y_i, a_i) \\ & \quad = q_i p_i(x_i, y_i, a_i) \underbrace{\prod_{k \prec i} (1 - p_k(x_k^*, y_k^*, a_k^*))}_{known} \\ & \quad - (1 - p_i(x_i, y_i, a_i)) \left(\underbrace{\sum_{i' \succ i} \tilde{q}_{i'} (1 - \tilde{p}_{i'}(x_{i'}, y_{i'}, a_{i'})) \prod_{\substack{j \prec i' \\ j > i}} (1 - \tilde{p}_j(x_j, y_j, a_j))}_{estimated} \underbrace{\prod_{\substack{k \prec i' \\ k < i}} (1 - p_k(x_k^*, y_k^*, a_k^*))}_{known} \right) \end{aligned} \tag{19}$$

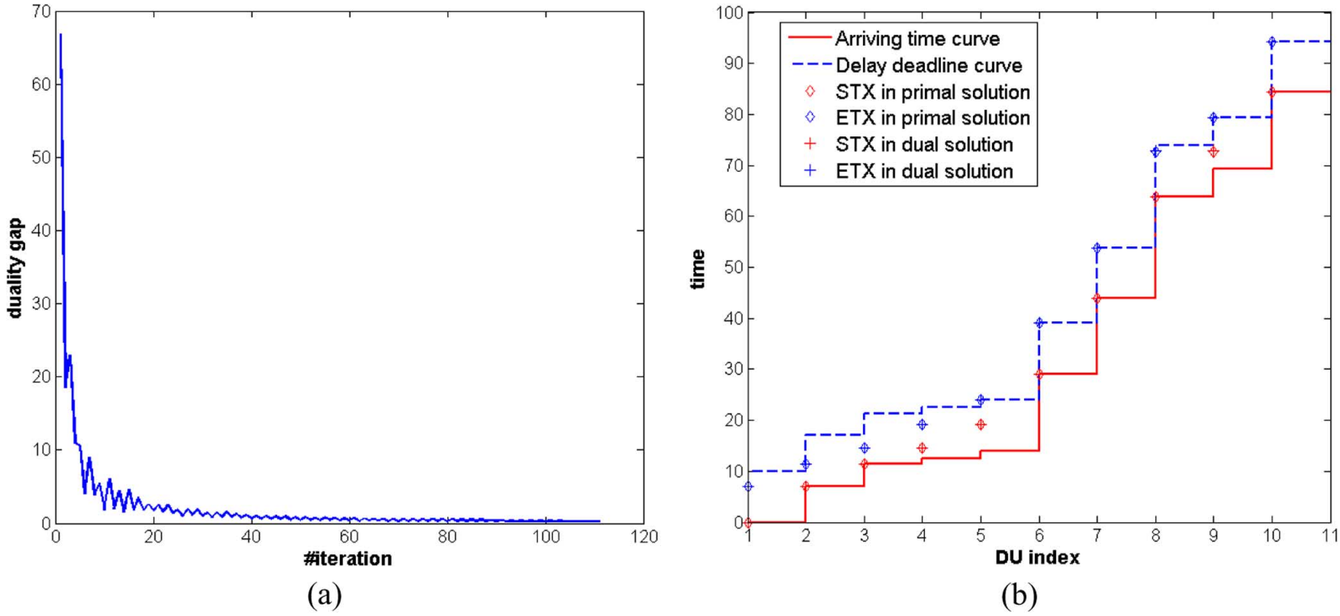

Fig. 3. (a) Duality gap between the dual and primal solutions for independent DUs. (b) Dual and primal optimal scheduling time for independent DUs.

