Channel, Deadline, and Distortion (CD^2) Aware Scheduling for Video Streams Over Wireless

Aditya Dua, Carri W. Chan, and Nicholas Bambos
Department of Electrical Engineering
Stanford University
350 Serra Mall, Stanford CA 94305
Email: {dua,cwchan,bambos}@stanford.edu
John Apostolopoulos
Streaming Media Systems Group, HP Labs
Palo Alto, CA
Email: john_apostolopoulos@hp.com

***Abstract*— We study scheduling of multimedia traffic on the downlink of a wireless communication system. We examine a scenario where multimedia packets are associated with strict deadlines and are equivalent to lost packets if they arrive after their associated deadlines. Lost packets result in degradation of playout quality at the receiver, which is quantified in terms of the "distortion cost" associated with each packet. Our goal is to design a scheduler which minimizes the aggregate distortion cost over all receivers. We study the scheduling problem in a dynamic programming (DP) framework. Under well justified modeling reductions, we extensively characterize structural properties of the optimal control associated with the DP problem. We leverage these properties to design a low-complexity Channel, Deadline, and Distortion (CD^2) aware heuristic scheduling policy amenable to implementation in real wireless systems. We evaluate the performance of CD^2 via trace-driven simulations using H.264/MPEG-4 AVC coded video. Our experimental results show that CD^2 comfortably outperforms benchmark schedulers like earliest deadline first (EDF) and best channel first (BCF). CD^2 achieves these performance gains by using the knowledge of packet deadlines, wireless channel conditions, and application specific information (per-packet distortion costs) in a systematic and unified way for multimedia scheduling.**

***Index Terms*— Wireless networks, video streaming, packet scheduling, dynamic programming.**

I. Introduction

The advent of the third generation (3G) of cellular wireless communication systems has sparked an ever increasing interest in mobile wireless multimedia applications like video streaming. Transmission of multimedia traffic over wireless links poses challenging theoretical as well as practical problems. This is attributed to temporal and spatial variations in wireless channel quality, stringent availability of resources like bandwidth, and unique characteristics of multimedia traffic such as packet interdependencies and deadline constraints.

Scheduling algorithms employed at the base station (BS) or access point (AP) play a key role in determining the performance of wireless systems. The problem of downlink scheduling, wherein a single transmitter at the BS is shared amongst multiple downlink users, has been studied extensively (see [1] for an overview). Most initial work in downlink scheduling focused on maximizing throughput and optimizing system performance for non-real-time delay tolerant traffic. The unifying thread for all this work was the idea of *opportunistic scheduling* (see [2], [3] and references therein), which entails exploiting *multiuser diversity* inherent in wireless systems due to fluctuating channel conditions. However, such schedulers, being oblivious to packet deadlines, perform poorly in the context of delay-sensitive multimedia applications.

A. Related work

More recently, the idea of deadline-aware packet scheduling has received attention in the wireless community. Georgiadis et al. [4] showed the optimality of the earliest deadline first (EDF) scheduling policy for deadline constrained scheduling over wired (error-free) channels. However, EDF is not well-suited to the wireless scenario, owing to its disregard for channel variations. In [5], Shakkottai and Srikant modeled the wireless channel as a two-state ON-OFF Markov chain, and showed that using EDF for ON users in each time slot is "nearly optimal" for minimizing the number of packets dropped due to missed deadlines. They were the first to study a channel-aware version of EDF. Khattab and Elsayed [6] proposed a heuristic channel dependent EDF policy, and demonstrated its performance gains via simulations.

In [7], Ren et al. used dynamic programming (DP) for a simulation based study of scheduling constant bit-rate (CBR) traffic over wireless channels modeled as finite-state Markov chains. Johnsson and Cox [8] proposed a heuristic cost function, and showed via simulations that a policy which minimizes this cost function performs well with respect to the number of missed packet deadlines. Dua and Bambos [9] studied deadline and channel aware scheduling in a DP framework. They leveraged provable structural properties of the optimal solution to the DP problem to design low-complexity, near-optimal scheduling policies. Our current work is similar to [9] in spirit.

Even though schedulers proposed in the work cited above account for both channel conditions and packet deadlines,

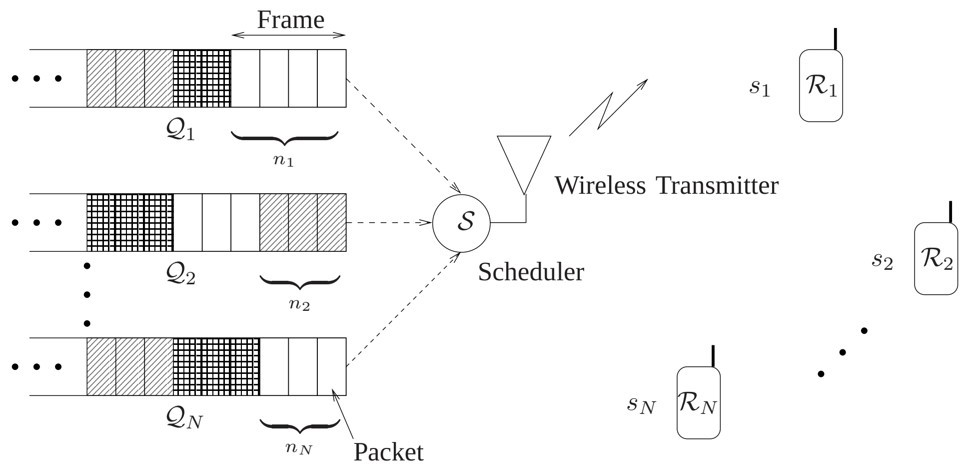

Fig. 1. Schematic of the wireless downlink with N parallel queues and a single time-multiplexed scheduler $\mathcal{S}$ at the base-station.

none of them take into consideration the unique characteristics of multimedia traffic. Amongst the several authors who have explicitly accounted for characteristics of multimedia traffic, Chou and Miao [10] studied *Rate-Distortion* (RD) optimized streaming of packetized media. In their work, the "importance" of every packet is determined by its associated distortion value, and packet (re)-transmissions are scheduled in order to minimize distortion, given the rate constraint of the channel. Wee et al. [11] focused on networks with large delay variations, and achieved improvements in video playout quality by maximizing the probability of on-time delivery of more important packets. Liebl et al. [12] proposed a heuristic cost function which incorporates deadline, channel, and distortion information. They demonstrated via simulations that a scheduler which minimizes this cost yields considerable performance gains over benchmark schedulers. Apostolopoulos [13] examined low-complexity RD optimized streaming of multiple encrypted video streams over a shared bandwidth bottleneck. Chakareski and Frossard [14] studied RD optimized streaming of multiple video streams by prioritizing re-transmissions based on packet contents. They expressed their optimization problem in a Lagrangian framework and used sub-gradient methods to solve it. Kalman et al. [15] used an expected peak signal-to-noise ratio (PSNR) maximization technique to examine scheduling of multiple transcoded video streams over a shared wireless link. Pahalawatta et al. [16] proposed a cross-layer scheduling algorithm for streaming pre-encoded video over wireless downlink packet access networks to multiple users. In their proposal, user data rates were dynamically varied based on channel quality as well as gradients of utility functions, which were designed as functions of the distortion of the received video.

B. Contributions

Our goal in this work is to design a scheduling algorithm which combines knowledge of multimedia characteristics with deadline and wireless channel information in a systematic way to enhance system performance. We consider video transmission over wireless channels with time-varying reliability. Distortion is incurred at the receiver if a packet misses its playout deadline. Only one user can be scheduled in each time slot. The scheduler must decide which user to schedule and which packet to transmit to the scheduled user in every time slot to minimize aggregate distortion incurred over all users. Under well justified modeling reductions, we formulate the scheduling problem in a dynamic programming (DP) framework [17] and establish key structural properties of the optimal control. Prominent amongst these are the optimality of a *switch-over* policy [18], the *time-invariance* of switch-over curves for a two user problem, and the optimality of a *pairwise comparison* approach for a problem with more than two users. We leverage these properties to propose the CD^2 scheduling algorithm, which is amenable to implementation in real systems. We also demonstrate the performance gains achieved by CD^2 over benchmark schedulers via trace-driven simulations.

To the best of our knowledge, our work is the first to study channel, deadline, and distortion aware scheduling of multimedia traffic over a shared wireless link in a mathematical framework.

C. Paper outline

The rest of this paper is organized as follows: We present the system model in Section II, where we discuss the wireless channel model, the distortion cost model, and the optimal packet prioritization policy. In Section III, we formulate the scheduling problem as a DP, under well justified modeling reductions. We then propose our heuristic **C**hannel, **D**eadline, and **D**istortion (CD^2) aware scheduling algorithm, based on a *quasi-static* approach to scheduling. CD^2 has the optimal solution to the DP problem (for the reduced model) at its core. In Section IV, we establish key structural properties of the optimal control for the DP, which are leveraged to develop a low complexity implementation of CD^2. In Section V, we employ trace-driven simulations (using H.264/MPEG-4 AVC coded video) to demonstrate the efficacy of CD^2 and its performance gains (2-12 dB increase in average PSNR) relative to benchmark schedulers like "earliest deadline first" and "best channel first". We furnish concluding remarks in Section VI.

