Abstract—We present a cross-layer design of transmitting scalable video streams from a base station to multiple clients over a shared fading wireless network by jointly considering the application layer information and the wireless channel conditions. We first design a long-term resource allocation algorithm that determines the optimal wireless scheduling policy in order to maximize the weighted sum of average video quality of all streams. We prove that our algorithm achieves the global optimum even though the problem is not concave in the parameter space. We then devise two on-line scheduling algorithms that utilize the results obtained by the long-term resource allocation algorithm for user and packet scheduling as well as video frame dropping strategy. We compare our schemes with existing video scheduling and buffer management schemes in the literature and simulation results show our proposed schemes significantly outperform existing ones.

Index Terms—Scheduling, video streaming, scalable video, fading, wireless networks

I. INTRODUCTION

Recent years have witnessed increasing popularity of streaming video over wireless networks as both wireless data communication and video compression techniques undergo significant progress. On one hand, the data transmission rates of wireless networks are steadily growing, e.g., 1Gbps target rate for nomadic and 100Mbps for mobile users in 4G systems [9]. On the other hand, H.264/MPEG4-AVC [1] achieves more efficient video compression and the Scalable Video Coding (SVC) extension [16] of H.264/MPEG4-AVC obtains both high coding efficiency and high scalability. Nevertheless, because the wireless medium is often shared by many users, it is still important to adapt to the wireless channel conditions in order to satisfy stringent bandwidth and delay requirement of video traffic.

Streaming video over wireless networks has been studied extensively by many researchers, but much of the previous work ([7], [8], [25], [15], [22], [26] and the references therein) has focused on the single-stream scenario where the transmitter of a video streaming service adaptively adjusts its transmission rate, re-transmission, video-truncation, Forward Error Correction (FEC) and/or Hybrid ARQ (HARQ) policy in order to optimize the received video quality. Multi-user streaming where the wireless radio resources are shared by multiple streaming users has also been considered in [24], [10], [11], [13], [14], [12]. However, none of them considered exploiting the fading wireless channel characteristics and the scalable video encoding jointly.

Realtime radio resource scheduling algorithms have been studied in [17], [3], [18] by considering the delay requirement and channel conditions. An alternative formulation based on optimizing a concave utility function and rate control over a fading wireless networks has been considered [6], [19], [4]. However, these algorithms only warrant asymptotic convergence without explicitly considering video applications (such as the hard deadline constraint and bursty rate requirements).

In this work, we consider the cross-layer optimization of rate adaptation and exploiting the multi-user diversity for video streaming using scalable video coding over a shared fading wireless channel. In a fading wireless channel, it is important to exploit the multi-user diversity, i.e., by scheduling users in relatively good channel conditions. With SVC [16], it is possible to adapt the video transmission rate to the wireless channel capacity.

We first develop an empirical model to relate the average video quality (measured by PSNR (Peak Signal-to-Noise Ratio)) and the average throughput based on SVC [16]. We then formulate the following cross-layer problem: maximizing the weighted sum of video quality of all users subject to the achievable long-term (ergodic) rate constraint under a fading wireless channel model. To solve the problem, we develop a long-term radio resource allocation algorithm which determines the wireless scheduling policy and the parameters used by the scheduling policy. We prove the convergence and the optimality of the proposed algorithm under mild conditions.

Aiming to exploit multi-user diversity, we propose two online scheduling algorithms that meet the realtime video traffic QoS (Quality of Service) requirement. The Static scheduling algorithm only uses the results obtained from the aforementioned long-term resource allocation algorithm. And, the Dynamic scheduling algorithm further adapts the scheduling parameters to meet the instantaneous rate, deadline requirement of video traffic and wireless channel conditions. The underlying objective function for the optimal scheduling problem is non-convex/concave and non-differentiable. We transform the problem to an equivalent, but differentiable one. We then develop a gradient-based approach to solve the problem and prove that it converges to the optimal solution even though the objective function is non-convex/concave. We also design frame dropping strategies that determine when and which frames will be dropped.

We carry out extensive simulations to validate our proposed schemes using SVC-encoded real video sequences. Simulation results show that our proposed scheduling schemes achieve significant improvement over existing real-time video schedul-
ing algorithms in the literature. Our proposed schemes obtain up to 8-10 dB gain of average video quality compared to several well-known existing schemes. Moreover, our proposed schemes are much more robust under weak wireless channel conditions. At low SINR, some videos are not decodable with existing video scheduling schemes because of heavy packet drop, while nearly all videos are decodable even at very weak channel conditions by using the dynamic scheme. Finally, our proposed algorithms are robust to channel estimation errors.