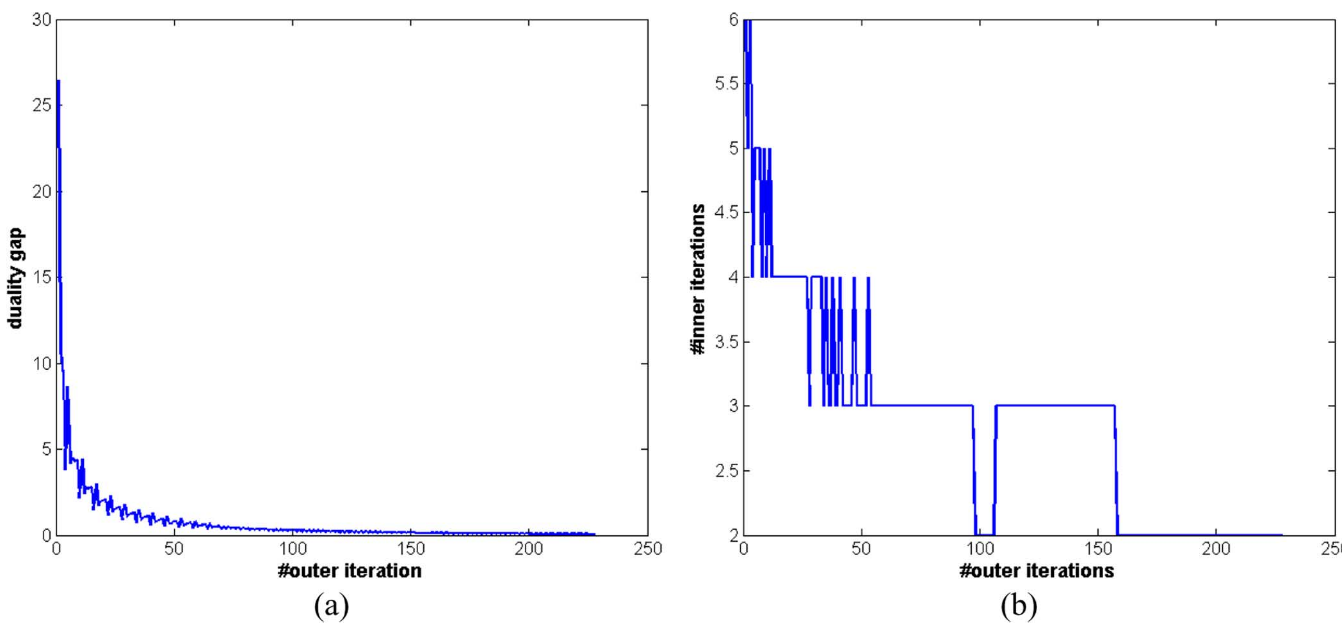

Fig. 4. (a) Duality gap between the dual and primal solutions for interdependent DUs. (b) Number of inner iterations per outer iterations for the cross-layer optimization of interdependent DUs.

M DUs' attributes and underlying channel conditions (we call this the "oracle" cross-layer optimization). The third one performs the cross-layer optimization for each DU (i.e., $M = 1$, called myopic online optimization). We will refer to the transmission of 10 DUs as one cycle.

Fig. 5 depicts the distortion reduction of each cycle (one cycle corresponds to $M = 10$ DUs.) under various resource constraints for these three algorithms. From this figure, we note that, on the one hand, the online cross-layer optimization proposed in Section IV outperforms the myopic online optimization by around 6% for various energy constraints because the proposed online optimization can predict the impact on the future DUs through the state-value function and allocate the energy for each cycle based on the importance of DUs. On the other hand, the "oracle" cross-layer optimization outperforms the proposed online cross-layer optimization by around 4% since the "oracle" cross-layer optimization explicitly considers the exact information of future DUs which is not available in the online cross-layer optimization. However, the proposed online cross-layer optimization has the following advantages, compared to the "oracle" cross-layer optimization: (i) it performs the cross-layer optimization for each DU and updates λ and state-value function $V(s)$ for each DU without requiring multiple iterations, which significantly reduces the computational complexity; (ii) it does not require exact information about the future DUs' attributes and channel conditions.