II. Model Construction

We study a time slotted wireless system with N downlink users and a time-multiplexed scheduler $\mathcal{S}$ at the base station (BS). There is a queue corresponding to each downlink user at the BS, which buffers video frames the user wishes to receive. The queue for the i^{th} user is denoted $\mathcal{Q}_i$. A schematic of the system is depicted in Fig. 1. Each video frame is divided into multiple network packets. The video is encoded to achieve a roughly constant quality for each frame, which leads to a variable number of network packets per frame, depending on the difficulty in compressing each frame. The video quality is measured in terms of peak signal-to-noise ratio (PSNR), defined as PSNR$\triangleq 10\log_{10}(255^2/\text{Distortion})$. The distortion is measured in terms of mean-squared error.

In each time slot, $\mathcal{S}$ schedules one packet from the head-of-line (HOL) frame of one of the N queues for transmission according to some scheduling policy. The HOL frame of $\mathcal{Q}_i$ comprises of N_i packets and is associated with a deadline D_i. This deadline reflects the time by which the frame must be received at the downlink receiver to ensure uninterrupted playout. All packets in the HOL frame share this common deadline. Any packets in the frame which are not successfully transmitted before the expiration of the deadline are dropped. This results in a degradation of video quality at the receiver due to increased distortion. The objective of the scheduler is to minimize aggregate distortion at the downlink receivers due to missed packet deadlines.

A. Wireless Channel Model

Wireless channels exhibit temporal and spatial fluctuations, which are attributed to user mobility, interference from concurrent transmissions, and signal attenuation due to physical phenomena. Different models have been used in the literature to abstract this behavior of wireless channels. While some authors model the wireless channel as a reliable "bit-pipe" with time-varying capacity, others model it as a fixed-size "bit-pipe" with time-varying reliability. We adopt the latter approach and quantify the channel quality in time slot t by the probability of successful transmission of a packet over the channel, if the channel is used in time slot t. For $\mathcal{Q}_i$, we denote this success probability by s_i^t. For example, s_i^t can be modeled as being modulated by an underlying finite-state Markov chain (FSMC) [19], where each state corresponds to a different probability of successful transmission. We employ a two-state FSMC model for performance evaluation in Section V. We further assume that the wireless channels from the BS to the downlink users are independent of each other.

B. Distortion Costs

If one or more packets in a frame miss their deadline, the decoder is forced to use error concealment techniques such as "previous frame copy", and a distortion cost is incurred. We measure distortion in the mean-squared error (MSE) sense. The distortion cost associated with each packet, which expresses the packet's application layer importance, is placed in the packet header and thereby is accessible to the scheduler [13].

We denote by $\omega_i(k_i)$ the distortion cost incurred if k_i packets from the HOL frame of $\mathcal{Q}_i$ miss their deadline. We assume that $\omega_i(k_i)$ is a non-negative, strictly increasing, and convex function of $k_i\ \forall\ i$. While the first two assumptions are consistent with intuition, the convexity assumption is corroborated by empirical data. Fig. 2 depicts plots of $\omega_i(k_i)$ for four different frames of the "Foreman sequence" (a test sequence widely used by the video community) in CIF format encoded using H.264/MPEG-4 AVC with a single leading I-frame followed by 299 P-frames. Observe that the empirical results are in accordance with our assumptions.

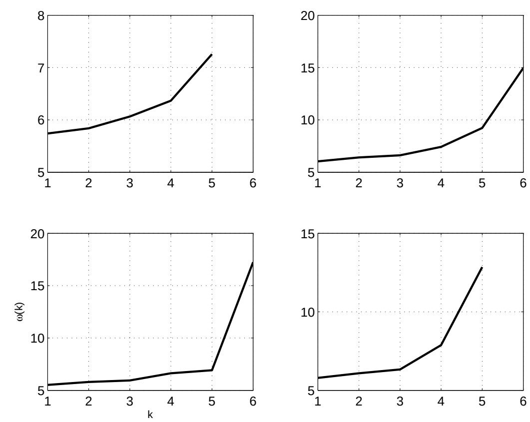

Fig. 2. Plots of $\omega_i(k_i)$ for four different frames of the Foreman sequence

We compute $\omega_i(k_i)$ by dropping packets, decoding, and computing the resultant MSE. We assume that distortion is additive across multiple dropped packets. There are 2^K possible ways (dropping patterns) of dropping packets from a frame comprised of K packets. $\omega_i(1)$ is defined as the minimum MSE distortion incurred by dropping only one packet from the HOL frame of $\mathcal{Q}_i$. $\omega_i(k_i+1)$ is defined as the minimum MSE distortion incurred by dropping one more packet, in addition to the packets dropped to incur $\omega_i(k_i)$. This embedded computation of $\omega_i(k_i)$ imposes a rank ordering on the packets within a frame, which might be quite different from the ordering imposed by encoding. These computations are performed offline at the time of encoding, and the results are placed in the packet/frame headers, so that they are readily available to a distortion-aware scheduler. The rank of a packet in a frame expresses its priority for transmission by a distortion-aware scheduler. The number of dropping patterns now reduces to $K+1$ from 2^K.

III. Problem Formulation

In general, the scheduling problem can be formulated within a control/optimization framework and solved using numerical techniques, given statistical characterizations or actual realizations of the distortion cost curves and channel conditions for all users. Such an approach, apart from being computationally prohibitive from an implementation perspective, does not provide insight into the fundamental trade-offs inherent in the scheduling problem. Moreover, detailed knowledge of traffic or channel conditions is not available to the scheduler in real wireless systems. We seek a formulation which encapsulates the fundamental scheduling tradeoffs, is amenable to analysis, and leads to implementation friendly scheduling policies. To this end, we (sequentially) introduce two modeling *reductions*.

- *Modeling Reduction R1*: To formulate our optimal control problem, we assume that each queue contains only one frame.
- *Modeling Reduction R2*: We assume static (in a probabilistic sense) channel conditions in the control problem formulated under reduction R1.

In Section III-A, we will study an optimal control problem which incorporates reduction R1 *only*. Then, in Section III-B, we will study a control problem which incorporates *both* reductions.

A. Single frame optimal — problem formulation under R1 only

R1 is a reasonable assumption if the frames are being generated periodically by a real-time media source, so that a new frame arrives to a queue only after the current frame has been transmitted. This assumption is also consistent with the principles of low-latency media system design. We consider a finite time-horizon of T time slots starting at $t = 1$, where $T = \max(D_1, \ldots, D_N)$ and D_i is the deadline associated with the HOL frame of $\mathcal{Q}_i$. The deadlines on all HOL frames expire by the end of the time-horizon T. Any residual packets at the end of the horizon are dropped and a cost is incurred, as described in Section II-B.

Our objective is to design a scheduling policy which minimizes the sum, over all users, of expected dropping costs at the end of time-horizon T. The scheduler is assumed to know the channel statistics in terms of success probabilities for all users in time slots $t = 1, \ldots, T$. We call this problem the *single frame optimal* and the associated optimal control policy $\mathcal{P}^\star(N)$.

We adopt the methodology of dynamic programming (DP) to compute $\mathcal{P}^\star(N)$. Let $\mathbf{n} = (n_1, \ldots, n_N)$ denote the *state* of the system (all vectors are denoted in **boldface**), where $n_i \leq N_i$ is the number of remaining packets in the HOL frame of $\mathcal{Q}_i$ at the beginning of the current time slot. If $\mathcal{Q}_i$ is scheduled in time slot t, the state in time slot $t+1$ changes to $\mathbf{n} - \mathbf{e}_i$ with probability (w.p.) s_i^t (transmission successful) and $\mathbf{n}$ w.p. $\bar{s_i}^t \triangleq 1 - s_i^t$ (transmission fails). Here $\mathbf{e}_i$ is the standard i^{th} unit vector in $\mathbb{R}^N$, that is, $\mathbf{e}_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ with the 1 in the i^{th} location. Without loss of generality, we assume $0 < D_1 \leq \ldots \leq D_N = T$. The assumption implies that $\mathcal{Q}_i$ is not a scheduling candidate after $t = D_i \leq T$, since the deadline on its HOL frame expires in time slot $t = D_i$.

Denote by $V^t(\mathbf{n})$ the *expected cost-to-go* in time slot t, starting in state $\mathbf{n}$. By definition, $V^t(\mathbf{n})$ is the minimum expected cost incurred by the optimal policy $\mathcal{P}^\star(N)$ over time slots $t, \ldots, T$, starting in state $\mathbf{n}$ in time slot t. $V^t(\mathbf{n})$ is computed from the following recursive DP equations:

$$V^t(\mathbf{n}) = \min_{i=1,\ldots,N} \left\{\alpha_i^t(\mathbf{n})\right\} + V^{t+1}(\mathbf{n}), \quad t = 1, \ldots, T, \quad (1)$$

and the boundary conditions $V^{T+1}(\mathbf{n}) = \sum_{i=1}^{N} \omega_i(n_i)$, where

$$\alpha_i^t(\mathbf{n}) = \begin{cases} s_i^t[V^{t+1}(\mathbf{n} - \mathbf{e}_i) - V^{t+1}(\mathbf{n})], & n_i > 0, t \leq D_i \\ 0, & \text{else.} \end{cases} \quad (2)$$

Setting $\alpha_i^t(\mathbf{n}) = 0$ when $n_i = 0$ or $t > D_i$ eliminates $\mathcal{Q}_i$ from consideration as a scheduling candidate when $\mathcal{Q}_i$ is empty, or the deadline on its HOL frame has expired.