Our main contributions are threefold. First, we design a long-term radio resource allocation algorithm and prove that it achieves the global optimum of the objective function. Second, we design two online scheduling algorithms and also prove the optimality of the obtained scheduling parameters used by the dynamic scheduling algorithm. Third, we develop intelligent frame/layer dropping strategies based on both the long-term resource allocation algorithm and the dynamic scheduling algorithm.

The rest of the paper is organized as follows. In Section II, we discuss SVC and the rate-quality model. The proposed long-term radio resource allocation scheme is given in Section III. We present two algorithms for on-line scheduling of video streaming traffic in Section IV. Simulation results based on SVC-encoded real video sequences are reported in Section V and the paper is concluded in Section VI.

II. SCALABLE VIDEO CODING AND RATE-QUALITY MODEL

SVC can be referred to as both the general concept of scalable video coding and the special extension [16] of H.264/MPEG4-AVC [1]. As a general concept, an SVC stream has a base layer and several enhancement layers. As long as the base layer is received, the receiver can decode the video stream. As more enhancement layers are received, the decoded video quality is improved. For a detailed overview of SVC, please refer to [16].

As in [27], we use PSNR (Peak Signal to Noise Ratio) as a measure of video quality and develop a model to characterize the relationship between the rate and PSNR. It turns out that the relationship between the PSNR $S_i$ of video stream $i$ and the video rate $r$ can be described as a piece-wise linear function:

$$ S_i(r) = \begin{cases} 
S_i^0 + L_i(r - r_i^0) & \text{if } r \leq r_i^0 \\
S_i^0 + K_i(r - r_i^0) & \text{if } r_i^0 < r \leq r_i^{\text{max}} \\
S_i^0 + K_i(r_i^{\text{max}} - r_i^0) & \text{else}
\end{cases} $$

(1)

where $r_i^0, S_i^0$ are the rate and PSNR when only the base quality layers are included, and the last line specifies the maximum encoding rate of a video sequence. In Figure 1, we plot both the sample points of rate and PSNR and the model we have obtained for eight video sequences: News, Hall, Silent, City, Foreman, Crew, Harbour, and Mobile, all of which can be downloaded from [21] (more video sequence models are obtained but are omitted for the clarity of the figure). The points in the figure show the sample points of the rate and PSNR and the lines show the regression models we obtained based on Eq. (1). It can be seen that our model is quite accurate. Note that $L_i > K_i > 0$ based on the models of the real video sequences, so the function $S_i(r)$ is concave, continuous, and non-decreasing with respect to $r$.

Along with the model, we also obtain the priority of different video layers. Note that a layer can be specified with a temporal level $t$ and quality level $q$ (when the video is encoded with temporal and quality scalability). We use $q = 0$ to represent a base (quality) layer. We then determine the priority of each layer based on the average ratio of the PSNR drop to the rate decrease for each truncated layer; the details are omitted due to space limit. Once the priority is determined, we can drop video layers in ascending order of their priority to obtain any desired transmission rate and the corresponding PSNR can be obtained based on the model (1).

Stuhlmuller [20] et al. also proposed a rate-distortion model. The major difference between our model and their model is that the different rates in the model in [20] are obtained by encoding with different source coding rates and INTRA rates while those in our model are obtained via dropping layers in the increasing order of layer priority. As a result, our model is more appropriate in wireless networks where video packets may be dynamically dropped depending on wireless channel conditions.

III. LONG-TERM RADIO RESOURCE ALLOCATION FOR STREAMING VIDEO

When a wireless base station receives multiple requests for streaming service of different video sequences, it has to decide i) how much radio resources should be allocated to each user in order to maximize the overall video quality, ii) how to achieve the desired radio resource allocation. In a non-fading wireless network, these problems can be simply solved by Time Division Multiple Access (TDMA) (e.g. [11], [12]). But mobile networks often experience fast fading. In a fading wireless network, channel state dependent scheduling is often used to exploit multi-user diversity.

We describe the problem formulation under a fading channel in subsection III-A. In subsection III-B, we develop an algorithm to find the optimal scheduling policy and the associated parameters. In subsection III-C, we rigorously prove that the obtained scheduling policy and its parameters achieve the global optimum of the overall video quality defined in subsection III-A under mild conditions.
A. Problem formulation

We assume that the instantaneous transmission rate for each user \(i\) at each time slot \(t\) is

\[
C(h_i(t)) = B \log(1 + \rho |h_i(t)|^2 / \Gamma)
\]

where \(B\) is the channel bandwidth, \(\rho\) is the transmission power, \(h_i(t)\) is the channel gain of user \(i\), normalized with respect to the standard deviation of noise (and interference), \(\Gamma \geq 1\) represents the gap between the actual coding scheme and the Shannon capacity. We assume discrete time system and a TDMA transmission strategy: at each time slot, the server picks only one user (which may depend on the channel states of all the users) and sends information with the supportable rate of the channel of this scheduled user.