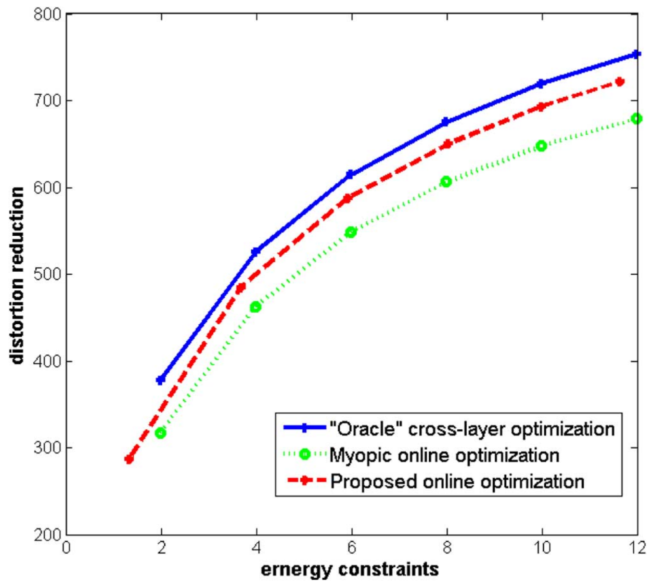

Fig. 5. The distortion reduction under various energy constraints for independent DUs.

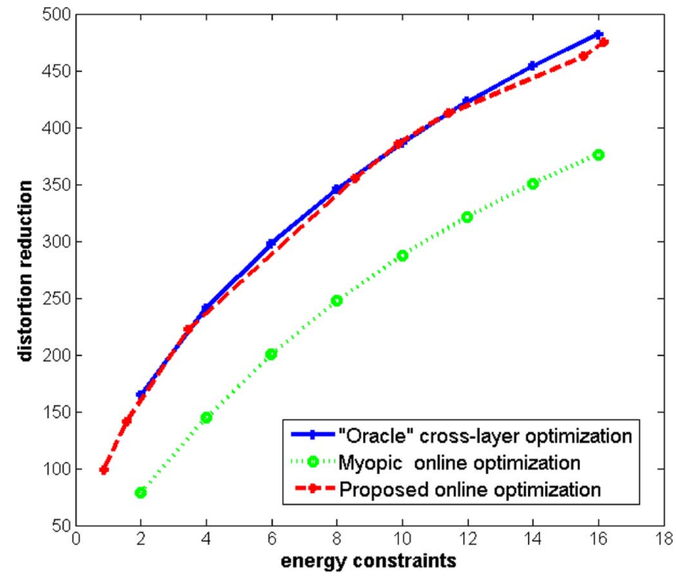

Fig. 6. Distortion reduction under various energy constraint for interdependent DUs.

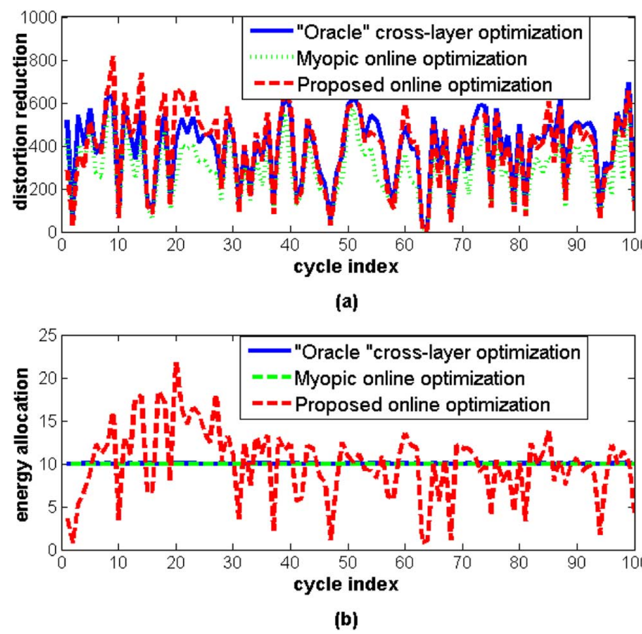

Fig. 7. (a) Distortion reduction. (b) Average energy consumption for each cycle.