Solving the DP equations to compute $\mathcal{P}^\star(N)$ requires non-causal knowledge of channel conditions over a period of T time slots, which is unavailable to the scheduler in real wireless systems. Thus, the single frame offline optimal problem (based on R1 alone) does not immediately lead to implementable scheduling policies.

B. A quasi-static approach to scheduling — CD^2

Motivated by the foregoing discussion, we now introduce reduction R2 into the our formulation, in addition to R1. Under R2, the probability of successful transmission is fixed for each user over a horizon of T time slots. Equivalently $s_i^t = s_i$ for $t = 1, \ldots, T$, $\forall\, i$. This is a reasonable assumption for slowly varying channels. We denote the optimal control in this case by $\mathcal{P}_s^\star(N)$, which is computed via (1) and (2), with s_i^t replaced by s_i. Thus, $\mathcal{P}_s^\star(N)$ is a special case of $\mathcal{P}^\star(N)$ where the success probability for each user is constant over the horizon T of interest.

How does $\mathcal{P}_s^\star(N)$ translate into an implementable scheduling policy? To answer this question, we propose a *quasi-static* approach to scheduling. We name our proposed policy CD^2, since it is a **C**hannel, **D**eadline, and **D**istortion aware scheduling policy. The key steps in CD^2 are:

1) Given a system characterization in terms of instantaneous channel conditions, HOL frame deadlines, and number of packets in the HOL frame of each queue in the current time slot, compute $\mathcal{P}_s^\star(N)$ by solving (1) and (2) under the assumptions imposed by R1 and R2.
2) Schedule a packet in the system based on the decision of $\mathcal{P}_s^\star(N)$ computed in Step 1.
3) Update the system parameters based on the outcome of Step 2 and most recently acquired channel knowledge (through receiver feedback or measurements made by the BS).
4) Repeat steps 1-3 in every time slot.

Thus, the scheduling decision of CD^2 in each time slot is based on the static channel assumption (R2). However, the static operating point is updated in each time slot as wireless channels evolve over time. This justifies the nomenclature quasi-static. Note that CD^2 requires only instantaneous channel knowledge, rather than non-causal channel knowledge or a detailed statistical characterization of the channel behavior.

We reiterate that CD^2 is based on $\mathcal{P}_s^\star(N)$, which is the optimal control policy for the scheduling problem formulated under modeling reductions R1 and R2. The gains provided by CD^2 relative to benchmark policies (Section V) suggests that the reduced model was a reasonable one to study from the perspective of efficient scheduler design.

IV. Structural Properties of $\mathcal{P}_s^\star(N)$

In this section, we present important structural properties of $\mathcal{P}_s^\star(N)$, which is at the core of CD^2. We initially focus on a two user scenario ($N=2$), and show in Section IV-B that $\mathcal{P}_s^\star(N)$ for $N>2$ can be computed by using $\mathcal{P}_s^\star(2)$ multiple times in a pairwise fashion.

A. Key properties of $\mathcal{P}_s^\star(2)$

To make the static channel assumption (R2) explicit, we suppress the superscript t from the successful transmission probabilities and simply denote them by s_1 and s_2. Once again, we assume without any loss of generality that $D_1 \le D_2$. Thus, $\mathcal{Q}_1$ is a scheduling candidate for $t=1,\dots,D_1$ provided $n_1>0$, while $\mathcal{Q}_2$ is a scheduling candidate over the entire time-horizon provided $n_2>0$. We employ the notation $\mathbf{n}=(n_1,n_2)$, $\mathbf{e}_1=(1,0)$, and $\mathbf{e}_2=(0,1)$. Since no scheduling decision needs to be made after $t=D_1$ ($\mathcal{Q}_2$ is scheduled, if non-empty), we reformulate our control problem for a time-horizon of length $T=D_1$ (instead of D_2, as in Section II). The DP equations can be re-written as

$$V^t(\mathbf{n}) = \min\left\{\alpha_1^t(\mathbf{n}), \alpha_2^t(\mathbf{n})\right\} + V^{t+1}(\mathbf{n}), \quad t=1,\dots,T, \tag{3}$$

along with the boundary conditions $V^{T+1}(\mathbf{n}) = \omega_1(n_1) + \phi_2(n_2, D_2-D_1)$, where

$$\alpha_i^t(\mathbf{n}) = \begin{cases} s_i[V^{t+1}(\mathbf{n}-\mathbf{e}_i) - V^{t+1}(\mathbf{n})], & n_i>0 \\ 0, & \text{else}, \end{cases} \tag{4}$$

and $\phi_2(\cdot,\cdot)$ is computed via the recursion

$$\phi_2(y,t) = \begin{cases} 0, & y=0 \\ s_2\phi_2(y-1,t-1) + \bar{s}_2\phi_2(y,t-1), & y,t>0 \\ \omega_2(y), & t=0. \end{cases} \tag{5}$$

Here, $\omega_1(n_1)$ is the distortion cost associated with dropping n_1 packets from $\mathcal{Q}_1$ at the end of the time-horizon, while $\phi_2(n_2, D_2-D_1)$ is the expected distortion cost incurred in transmitting n_2 packets from $\mathcal{Q}_2$ over a static channel with success probability s_2 during time slots $T+1,\dots,D_2$.

Lemma 1: * $\phi_2(y,t) = \displaystyle\sum_{j=0}^{\min\{y,t\}} \binom{t}{j} \omega_2(y-j) s_2^j (1-s_2)^{t-j}$.

Lemma 2: $\phi_2(y,t)$ is a non-decreasing and convex function of y for fixed t.

Now, define the *decision function* $\gamma^t(\mathbf{n})$ by

$$\gamma^t(\mathbf{n}) \triangleq \alpha_1^t(\mathbf{n}) - \alpha_2^t(\mathbf{n}), \quad t=1,\dots,T. \tag{6}$$

Clearly, $\mathcal{P}_s^\star(2)$ schedules $\mathcal{Q}_1$ in state $\mathbf{n}$ in time slot t if $\gamma^t(\mathbf{n}) \le 0$, and schedules $\mathcal{Q}_2$ else. Thus, $\mathcal{P}_s^\star(2)$ is completely determined by the *sign* of $\gamma^t(\mathbf{n})$. We now state a key property of $\gamma^t(\mathbf{n})$.

Lemma 3: $\gamma^t(\mathbf{n})$ is a non-increasing function of n_1 and a non-decreasing function of n_2.

An immediate and important consequence of Lemma 3 is the optimality of a *switch-over* type policy in each time slot. We first formally define a switch-over type policy.

*Proofs of all lemmas/theorems are available in the Appendix.

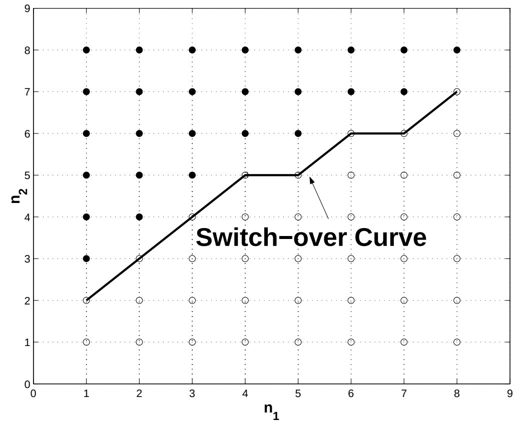

Fig. 3. A typical switch-over curve; ○ and ● denote the states in which it is optimal to schedule $\mathcal{Q}_1$ and $\mathcal{Q}_2$, respectively.

Definition: A scheduling policy is of switch-over type if in every time slot t, the policy can be characterized by a non-decreasing switch-over curve $\psi^t : \mathbb{N} \mapsto \mathbb{N}\cup\{0\}$, such that the policy schedules $\mathcal{Q}_2$ in time slot t if $n_2 > \psi^t(n_1)$, and schedules $\mathcal{Q}_1$ else (see Fig. 3).

Theorem 1 (Optimality of Switch-over Policy): The policy $\mathcal{P}_s^\star(2)$ is of switch-over type.

The scheduling decision of $\mathcal{P}_s^\star(2)$ in the current time slot ($t=1$) is determined by ψ^1. Since our problem (3)-(5) is formulated as a *backward recursion*, one expects that $\psi^T,\dots,\psi^2$ must be computed prior to computing ψ^1. Interestingly, our next result shows that this is not the case.

Theorem 2 (Time-invariance): The switch-over curves ψ^t which characterize $\mathcal{P}_s^\star(2)$ are time-invariant, that is, $\psi^t = \psi, \forall\, t=1,\dots,T$.