Our objective is to maximize the weighted sum of average PSNR of all users:

\[
\max \sum_{i=1}^{n} w_i S_i(r_i) \quad \text{s.t.} \quad r \in \mathcal{R}
\]

where \(w_i\) is the weight of user \(i\), \(S_i\) is the PSNR of user \(i\) as modeled in Eq. (1), \(r = (r_1, \cdots, r_n)\) and \(\mathcal{R}\) is the achievable ergodic (long-term average) rate region.

The major challenge in solving problem (3) is that the achievable rate region cannot be explicitly specified in a fading environment. We next develop an algorithm to solve the problem.

B. Algorithm to solve problem (3)

We first consider a randomized channel-state-dependent TDMA scheduling strategy (which is also called static service split (SSS) in [2]): given the channel states \(h\) of all users at a time slot, the scheduler picks user \(i\) with probability \(\pi_i(h)\) and \(\sum_{i=1}^{n} \pi_i(h) \leq 1\). The achievable rate region under this randomized TDMA strategy is decided by the randomized scheduling function \(\pi(h)\):

\[
\mathcal{R} = \{ r : r_i \leq \mathbb{E}_h [\pi_i(h) C(h_i)], 1 \leq i \leq n \}
\]

It was shown in [19] that the achievable rate region under this randomized TDMA strategy is convex and bounded, and closed.

Because the PSNR function \(S_i(r)\) is a non-decreasing function of the rate \(r\), there must exist an optimal solution to problem (3) on the boundary of the achievable rate region (i.e., which is Pareto-optimal). It was proved in [2] that for any achievable rate vector \(r\) on the boundary of the achievable rate region, there exists a vector \(\mu = (\mu_i > 0, i = 1, \cdots, n)\) such that the rate vector \(r\) can be achieved by the following scheduling function:

\[
\pi_i(h) > 0 \Rightarrow h \in \{ h : \mu_i C(h_i) \geq \mu_k C(h_k), \forall k \neq i \}.
\]

The solution in (5) is essentially a deterministic scheduling function: at each time slot, the user with the largest \(\mu_i C_i\) is chosen for scheduling, if we ignore the set of channel states \(h\) for which \(\mu_i C(h_i) = \mu_k C(h_k)\) which in fact has zero probability if the distribution of \(h\) is continuous. We call the scheduling policy defined by \(s(h)\) in (5) as maximal scheduling policy, due to the fact that this scheduler only obtains the set of rates in the boundary which are Pareto-optimal.

We can now view a boundary point \(r = (r_1, \cdots, r_n)\) as a function of the parameter set \(\mu = (\mu_1, \cdots, \mu_n)\), where the average rate \(r_i\) of user \(i\) can be written as

\[
r_i = \mathbb{E}[C(h_i)(\mu_i C(h_i) > \mu_j C(h_j) \text{ for all } j \neq i)].
\]

where \(\mathcal{I}\) is an indicator function.

Let \(\gamma_i = \rho |h_i|^2\) denote the SINR (Signal-to-Noise-Ratio) of the user \(i\) with PDF (Probability Density Function) function \(f_{\gamma_i}(\gamma)\) and CDF (Cumulative Density Function) function \(F_{\gamma_i}(\gamma)\). Let \(R(\gamma) = B \log(1 + \gamma / \Gamma)\). Thus, the rate \(r_i\) can be computed as

\[
r_i(\mu) = \int_0^\infty R(\gamma) \mathcal{I}_{\gamma \neq \mu_i} F_{\gamma_i}(R^{-1}(\mu_i R(\gamma) / \mu_j)) f_{\gamma_i}(\gamma) d\gamma
\]

Now our objective is simply to

\[
\maximize \quad Y = \sum_{i=1}^{n} w_i S_i(r_i(\mu))
\]

Note that \(S_i(r_i)\) is a non-decreasing concave continuous function of \(r_i\).

A general approach to solve an optimization problem is the gradient-based method. But the challenge with problem (8) is i) the function \(Y\) is not differentiable at the points when some \(r_i = r_i^0\) or \(r_i = r_i^{max}\), ii) the function \(Y\) is generally not concave (or convex) with respect to \(\mu\). When a function is not concave or convex, the solution generated by the gradient-based approach is often only a local maximum (or minimum) but not a global maximum. In the following we develop an algorithm and prove that the limit point of the algorithm is the global maximum of problem (8) under mild conditions.

To resolve the first issue, we note that although \(S_i\) is not differentiable at \(r_i = r_i^0\) or \(r_i = r_i^{max}\), it has one-sided derivatives. For functions with one-sided derivatives, the following lemma is a simple generalization of the first-order necessary optimality conditions.