E. Online Cross-Layer Optimization for Interdependent DUs

In this simulation, we also consider three online algorithms as described in Section V-D for the scenario with interdependent DUs. The interdependencies (represented by a DAG) are generated randomly every 10 DUs. The interdependency between DUs happens only within one cycle [for instance, a cycle could represent one group of pictures (GOP) of the video sequences]. Fig. 6 shows the distortion reduction of each cycle under various energy constraints. From this figure, we note that, for interdependent DUs, our proposed online cross-layer optimization can significantly improve the performance (more than 28% increased) compared to the myopic online optimization, and has similar performance as the "oracle" cross-layer optimization. We further show the distortion reduction and energy allocation for each cycle when the average energy constraint is 10 (i.e., $W = 10$) in Fig. 7. From this figure, we observe that, after the initial learning stage (about 30 cycles), our proposed online solution achieves the similar performance to the "oracle" solution. We will also verify this observation in a more realistic scenario which is presented in Section V-F. The reason that our proposed solution can have similar performance to the "oracle" solution is as follows: for the interdependent DUs, the amount of the distortion reduction is mainly determined by the important DUs (on which many other DUs depend on) and our solution can ensure that more important DUs are successfully transmitted by allocating more energy to them.

F. Online Cross-Layer Optimization for Video Streaming

In this simulation, we consider a communication scenario in which the wireless user streams the video sequence "Foreman" or "Coastguard" (CIF resolution, 30 Hz) over the time-varying wireless channel. For the compression of the video sequence, we used a scalable video coding scheme [25]. Such scalable video compression is attractive for wireless streaming applications because it provides on-the-fly adaptation to channel conditions, support for a variety of wireless receivers with different resource capabilities and power constraints, and easy prioritization of various coding layers and video packets. We compare four different cross-layer optimization methods: "oracle" cross-layer optimization with $M = 8$ (i.e., cross-layer optimization with complete knowledge), cross-layer optimization given constant channel conditions, myopic online optimization, and the proposed online optimization. The cross-layer optimization given constant channel conditions is performed similarly to the "oracle" cross-layer optimization, but assuming that the video data experiences a constant channel condition, which is similar to [15]. Fig. 8 shows the received video quality in terms of peak signal-to-noise ratio (PSNR) under various energy constraints for both the "Foreman" and "Coastguard" sequences. From this