Since the switch-over curves are time-invariant, computing the desired switch-over curve ψ^1 is equivalent to computing ψ^T. However, ψ^T is determined by the sign of $\gamma^T(\mathbf{n})$, which was computed as a function of $\omega_1(\cdot)$ and $\phi_2(\cdot)$ in the proof of Lemma 3 (see Section VII-C). We reproduce the expression here for convenience:

$$\begin{aligned}\gamma^T(\mathbf{n}) = &-s_1[\omega_1(n_1) - \omega_1(n_1-1)] \\ &+ s_2[\phi_2(n_2, D_2-D_1) - \phi_2(n_2-1, D_2-D_1)].\end{aligned} \tag{7}$$

Also, recall that $\phi_2(\cdot,\cdot)$ was computed as a function of $\omega_2(\cdot)$ in Lemma 1. In summary, ψ^T, and hence ψ^1 can be explicitly computed in terms of the distortion cost functions $\omega_1(\cdot)$ and $\omega_2(\cdot)$, which are available to the scheduler from the packet headers. The implication is that we have the optimal two user policy for the scheduling problem formulated under reductions R1 and R2 in closed form. Note that the foregoing analysis is valid under the assumption $D_2 \ge D_1$. Analogous results for the case $D_1 \ge D_2$ are gotten by interchanging the roles of $\mathcal{Q}_1$ and $\mathcal{Q}_2$.

An alternate interpretation of Theorem 1 and 2 is as follows: For fixed n_1 and n_2, $\mathcal{P}_s^\star(2)$ is characterized by a switchover curve on the (D_1, D_2) plane, which is a straight line with slope $\pi/4$. This straight line is offset from the origin, and the offset is a function of $s_1, s_2, n_1, n_2, \omega_1(\cdot)$, and $\omega_2(\cdot)$, but not of t. If the line passed through the origin, $\mathcal{P}_s^\star(2)$ would reduce

Repeat:
- If $\mathbf{Q}^t = \emptyset^\dagger$, *quit*.
- If $\mathbf{Q}^t = \{k\}$, schedule $\mathcal{Q}_k$ and *quit*.
- Set $\mathcal{U} = \emptyset$ and $\mathfrak{Q} = \mathbf{Q}^t$. *Repeat*:
  - If $\mathfrak{Q} = \emptyset$, *quit*.
  - If $\mathfrak{Q} = \{k\}$, set $\mathcal{U} = \mathcal{U} \cup \{k\}$, $\mathfrak{Q} = \emptyset$.
  - If $|\mathfrak{Q}| \geq 2$, select $k, l,\ k \neq l$ randomly from $\mathfrak{Q}$.
    - Use $\mathcal{P}_s^\star(2)$ to choose one of either $\mathcal{Q}_k$ or $\mathcal{Q}_l$.
    - If $\mathcal{Q}_k$ is chosen, set $\mathcal{U} = \mathcal{U} \cup \{k\}$, else set $\mathcal{U} = \mathcal{U} \cup \{l\}$. In both cases, set $\mathfrak{Q} = \mathfrak{Q} \setminus \{k, l\}$.
- Set $\mathbf{Q}^t = \mathcal{U}$.

† $\mathbf{Q}^t$ denotes the set of scheduling candidates (non-empty queues) in time slot t.

TABLE I

IMPLEMENTATION OF STEP 2 OF PAIRWISE CD^2

to earliest deadline first (EDF). However, a non zero offset demonstrates that $\mathcal{P}_s^\star(2)$ accounts for channel conditions and queue state, in addition to deadline information. This explains why CD^2, which has $\mathcal{P}_s^\star(2)$ at its core, outperforms EDF, which makes scheduling decisions based on deadlines alone.

B. Optimality of pairwise comparisons, and CD^2 re-visited

How do the above results generalize to $\mathcal{P}_s^\star(N)$, the optimal control for a system with $N > 2$ users? To answer this question, we define the *pairwise decision functions*:

$$\gamma_{ij}^t(\mathbf{n}) \triangleq \alpha_i^t(\mathbf{n}) - \alpha_j^t(\mathbf{n}), \quad t = 1, \ldots, T. \tag{8}$$

$\mathcal{P}_s^\star(N)$ "prefers" $\mathcal{Q}_i$ over $\mathcal{Q}_j$ in time slot t in state $\mathbf{n}$ if $\gamma_{ij}^t(\mathbf{n}) \leq 0$, and prefers $\mathcal{Q}_j$ else. Now consider another decision rule, namely $\Pi_{\text{PW}}(N)$, which discriminates between $\mathcal{Q}_i$ and $\mathcal{Q}_j$ in time slot t in state $\mathbf{n}$ based on the sign of $\gamma_{ij}^t(\mathbf{n}^{ij})$ instead of the sign of $\gamma_{ij}^t(\mathbf{n})$, where $\mathbf{n}^{ij}$ agrees with $\mathbf{n}$ in the i^{th} and j^{th} locations, and is zero elsewhere. $\Pi_{\text{PW}}(N)$ is therefore a *pairwise comparison* rule which solves the N-user problem as a sequence of two user problems. Clearly, $\mathcal{P}_s^\star(N) = \Pi_{\text{PW}}(N)$ for $N = 2$. Does $\mathcal{P}_s^\star(N) = \Pi_{\text{PW}}(N)\ \forall\ N$? Yes!

Theorem 3: For the scheduling problem formulated under reductions R1 and R2, the pairwise comparison rule $\Pi_{\text{PW}}(N)$ is optimal, that is, $\mathcal{P}_s^\star(N) = \Pi_{\text{PW}}(N)$.

Pairwise CD^2: Recall from Section III-B that CD^2 computes $\mathcal{P}_s^\star(N)$ in each time slot (Step 1) and schedules a packet in the system based on the decision of $\mathcal{P}_s^\star(N)$ (Step 2). Theorem 3 provides an alternative way of implementing Step 2 of CD^2, based on computing $\mathcal{P}_s^\star(N)$ by using the pairwise comparison rule $\Pi_{\text{PW}}(N)$. In Step 2 of pairwise CD^2, users are grouped randomly into pairs. Users within a pair are compared using policy $\mathcal{P}_s^\star(2)$, which is computable in closed form, as shown in Section IV-A. The winner of each pair is promoted to the next round. The process continues till only one user survives. This user is scheduled in the current time slot. Implementation details of Step 2 of pairwise CD^2 are enumerated in Table I. Steps 1,3, and 4 are identical to CD^2.

Pairwise CD^2 based on $\Pi_{\text{PW}}(N)$ requires at most $N-1$ pairwise comparisons to make a scheduling decision and hence has a computational complexity $\mathcal{O}(N)^\dagger$, since the complexity of each pairwise comparison based on $\mathcal{P}_s^\star(2)$ is $\mathcal{O}(1)$ (due to the time-invariance property). In contrast, CD^2 based on solving the DP equations directly has a computational complexity of $\mathcal{O}(n^N D)$ if $n_i = \mathcal{O}(n)$ and $D_i = \mathcal{O}(D)\ \forall\ i$.

V. SIMULATION RESULTS

In this section, we experimentally examine the performance of our proposed CD^2 scheduling policy. We compare CD^2 to the following benchmarks: Round Robin (RR), which schedules users in periodic fashion; Earliest Deadline First (EDF), which schedules the user with the most imminent deadline; and Best Channel First (BCF), which schedules the user with the best instantaneous channel condition. CD^2 jointly accounts for channel conditions, deadlines, and distortion costs in its scheduling decision. We consider two versions of each of the benchmark schedulers — a *basic* version which ignores packet distortion costs and transmits packets within a frame in sequential order, and a *distortion-aware* version which uses distortion information to reorder packets within a frame according to the prioritization rule described in Section II-B. Table II summarizes the decision criteria of all scheduling policies considered here.

We examine a system with four downlink users. Video frames for users arrive periodically to their respective queues at the BS, and get associated with a deadline equal to the period of arrival. A frame is comprised of multiple network packets. Any packets within a frame which are not successfully transmitted before deadline expiration are dropped, resulting in degradation of video quality at the corresponding downlink receiver. The received video quality is characterized by its PSNR (peak signal-to-noise ratio), defined as PSNR$\triangleq$ $10\log_{10}(255^2/\text{Distortion})$. Distortion is measured in terms of mean-squared error. We use average PSNR (averaged over all four users) as a performance metric to compare different schedulers. PSNR is the most widely used metric for quantifying video quality. Typically, a 0.5dB difference in PSNR is

†Let m_N be the number of pairwise comparisons required by $\Pi_{\text{PW}}(N)$. Then, $m_N = N/2 + m_{N/2}$ if N is even and $m_N = (N-1)/2 + m_{(N+1)/2}$ if N is odd. It is now easily verified that $m_N = (N-1)\ \forall\ N$.