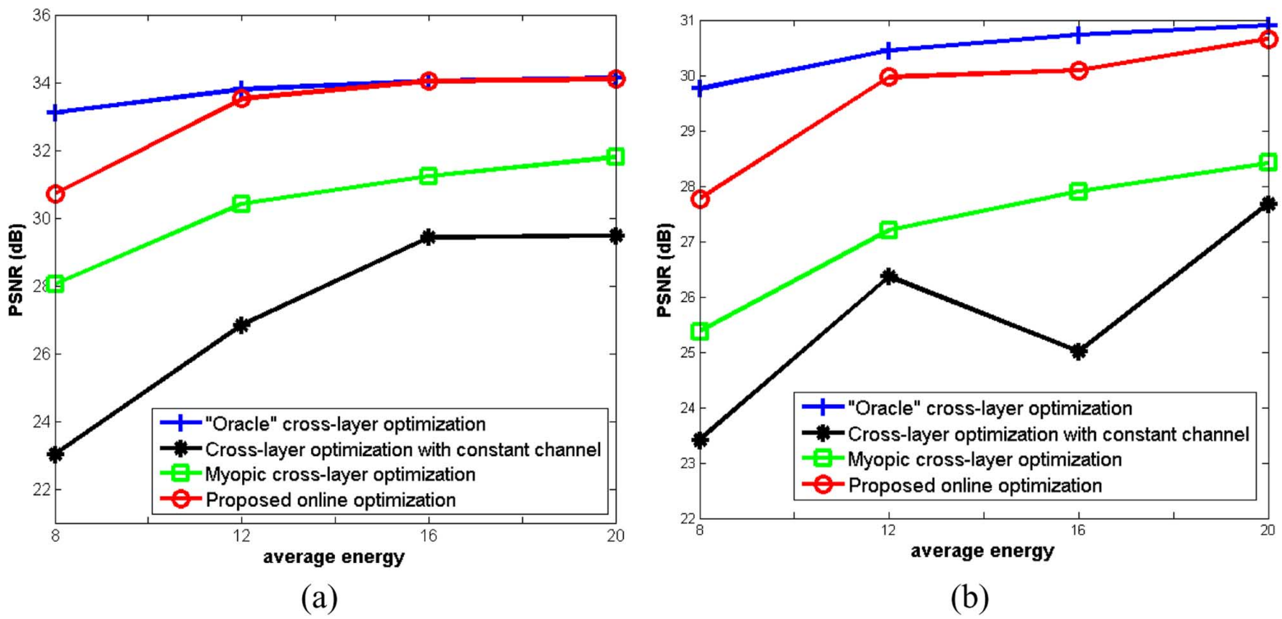

Fig. 8. Video quality (PSNR) under various energy constraints for different cross-layer optimization methods for (a) "Foreman." (b) "Coastguard."

figure, we note that our proposed online optimization outperforms the myopic cross-layer optimization on average by around 3 dB for "Foreman" and 2 dB for "Coastguard," and outperforms the cross-layer optimization with constant channel on average by around 5 dB for both "Foreman" and "Coastguard." We further note that, for lower energy budgets (e.g., $W = 8$), our proposed online optimization achieves around 2 dB lower performance than the "oracle" cross-layer optimization. However, as the budget is increased, our proposed solution can achieve similar video quality (less than 0.5 dB)[10] as indicated in Section E.

Fig. 9 further depicts how the received video quality in terms of PSNR changes over time for the "Coastguard" sequence with the energy budget $W = 12$. From this figure, we note that our proposed online cross-layer optimization can improve the video quality over time through the learning procedure. The achieved video quality in our solution is much smoother (i.e., the PSNRs of all the frames do not vary dramatically) compared to the myopic case and the cross-layer optimization given constant channel conditions, thereby improving the visual experience of the user. Interestingly, we note that our proposed online optimization achieves a higher PSNR than the "oracle" method for the frames indexed from 250–260. This is because the "oracle" method performs the cross-layer optimization for every $M = 8$ DUs [corresponding to one group of pictures (GOP)] without considering the mutual impact among different GOPs. This impact is due to the fact that all the DUs share the same energy constraint. However, our proposed online optimization systematically learns the impact of the current cross-layer action on all the future DUs through the state value function $V(s)$. In other words, our proposed approach optimizes the current cross-layer action as in (20) by considering the impact on not only the DUs in the same GOP but also the DUs from future GOPs.

[10]Note that it is well known that performance improvement less than 0.5 dB is often invisible. However, 1 dB performance improvement is visible for any observer and 2 dB or more results in significantly visible performance improvements.

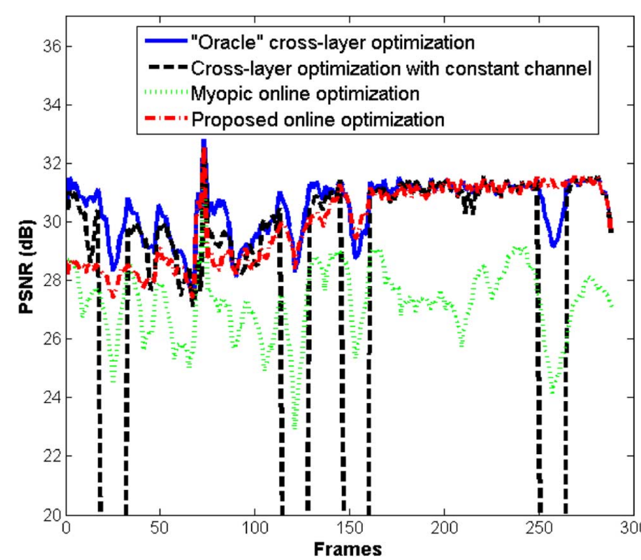

Fig. 9. PSNR for the video sequence "Coastguard" under four cross-layer optimization methods.

VI. Conclusion

In this paper, we consider the problem of cross-layer optimization for delay-sensitive applications, and we develop decomposition principles that guarantee the optimal performance of the application while requiring the necessary message exchanges between neighboring DUs. To account for the unknown and dynamic characteristics of real-time delay-sensitive applications, we further propose an efficient online cross-layer optimization with low complexity, which can be used for live events (e.g., real-time encoding and streaming of ongoing events, video conferencing, etc.), when the encoding is done in real-time and the wireless user does not have *a priori* information about future application data and channel conditions.

REFERENCES

[1] M. van der Schaar and S. Shankar, "Cross-layer wireless multimedia transmission: Challenges, principles, and new paradigms," *IEEE Wireless Commun. Mag.*, vol. 12, no. 4, Aug. 2005.

[2] V. Kawadia and P. R. Kumar, "A cautionary perspective on cross-layer design," *IEEE Wireless Commun.*, vol. 12, no. 1, pp. 3–11, Feb. 2005.

[3] R. Berry and R. G. Gallager, "Communications over fading channels with delay constraints," *IEEE Trans. Inf. Theory*, vol. 48, no. 5, pp. 1135–1149, May 2002.

[4] Q. Liu, S. Zhou, and G. B. Giannakis, "Cross-layer combing of adaptive modulation and coding with truncated ARQ over wireless links," *IEEE Trans. Wireless Commun.*, vol. 4, no. 3, May 2005.

[5] M. Goyal, A. Kumar, and V. Sharma, "Optimal cross-layer scheduling of transmissions over a fading multiacess channel," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3518–3536, Aug. 2008.

[6] A. Fu, E. Modiano, and J. N. Tsitsiklis, "Optimal transmission scheduling over a fading channel with energy and deadline constraints," *IEEE Trans. Wireless Commun.*, vol. 5, pp. 630–641, Mar. 2006.

[7] T. Holliday, A. Goldsmith, and P. Glynn, "Optimal power control and source-channel coding for delay constrained traffic over wireless channels," in *Proc. IEEE Int. Conf. Commun.*, May 2002, vol. 2, pp. 831–835.

[8] W. Chen, M. J. Neely, and U. Mitra, "Energy-efficient transmission with individual packet delay constraints," *IEEE Trans. Inf. Theory*, vol. 54, pp. 2090–2109, May 2008.

[9] W. Chen, U. Mitra, and M. J. Neely, "Energy-efficient scheduling with individual packet delay constraints over a fading channel," *Wireless Netw.*, vol. 15, no. 5, pp. 601–618, Jul. 2009.

[10] E. Uysal-Biyikoglu, B. Prabhakar, and A. El Gamal, "Energy-efficient packet transmission over a wireless link," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 487–499, Aug. 2002.

[11] M. Zafer and E. Modiano, "Delay-constrained energy efficient data transmission over a wireless fading channel," in *Inf. Theory and Appl. Workshop*, Feb. 2007.

[12] A. Faridi and A. Ephremides, "Distortion control for delay-sensitive sources," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3399–3411, Aug. 2008.

[13] A. Banihashemi and A. Hatam, "A distortion optimal rate allocation algorithm for transmission of embedded bitstreams over noisy channels," *IEEE Trans. Commun.*, vol. 56, no. 10, pp. 1581–1584, Oct. 2008.