Policy	Channel	Deadline	Distortion
Round Robin (w/o reordering)			
Round Robin (w/ reordering)			✓
Earliest Deadline First (w/o reordering)		✓	
Earliest Deadline First (w/ reordering)		✓	✓
Best Channel First (w/o reordering)	✓		
Best Channel First (w/ reordering)	✓		✓
CD^2	✓	✓	✓

TABLE II
DECISION CRITERIA FOR DIFFERENT SCHEDULING POLICES

noticeable, while a 2dB improvement in PSNR translates to significant improvement in perceived video quality.

In our simulation setup, each users wishes to receive 300 frames of the "Foreman sequence" (a commonly used test video sequence) at 352x288 pixels/frame (CIF format), 30 frames/sec, encoded using the new H.264/MPEG-4 AVC video compression standard [20] with a leading I-frame followed by 299 P-frames. All P-frames were chosen in order to produce a homogeneous stream of coded frames, in the sense that the coded frames (and associated packets) were a priori of approximately equal importance. The video was coded using H.264 reference software version JM10.2 [21]. Each coded frame was divided into independently decodable network packets of size 1500 bytes or less. This resulted in three to seven packets per frame, depending on the video content encoded in the frame. For example, a frame which captures a sudden scene change is likely to contain more packets than a frame which encodes a relatively static scene. "Frame copy error" concealment techniques were used to estimate missing information when one or more packets in a frame missed their decoding deadlines. A perfectly received copy of the Foreman sequence corresponds to a PSNR of 40.7dB. This establishes an upper-bound on the performance achievable by *any* scheduler. Note that this upper bound is finite because of the distortion introduced by lossy compression of the original video stream.

We used a two-state Gilbert-Elliot model for simulating bursty downlink wireless channels. The two states, GOOD and BAD, were associated with success probabilities s_G and s_B respectively, with $s_G > s_B$. The probability of transition (in every time slot of duration $\sim$ 1.3ms) from the GOOD to BAD state, as well as from the BAD to GOOD state, was fixed at 0.05. The success probabilities for users 2,3, and 4 were fixed at $s_B = 0.75$ and $s_G = 0.95$, 0.97, and 0.99, respectively. Also, $s_G = 0.9$ was fixed for user 1, while s_B was varied from 0.1 to 0.9, in steps of 0.1. For our choice of parameters, the stationary probability of being in either channel state is 0.5. Thus, the average success probability is computed as $s^{avg} = 0.5(s_G + s_B)$. Under the assumption of additive distortion across multiple packets [13], we simulated 100 channel realizations for each policy and for 9 different success probabilities, for a total of 7200 channel realizations. We contrasted the performance of CD^2 to other benchmark policies over identical channel realizations.

Fig. 4 depicts the average PSNR (averaged over all users) as a function of the average success probability for user 1 (s_1^{avg}), keeping $s_2^{avg} = s_3^{avg} = s_4^{avg}$ fixed. CD^2 comfortably outperforms the basic versions of RR, EDF, and BCF by several dB of PSNR. CD^2 also achieves significant gains of 0.5-2dB over the distortion-aware versions of RR, EDF, and BCF. The improvement is largest over PSNR ranges where viewing is desired — 35dB. As the PSNR falls below 35dB, the perceived video quality falls quickly, and when it falls below roughly 30dB the quality can become unacceptable. Note that CD^2 achieves an average PSNR of 35dB at $s_1^{avg} \approx 0.65$, whereas basic versions of benchmark schedulers do not achieve that performance level even at $s_1^{avg} = 0.9$. There is a significant improvement in the performance of benchmark schedulers when they are allowed to prioritize packet transmissions based on per-packet distortion information. For instance, EDF with and without reordering drop an identical number of packets for each corresponding frame. However, EDF with reordering drops packets which cause the least amount of distortion, leading to 4-5dB gains. The results emphasize the importance of the preprocessing required to compute per-packet distortion information to include in packet headers to enhance system performance.

Fig. 5 shows the performance of the worst-case user for each policy. CD^2 achieves up to 4dB gains over the next best benchmark policy (distortion aware EDF). The gains are greater relative to average PSNR performance because the disparity between all users in the benchmark policies is quite large. However, for CD^2 the variance in PSNR across users is fairly small — the PSNR of the best user drops slightly in order to increase the PSNR of the worst user. Thus, CD^2 has better *fairness* properties than benchmark policies. CD^2 achieves a worst case PSNR of 35dB for $s_1^{avg} \approx 0.73$, while none of the benchmarks (basic or distortion aware) achieve that even for $s_1^{avg} = 0.9$. This clearly demonstrates the superiority of CD^2 under disparate channel conditions, a situation very likely to arise in real wireless systems, where users far from the BS are more likely to experience poor channels.

Fig. 6 depicts the average number of packets dropped under each policy. Interestingly, in some cases CD^2 drops *more* packets than EDF and BCF, but the average PSNR for CD^2 is still significantly higher. This is attributed to the fact that EDF and BCF (both basic and distortion-aware versions) ignore channel conditions and frame deadlines respectively while

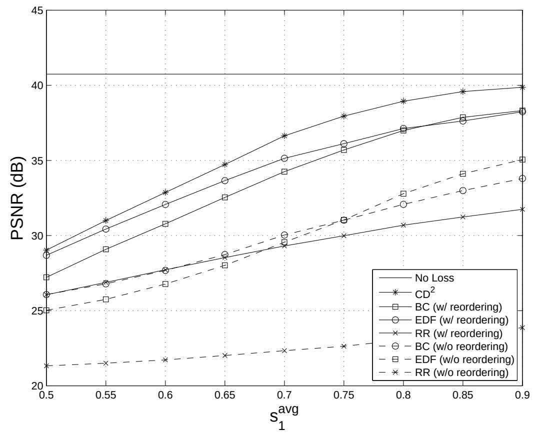

Fig. 4. Performance gains for 4 user case: Average PSNR (in dB) versus average probability of successful transmission for user 1, s_1^{avg}

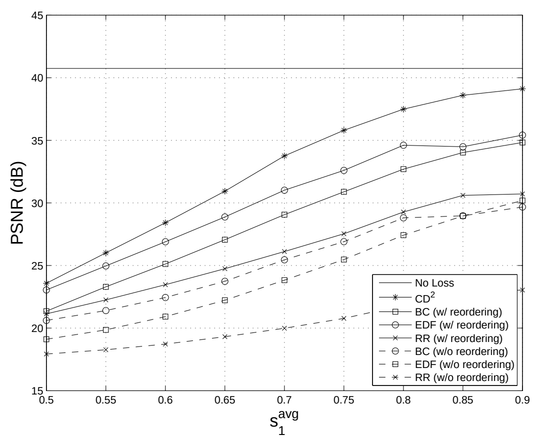

Fig. 5. Performance gains for 4 user case: Worse Case PSNR (in dB) versus average probability of successful transmission for user 1, s_1^{avg}

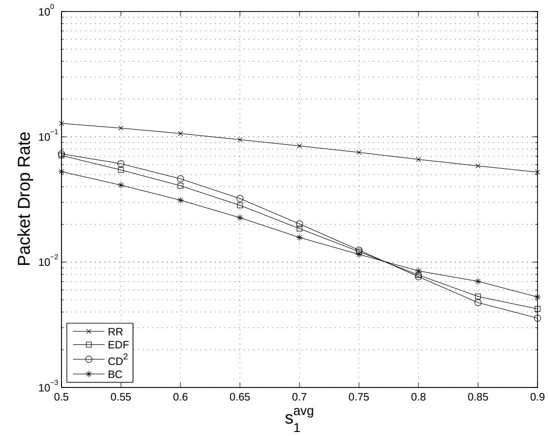

Fig. 6. Packet drop rate versus average probability of successful transmission for user 1, s_1^{avg}

making their scheduling decisions. In contrast, CD^2 jointly utilizes all available information to make more "intelligent" scheduling decisions.

VI. Conclusions

This paper examined the problem of scheduling multiple video streams across a shared wireless channel. We proposed the **C**hannel, **D**eadline, and **D**istortion (CD^2) aware scheduling algorithm to provide a unified and systematic way to enhance system performance. CD^2 determines the best schedule based on channel characteristics, packet delay deadlines, and packet importance, and prioritizes transmission of packets within a stream as well as across multiple streams, in order to minimize the expected aggregate distortion across all of the video streams. Our experimental results show that CD^2 provides significant gains vis-à-vis benchmark schedulers.

VII. Appendix

A. Proof of Lemma 1

Denote $\beta(x, y, s) \triangleq \binom{x}{y} s^y (1-s)^{x-y}$. Two distinct cases arise:

1) $y \geq t$: At least $y-t$ packets get dropped. Additionally, $t-j$ packets get dropped w.p. $\beta(t, j, s_2)$. Thus, a total of $y-j$ packets get dropped w.p. $\beta(t, j, s_2)$, implying $\phi_2(y, t) = \sum_{j=0}^{t} \omega_2(y-j)\beta(t, j, s_2)$.

2) $y \leq t$: Consider each channel use as equivalent to a coin toss with bias s_2. If y or more tosses result in success, no packets are dropped. If $j < y$ tosses result in success, $y-j$ packets get dropped. Thus, $\phi_2(y, t) = \sum_{j=0}^{y} \omega_2(y-j)\beta(t, j, s_2)$.