[14] M. van der Schaar and D. Turaga, "Cross-layer packetization and retransmission strategies for delay-sensitive wireless multimedia transmission," *IEEE Trans. Multimedia*, vol. 9, pp. 185–197, Jan. 2007.

[15] P. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, vol. 8, pp. 390–404, 2005.

[16] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, 1999.

[17] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Belmont, MA: Athena Scientific, 2005.

[18] M. Dai, D. Loguinov, and H. Radha, "Rate distortion modeling for scalable video coders," in *Proc. ICIP*, 2004.

[19] J. Tsitsiklis and B. Van Roy, "An analysis of temporal difference learning with function approximation," *IEEE Trans. Autom. Control*, vol. 42, May 1997.

[20] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23–50, 1998.

[21] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[22] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*. New York: Springer-Verlag, 1997.

[23] D. S. Turaga and T. Chen, "Hierarchical modeling of variable bit rate video sources," *Packet Video*, 2001.

[24] , M. van der Schaar and P. Chou, Eds., *Multimedia Over IP and Wireless Networks: Compression, Networking, and Systems*. New York: Academic, 2007.

[25] J. R. Ohm, "Three-dimensional subband coding with motion compensation," *IEEE Trans. Image Process.*, vol. 3, Sep. 1994.

[26] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 560–576, Jul. 2003.

[27] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[28] Q. Zhang and Y.-Q. Zhang, "Cross-layer design for QoS support in multi-hop wireless networks," *Proc. IEEE (Invited)*, vol. 96, no. 1, pp. 64–76, Jan. 2008.

[29] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[30] E. Altman, *Constrained Markov Decision Processes*. New York: Chapman and Hall/CRC, 1999.

[31] D. Djonin and V. Krishnamurthy, "Transmission control in fading channels—A constrained Markov decision process formulation with monotone randomized policies," *IEEE Trans. Signal Process.*, vol. 55, no. 10, pp. 5069–5083, Oct. 2007.

[32] V. B. Tadic, A. Doucet, and S. Singh, "Two time-scale stochastic approximation for constrained stochastic optimization and constrained Markov decision problems," in *Proc. Amer. Control Conf.*, Jun. 2003, vol. 6, pp. 4736–4741.

Fangwen Fu (S'08) received the Bachelor's and Master's degrees from Tsinghua University, Beijing, China, in 2002 and 2005, respectively.

He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, University of California, Los Angeles. During summer 2006, he was an Intern with the IBM T. J. Watson Research Center, Yorktown Heights, NY. During summer 2009, he was an intern with DOCOMO USA Labs, Palo Alto, CA. His research interests include wireless multimedia streaming, resource management for networks and systems, stochastic optimization, applied game theory, video processing, and analysis.

Mr. Fu was selected by IBM Research as one of the 12 top Ph.D. students to participate in the 2008 Watson Emerging Leaders in Multimedia Workshop in 2008. He received the Dimitris Chorafas Foundation Award in 2009.

Mihaela van der Schaar (F'09) received the Ph.D. degree from Eindhoven University of Technology, The Netherlands, in 2001.

She is currently an Associate Professor with the Department of Electrical Engineering, University of California, Los Angeles. Since 1999, she has been an active participant in the ISO MPEG standard, to which she made more than 50 contributions. She is an Editor (with P. Chou) of *Multimedia over IP and Wireless Networks: Compression, Networking, and Systems* (New York: Academic, 2007). She has received 30 U.S. patents.

Prof. van der Schaar received the National Science Foundation CAREER Award in 2004, the IBM Faculty Award in 2005, 2007, and 2008, the Okawa Foundation Award in 2006, the Best IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY Paper Award in 2005, the Most Cited Paper Award from the EURASIP *Journal Signal Processing: Image Communications* from 2004 to 2006, and three ISO Recognition Awards. She was on the editorial board of several IEEE journals and magazines.