Combining the two cases, we get the desired result.

B. Proof of Lemma 2

For $y \geq t$, it follows from the proof of Lemma 1 that $\phi_2(y, t) = \sum_{j=0}^{t} \omega_2(y-j)\beta(t, j, s_2)$, which is a non-negative linear combination of non-decreasing and convex functions, and hence inherits the same properties. For $y < t$, we have

$$\phi_2(y+1, t) - \phi(y, t) = \sum_{j=0}^{y} \underbrace{[\omega_2(y-j+1) - \omega_2(y-j)]}_{\geq 0 \text{ from monotonicity of } \omega_2(\cdot)} \underbrace{\beta(t, j, s_2)}_{\geq 0} \geq 0, \quad (9)$$

$$[\phi_2(y+1, t) - \phi_2(y, t)] - [\phi_2(y, t) - \phi_2(y-1, t)] = \sum_{j=0}^{y-1} \underbrace{[\omega_2(y-j+1) - 2\omega_2(y-j) + \omega_2(y-j-1)]}_{\geq 0 \text{ from convexity of } \omega_2(\cdot)} \beta(t, j, s_2) + \underbrace{\omega_2(1)\beta(t, y, s_2)}_{\geq 0} \geq 0. \quad (10)$$

Monotonicity of $\phi(y, t)$ as a function of y now follows directly from (9) and convexity follows from (10) and a basic property of convex functions, viz., a non-decreasing derivative [22].

C. Proof of Lemma 3

The proof is based on inductive arguments.

1) Base Case ($t = T$): From (4), (6), and the boundary conditions for the two user problem it follows that $\gamma^T(\mathbf{n}) = -s_1[\omega_1(n_1) - \omega_1(n_1-1)] + s_2[\phi_2(n_2) - \phi_2(n_2-1)]$. Since $\omega_1(\cdot)$ is convex (by assumption) and $\phi_2(\cdot)$ is convex (by Lemma 2), the desired result follows.

2) Inductive Step ($t < T$): We will show that $\gamma^t(\mathbf{n})$ is a non-decreasing function of n_1. The proof for monotonicity of $\gamma^t(\mathbf{n})$ as a function of n_2 is similar. We assume that the lemma is true in all states $\mathbf{n}$ in time slot $t+1$, for some $t < T$. We introduce the following notation for the sake of compactness: $\Delta_i f^t(\mathbf{n}) \triangleq f^t(\mathbf{n}) - f^t(\mathbf{n}-\mathbf{e}_i)$ and $\Delta_{ij} f^t(\mathbf{n}) \triangleq \Delta_i f^t(\mathbf{n}) - \Delta_i f^t(\mathbf{n}-\mathbf{e}_j)$ for $i, j = 1, 2$ and any function $f^t(\mathbf{n})$. Now, by definition

$$\begin{aligned} \gamma^t(\mathbf{n}) &= -s_1\Delta_1 V^{t+1}(\mathbf{n}) + s_2\Delta_2 V^{t+1}(\mathbf{n}) \\ \gamma^t(\mathbf{n}+\mathbf{e}_1) &= -s_1\Delta_1 V^{t+1}(\mathbf{n}+\mathbf{e}_1) + s_2\Delta_2 V^{t+1}(\mathbf{n}+\mathbf{e}_1) \\ \gamma^t(\mathbf{n}+\mathbf{e}_2) &= -s_1\Delta_1 V^{t+1}(\mathbf{n}+\mathbf{e}_2) + s_2\Delta_2 V^{t+1}(\mathbf{n}+\mathbf{e}_2) \end{aligned} \tag{11}$$

We want to show that $\Delta_1\gamma^t(\mathbf{n}+\mathbf{e}_1) \le 0$. From (11),

$$\begin{aligned} &\Delta_1\gamma^t(\mathbf{n}+\mathbf{e}_1) = \\ &-s_1\Delta_{11} V^{t+1}(\mathbf{n}+\mathbf{e}_1) + s_2\Delta_{21} V^{t+1}(\mathbf{n}+\mathbf{e}_1) \\ &\Delta_2\gamma^t(\mathbf{n}+\mathbf{e}_2) = \\ &-s_1\Delta_{12} V^{t+1}(\mathbf{n}+\mathbf{e}_2) + s_2\Delta_{22} V^{t+1}(\mathbf{n}+\mathbf{e}_2). \end{aligned}$$

Five different cases arise, depending on whether $\mathcal{P}_s^\star(2)$ schedules $\mathcal{Q}_1$ or $\mathcal{Q}_2$ in states $\mathbf{n}-\mathbf{e}_1$, $\mathbf{n}$, $\mathbf{n}-\mathbf{e}_2$, $\mathbf{n}+\mathbf{e}_1$, and $\mathbf{n}+\mathbf{e}_1-\mathbf{e}_2$ in time slot $t+1$. Due to space constraints, we present details only for two representative cases. The remaining three cases can be treated in similar fashion.

- *Case 1*: $\mathcal{P}_s^\star(2)$ schedules $\mathcal{Q}_1$ in states $\mathbf{n}-\mathbf{e}_1$, $\mathbf{n}$, $\mathbf{n}-\mathbf{e}_2$, $\mathbf{n}+\mathbf{e}_1$, and $\mathbf{n}+\mathbf{e}_1-\mathbf{e}_2$ in time slot $t+1$. In this case:
$$\begin{aligned} &\Delta_1 V^{t+1}(\mathbf{n}+\mathbf{e}_1) - \Delta_1 V^{t+1}(\mathbf{n}) = \\ &s_1\Delta_{11} V^{t+2}(\mathbf{n}) + \bar{s_1}\Delta_{11} V^{t+2}(\mathbf{n}+\mathbf{e}_1) \\ &\Delta_1 V^{t+1}(\mathbf{n}+\mathbf{e}_1) - \Delta_1 V^{t+1}(\mathbf{n}+\mathbf{e}_1-\mathbf{e}_2) = \\ &s_1\Delta_{12} V^{t+2}(\mathbf{n}) + \bar{s_1}\Delta_{12} V^{t+2}(\mathbf{n}+\mathbf{e}_1). \\ &\Delta_1\gamma^{t+1}(\mathbf{n}) = \\ &-s_1\Delta_{11} V^{t+2}(\mathbf{n}) + s_2\Delta_{12} V^{t+2}(\mathbf{n}) \\ &\Delta_1\gamma^{t+1}(\mathbf{n}+\mathbf{e}_1) = \\ &-s_1\Delta_{11} V^{t+2}(\mathbf{n}+\mathbf{e}_1) + s_2\Delta_{12} V^{t+2}(\mathbf{n}+\mathbf{e}_1). \end{aligned}$$
Combining the above with (12) we get,
$$\begin{aligned} \Delta_1\gamma^t(\mathbf{n}+\mathbf{e}_1) = s_1 \underbrace{\Delta_1\gamma^{t+1}(\mathbf{n})}_{\le 0} + \\ \bar{s_1} \underbrace{\Delta_1\gamma^{t+1}(\mathbf{n}+\mathbf{e}_1)}_{\le 0} \le 0, \end{aligned} \tag{12}$$
where the non-negativity of the terms on the right follows from our inductive assumption.
- *Case 2*: $\mathcal{P}_s^\star(2)$ schedules $\mathcal{Q}_2$ in states $\mathbf{n}-\mathbf{e}_1$, $\mathbf{n}$, $\mathbf{n}-\mathbf{e}_2$, $\mathbf{n}+\mathbf{e}_1$ and $\mathbf{n}+\mathbf{e}_1-\mathbf{e}_2$ in time slot $t+1$. In this case:
$$\begin{aligned} &\Delta_1 V^{t+1}(\mathbf{n}+\mathbf{e}_1) - \Delta_1 V^{t+1}(\mathbf{n}) = \\ &s_2\Delta_{11} V^{t+2}(\mathbf{n}+\mathbf{e}_1-\mathbf{e}_2) + \bar{s_2}\Delta_{11} V^{t+2}(\mathbf{n}+\mathbf{e}_1) \\ &\Delta_1 V^{t+1}(\mathbf{n}+\mathbf{e}_1) - \Delta_1 V^{t+1}(\mathbf{n}+\mathbf{e}_1-\mathbf{e}_2) = \\ &s_2\Delta_{12} V^{t+2}(\mathbf{n}+\mathbf{e}_1-\mathbf{e}_2) + \bar{s_2}\Delta_{12} V^{t+2}(\mathbf{n}+\mathbf{e}_1) \\ &\Delta_2\gamma^{t+1}(\mathbf{n}+\mathbf{e}_1) = \\ &-s_1\Delta_{12} V^{t+2}(\mathbf{n}+\mathbf{e}_1) + s_2\Delta_{22} V^{t+2}(\mathbf{n}+\mathbf{e}_1) \\ &\Delta_2\gamma^{t+1}(\mathbf{n}+\mathbf{e}_1-\mathbf{e}_2) = \\ &-s_1\Delta_{12} V^{t+2}(\mathbf{n}+\mathbf{e}_1-\mathbf{e}_2) + s_2\Delta_{22} V^{t+2}(\mathbf{n}+\mathbf{e}_1-\mathbf{e}_2). \end{aligned}$$
Combining the above with (12) we get,
$$\begin{aligned} \Delta_1\gamma_t(\mathbf{n}+\mathbf{e}_1) = s_2 \underbrace{\Delta_1\gamma^{t+1}(\mathbf{n}+\mathbf{e}_1-\mathbf{e}_2)}_{\le 0} + \\ \bar{s_2} \underbrace{\Delta_1\gamma^{t+1}(\mathbf{n}+\mathbf{e}_1)}_{\le 0}, \end{aligned} \tag{13}$$
where the non-negativity of the terms on the right follows from our inductive assumption.

The hypothesis of the lemma now follows from the principle of mathematical induction.

D. Proof of Theorem 1

Recall that $\mathcal{P}_s^\star(2)$ is fully characterized by the sign of γ^t. For fixed t, it follows from Lemma 3 that γ^t changes sign at most once from negative to positive as n_2 increases for fixed n_1. Thus, for fixed n_1, $\exists\ n_2 = \psi^t(n_1)$ such that the optimal decision *switches over* from $\mathcal{Q}_1$ to $\mathcal{Q}_2$ in state $(n_1, \psi^t(n_1))$. Since γ^t is a non-increasing function of n_1, it follows that $\gamma^t(n_1', n_2)$ can change sign only later than $\gamma^t(n_1, n_2)$ for fixed $n_1' > n_1$ as n_2 increases, implying $\psi^t(n_1') > \psi^t(n_1)$. The desired result follows from the definition of a switch-over policy.

E. Proof of Theorem 2

We will show that $\text{sgn}[\gamma^{t+1}(\mathbf{n})] = \text{sgn}[\gamma^t(\mathbf{n})]\ \forall\ t < T,\ \forall\ \mathbf{n}$, where $\text{sgn}[x] = 1$ if $x \ge 0$ and $\text{sgn}[x] = -1$ if $x < 0$. Since the optimal decision in time slot t is completely determined by the sign of γ^t, the implication is that the decisions of $\mathcal{P}_s^\star(2)$ are identical in time slot t and time slot $t+1$ for every state. Since t is arbitrarily chosen, the claim of the theorem follows.

We first assume that $\gamma^{t+1}(\mathbf{n}) \le 0$. Lemma 3 implies that $\gamma^{t+1}(\mathbf{n}-\mathbf{e}_2) \le 0$. However, $\gamma^{t+1}(\mathbf{n}-\mathbf{e}_1)$ could be negative or positive. Accordingly, we have two cases:

1) $\gamma^{t+1}(\mathbf{n}-\mathbf{e}_1) \le 0$: In this case,

$$\begin{aligned} V^{t+1}(\mathbf{n}) &= s_1 V^{t+2}(\mathbf{n}-\mathbf{e}_1) + \bar{s_1} V^{t+2}(\mathbf{n}) \\ V^{t+1}(\mathbf{n}-\mathbf{e}_2) &= s_1 V^{t+2}(\mathbf{n}-\mathbf{e}_1-\mathbf{e}_2) + \bar{s_1} V^{t+2}(\mathbf{n}-\mathbf{e}_2) \\ V^{t+1}(\mathbf{n}-\mathbf{e}_1) &= s_1 V^{t+2}(\mathbf{n}-2\mathbf{e}_1) + \bar{s_1} V^{t+2}(\mathbf{n}-\mathbf{e}_1). \end{aligned}$$

From (14), $\gamma^t(\mathbf{n}) = s_1 \underbrace{\gamma^{t+1}(\mathbf{n}-\mathbf{e}_1)}_{\le 0} + \bar{s_1} \underbrace{\gamma^{t+1}(\mathbf{n})}_{\le 0} \le 0.$

2) $\gamma^{t+1}(\mathbf{n}-\mathbf{e}_1) > 0$: In this case,

$$V^{t+1}(\mathbf{n}) = s_1 V^{t+2}(\mathbf{n}-\mathbf{e}_1) + \bar{s_1} V^{t+2}(\mathbf{n})$$
$$V^{t+1}(\mathbf{n}-\mathbf{e}_2) = s_1 V^{t+2}(\mathbf{n}-\mathbf{e}_1-\mathbf{e}_2) + \bar{s_1} V^{t+2}(\mathbf{n}-\mathbf{e}_2)$$
$$V^{t+1}(\mathbf{n}-\mathbf{e}_1) = s_2 V^{t+2}(\mathbf{n}-\mathbf{e}_1-\mathbf{e}_2) + \bar{s_2} V^{t+2}(\mathbf{n}-\mathbf{e}_1).$$

From (14), $\gamma^t(\mathbf{n}) = \bar{s_1}\gamma^{t+1}(\mathbf{n}) \le 0$. Using Lemma 3 and the definition of $\gamma^t(\mathbf{n})$, we can establish analogous results under the assumption $\gamma^{t+1}(\mathbf{n}) > 0$. In conclusion, $\text{sgn}[\gamma^t(\mathbf{n})] = \text{sgn}[\gamma^{t+1}(\mathbf{n})]$.

F. Proof of Theorem 3

For ease of exposition, we outline the proof for the case $N = 3$. The proof presented here extends in a natural way to $N > 3$. By definition,

$$\gamma_{12}^t(\mathbf{n}) = s_1 V^{t+1}(\mathbf{n}-\mathbf{e}_1) - s_2 V^{t+1}(\mathbf{n}-\mathbf{e}_2) + (s_2 - s_1) V^{t+1}(\mathbf{n}). \tag{14}$$

We want to show that $\text{sgn}[\gamma_{12}^t(\mathbf{n})] = \text{sgn}[\gamma_{12}^t(\mathbf{n}^{12})]$ for $t = 1, \ldots, T$, where $\mathbf{n} = (n_1, n_2, n_3)$ with $n_3 > 0$ and $\mathbf{n}^{12} = (n_1, n_2, 0)$. In words, the result of the comparison between $\mathcal{Q}_1$ and $\mathcal{Q}_2$ is unaffected by the presence of $\mathcal{Q}_3$. The proof is based on inductive arguments.

1) Base Case ($t = T$): From (14) and the boundary conditions, $\gamma_{12}^T(\mathbf{n}) = s_1[\omega_1(n_1) - \omega_1(n_1-1, 1)] + s_2[\omega_2(n_2) - \omega_2(n_2-1)]$, which is independent of n_3, thereby completing the proof.

2) Inductive Step ($t < T$): We assume that the hypothesis of the theorem is true in time slot $t+1$. Several cases arise, depending on the decision of $\mathcal{P}_s^\star(3)$ in states $\mathbf{n}-\mathbf{e}_1$, $\mathbf{n}-\mathbf{e}_2$ and $\mathbf{n}$ in time slot $t+1$. Due to space constraints, we only treat three representative cases. All other cases can be treated in similar fashion.

- $\mathcal{P}_s^\star(3)$ schedules $\mathcal{Q}_1$ in states $\mathbf{n}-\mathbf{e}_1$, $\mathbf{n}-\mathbf{e}_2$ and $\mathbf{n}$ in time slot $t+1$: In this case, we can show $\gamma_{12}^t(\mathbf{n}) = s_1\gamma_{12}^{t+1}(\mathbf{n}-\mathbf{e}_1) + (1-s_1)\gamma_{12}^{t+1}(\mathbf{n}) \le 0$, where the inequality follows because $\gamma_{12}^{t+1}(\mathbf{n}-\mathbf{e}_1) \le 0$ and $\gamma_{12}^{t+1}(\mathbf{n}) \le 0$ by assumption. Also, our inductive assumption implies that $\Pi_{\text{PW}}(N)$ schedules $\mathcal{Q}_1$ in states $\mathbf{n}^{12}-\mathbf{e}_1$ and $\mathbf{n}^{12}$ in time slot $t+1$, implying $\gamma_{12}^{t+1}(\mathbf{n}^{12}-\mathbf{e}_1) \le 0$ and $\gamma_{12}^{t+1}(\mathbf{n}^{12}) \le 0$. It follows, $\gamma_{12}^t(\mathbf{n}^{12}) = s_1\gamma_{12}^{t+1}(\mathbf{n}^{12}-\mathbf{e}_1) + (1-s_1)\gamma_{12}^{t+1}(\mathbf{n}^{12}) \le 0$. We conclude $\text{sgn}[\gamma_{12}^t(\mathbf{n})] = \text{sgn}[\gamma_{12}^t(\mathbf{n}^{12})] = -1$, as desired.
- $\mathcal{P}_s^\star(3)$ schedules $\mathcal{Q}_2$ in states $\mathbf{n}-\mathbf{e}_1$, $\mathbf{n}-\mathbf{e}_2$ and $\mathbf{n}$ in time slot $t+1$: In this case, we can show $\gamma_{12}^t(\mathbf{n}) = s_2\gamma_{12}^{t+1}(\mathbf{n}-\mathbf{e}_2) + (1-s_2)\gamma_{12}^{t+1}(\mathbf{n}) > 0$, where the inequality follows from our assumption. Also, our inductive assumption implies that $\Pi_{\text{PW}}(N)$ schedules $\mathcal{Q}_2$ in states $\mathbf{n}^{12}-\mathbf{e}_2$ and $\mathbf{n}$ in time slot $t+1$, implying $\gamma_{12}^{t+1}(\mathbf{n}^{12}-\mathbf{e}_2) \le 0$ and $\gamma_{12}^{t+1}(\mathbf{n}^{12}) \le 0$. It follows, $\gamma_{12}^t(\mathbf{n}^{12}) = s_2\gamma_{12}^{t+1}(\mathbf{n}^{12}-\mathbf{e}_2) + (1-s_2)\gamma_{12}^{t+1}(\mathbf{n}^{12}) \le 0$. We conclude $\text{sgn}[\gamma_{12}^t(\mathbf{n})] = \text{sgn}[\gamma_{12}^t(\mathbf{n}^{12})] = +1$, as desired.
- $\mathcal{P}_s^\star(3)$ schedules $\mathcal{Q}_3$ in states $\mathbf{n}-\mathbf{e}_1$, $\mathbf{n}-\mathbf{e}_2$ and $\mathbf{n}$ in time slot $t+1$: In this case, we can show $\gamma_{12}^t(\mathbf{n}) = s_3\gamma_{12}^{t+1}(\mathbf{n}-\mathbf{e}_3) + (1-s_3)\gamma_{12}^{t+1}(\mathbf{n})$. Now, our inductive assumption implies that $\text{sgn}[\gamma_{12}^{t+1}(\mathbf{n}-\mathbf{e}_3)] = \text{sgn}[\gamma_{12}^{t+1}(\mathbf{n})] = \text{sgn}[\gamma_{12}^{t+1}(\mathbf{n}^{12})]$. Thus, we conclude $\text{sgn}[\gamma_{12}^t(\mathbf{n})] = \text{sgn}[\gamma_{12}^{t+1}(\mathbf{n}^{12})] = \text{sgn}[\gamma_{12}^t(\mathbf{n}^{12})]$, where the last equality follows from Theorem 2.

We have established that $\text{sgn}[\gamma_{12}^t(\mathbf{n})] = \text{sgn}[\gamma_{12}^t(\mathbf{n}^{12})]$. Using similar analysis, we can establish synonymous equalities for γ_{23}^t and γ_{31}^t, and also extend the results to $N > 3$.

References

[1] H. Fattah and C. Leung, "An overview of scheduling algorithms in wireless multimedia networks," *IEEE Wireless Commun. Mag.*, vol. 9, no. 5, pp. 76–83, Oct. 2002.

[2] X. Liu, E. K. P. Chong, and N. B. Shroff, "A framework for opportunistic scheduling in wireless networks," *Computer Networks*, vol. 41, no. 4, pp. 451–474, Mar. 2003.

[3] L. Georgiadis, M. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1–144, 2006.

[4] L. Georgiadis, R. Guerin, and A. Parekh, "Optimal multiplexing on a single link: delay and buffer requirements," *IEEE/ACM Trans. Netw.*, vol. 43, no. 5, pp. 1518–1535, Sep. 1997.

[5] S. Shakkottai and R. Srikant, "Scheduling real-time traffic with deadlines over a wireless channel," *ACM/Baltzer Wireless Networks*, vol. 8, no. 1, pp. 13–26, Jan. 2002.

[6] K. Elsayed and A. Khattab, "Channel-aware earliest deadline due fair scheduling for wireless multimedia networks," *Springer Wireless Pers. Commun.*, vol. 38, no. 2, pp. 233–252, 2006.

[7] T. Ren, I. Koutsopolous, and L. Tassiulas, "QoS provisioning for real-time traffic in wireless packet networks," in *Proc. IEEE Globecom'02*, Taipei, Taiwan, Nov. 2002, pp. 1673–1677.

[8] K. Johnsson and D. Cox, "An adaptive cross-layer scheduler for improved QoS support of multi-class data services on wireless systems," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 334–343, Feb. 2005.

[9] A. Dua and N. Bambos, "Downlink wireless packet scheduling with deadlines," *IEEE Trans. Mobile Comput.*, vol. 6, no. 12, pp. 1410–1425, Dec. 2007.

[10] P. Chou and Z. Miao, "Rate-Distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 390–404, Apr. 2006.

[11] S. Wee, W. Tan, J. Apostolopoulos, and M. Etoh, "Optimized video streaming for networks with varying delay," in *Proc. IEEE ICME'02*, Laussane, Switzerland, Aug. 2002, pp. 89–92.

[12] G. Liebl, M. Kalman, and B. Girod, "Deadline-aware scheduling for wireless video streaming," in *Proc. IEEE ICME'05*, Amsterdam, Netherlands, Jul. 2005.

[13] J. Apostolopoulos, "Secure media streaming & secure adaptation for non-scalable video," in *Proc. IEEE ICIP'04*, Singapore, Oct. 2004, pp. 1763–1766.

[14] J. Chakareski and P. Frossard, "Rate-distortion optimized distributed packet scheduling of multiple video streams over shared communication resources," *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 207–218, Apr. 2006.

[15] M. Kalman, P. van Beek, and B. Girod, "Optimized transcoding rate selection and packet scheduling for transmitting multiple video streams over a shared channel," in *Proc. IEEE ICIP'05*, Genoa, Italy, Sep. 2005, pp. 165–168.

[16] P. Pahalawatta, R. Berry, T. Pappas, and A. Katsaggelos, "Content-aware resource allocation and packet scheduling for video transmission over wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 4, pp. 749–759, May 2007.

[17] D. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Belmont, MA: Athena Scientific, 2000.

[18] J. Walrand, *An Introduction to Queueing Networks*. Englewood Cliffs, NJ: Prentice Hall, 1988.

[19] M. Hassan, M. Krunz, and I. Matta, "Markov-based channel characterization for tractable performance analysis in wireless packet networks," *IEEE Trans. Wireless Commun.*, vol. 3, no. 3, pp. 821–831, May 2004.

[20] *Advanced video coding for general audiovisual services*, ITU-T Recommendation H.264, Mar. 2005.

[21] ITU H.264/MPEG-4 AVC reference software, Ver. JM10.2. [Online]. Available: http://iphome.hhi.de/suehring/tml/

[22] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2003.

Aditya Dua received his B.Tech and M.Tech degrees in electrical engineering from the Indian Institute of Technology, Bombay, India in 2002 and his Ph.D. degree in electrical engineering from Stanford University, California, USA in 2007. He is currently a senior systems engineeer at Qualcomm Inc. in Campbell, California. His research interests encompass both theoretical and practical aspects of modeling, analysis, architecture design, and performance engineering of communication networks and computing systems. His recent focus has been on online resource scheduling problems for supporting deadline sensitive multimedia traffic over wireless and wired packet networks. He is also interested in physical layer design issues like multiuser detection and equalization for wireless communication systems.

Carri W. Chan is currently a Ph.D. candidate in Electrical Engineering at Stanford University. She received her B.S. degree in Electrical Engineering from the Massachusetts Institute of Technology in 2004 and her M.S. degree in Electrical Engineering from Stanford University in 2006. Her research interests include stochastic modeling and optimization of communication networks. Her recent focus has been on scheduling problems for packetized multimedia streaming over wireless networks.

Nick Bambos received his Ph.D. in electrical engineering and computer science from U.C. Berkeley in 1989, after graduating in electrical engineering from the National Technical University of Athens, Greece in 1984. He served on the electrical engineering faculty of UCLA from 1990 to 1995 and joined Stanford University in 1996, where he is now a professor in the electrical engineering department and the management science and engineering department. His current research interests are in performance engineering of communication networks and computing systems, including queuing and scheduling issues in wireless an wireline networks, as well as ergodic random processes, queuing theory and adaptive control of stochastic processing networks.

John G. Apostolopoulos (S91, M97, SM06, F08) received the B.S., M.S., and Ph.D. degrees in EECS from MIT. He joined Hewlett-Packard Laboratories in 1997, where he is currently a Distinguished Technologist and Lab Director for the Multimedia Communications and Networking Lab. He also teaches and conducts joint research at Stanford University, where he is a Consulting Associate Professor of EE. In graduate school, he worked on the U.S. Digital TV standard and received an Emmy Award Certificate for his contributions. He received a best student paper award for part of his Ph.D. thesis, the Young Investigator Award (best paper award) at VCIP 2001 for his work on multiple description video coding and path diversity, was named one of the worlds top 100 young (under 35) innovators in science and technology (TR100) by Technology Review in 2003, and was co-author for the best paper award at ICME 2006 on authentication for streaming media. His work on media transcoding in the middle of a network while preserving end-to-end security (secure transcoding) was recently adopted by the JPEG-2000 Security (JPSEC) standard. He currently serves as chair of the IEEE IMDSP and member of MMSP technical committees, and recently was general co-chair of VCIP06 and technical co-chair for ICIP'07. His research interests include improving the reliability, fidelity, scalability, and security of media communication over wired and wireless packet networks.