\textbf{Lemma 1} If \(\mu^*\) is a local maximum of \(Y\), then

\[
Y_{i+}(\mu^*) \leq 0 \quad \text{and} \quad Y_{i-}(\mu^*) \geq 0
\]

for all \(1 \leq i \leq n\), where \(Y_{i+}(\mu) = \frac{\partial Y}{\partial \mu_i+}(\mu)\) is the right-sided partial derivative and \(Y_{i-}(\mu) = \frac{\partial Y}{\partial \mu_i-}(\mu)\) is the left-sided partial derivative.

Now with the one-sided partial derivative, we can obtain an iterative modified gradient-based solution as follows. First we compute the modified gradient \(g(k) = (g_1^{(k)}, g_2^{(k)}, \cdots, g_n^{(k)})\) in each iteration \(k:\)

\[
g_i^{(k)} = \begin{cases} 
0, & \text{if } Y_{i+}(\mu^{(k)}) \leq 0 \text{ and } Y_{i-}(\mu^{(k)}) \geq 0 \\
Y_{i+}(\mu^{(k)}), & \text{if } Y_{i+}(\mu^{(k)}) > 0 \text{ and } Y_{i-}(\mu^{(k)}) \geq -Y_{i+}(\mu^{(k)}), \\
Y_{i-}(\mu^{(k)}), & \text{otherwise}
\end{cases}
\]

Let \(i_0 = \arg\max_i(g_i^{(k)})\). The ascent direction is chosen to be \(d^{(k)} = (0, \cdots, g_{i_0}^{(k)}, \cdots, 0)\). In other words, the ascent direction \(d^{(k)}\) is all zero except the \(i_0\)th element, which takes the value \(g_{i_0}^{(k)}\). Note that \(g^{(k)}\) is not necessarily an ascent direction but \(d^{(k)}\) is unless \(d^{(k)}\) is zero.
TABLE I

A1: Pseudo-code to find the solution to problem (8)

\[ T^n, \epsilon, \sigma, \alpha_0 \text{ are positive constant values, } \epsilon \text{ is close to 0, and } 0 < \sigma < 1. \]

1: Select a starting point \( \mu^{(0)} = 1 \) for all \( 1 \leq i \leq n. \)
2: Compute \( d^{(0)} \) from \( \mu^{(0)} \)
3: \( k = 0; \)
4: while \( (d^{(k)})^2 \geq \epsilon \) do
5: \( \alpha = \alpha_k; \)
6: while \( Y(\mu^{(k)} + \alpha d^{(k)}) - Y(\mu^{(k)}) - \sigma \alpha \cdot |d^{(k)}|^2 \) do
7: \( \alpha = \alpha \cdot \beta; \)
8: \( \text{end while} \)
9: \( \mu(k) = \mu^{(k)} + \alpha \cdot d^{(k)} \)
10: \( \text{end while} \)
11: \( \frac{T}{\partial \mu} \cdot t \)
12: \( \text{Recompute } d^{(k)} \) from \( \mu^{(k)} \)
13: \( \text{end while} \)

Stepsize selection using modified Armijo Rule: In any gradient-based approach, we also need to choose the step size \( \alpha(k) \) appropriately in order for the algorithm to converge to a local maximum. For this purpose, we apply a modified Armijo rule [5] where the gradient \( \nabla Y(\mu^{(k)}) \) is replaced with \( d^{(k)} \) because the gradient \( \nabla Y(\mu^{(k)}) \) may not exist. The pseudo-code of our algorithm A1 is listed in Table I.

We can now show the convergence of the algorithm, which is summarized next. The proof follows similar ideas in the standard proof of the limit points of the stationary points for gradient-based methods (see e.g., [5]) and is omitted.

**Lemma 2** Algorithm A1 converges to a point \( \mu^* \) satisfying the necessary conditions of a local maximum in Eq. (9), assuming that the finite stopping condition (in line 4 of Algorithm A1) is removed.

C. Optimality of the algorithm A1

For all practical purpose, we can assume that the PSNR function \( Y \) is continuously differentiable, non-decreasing and concave with respect to \( r \) (we can always connect the two line segments using a smooth curve to make the function continuously differentiable). Under this assumption, we can prove that the limit point of algorithm A1 is a global maximum. This is quite significant as the function \( Y \) is generally not concave with respect to the controlling variable \( \mu \). We first prove the following lemma.

**Lemma 3** For any rate vector \( r(\mu) \) achieved using the maximal scheduling policy (5) with parameter \( \mu \), all achievable rate region is below the hyper-plane defined by

\[
\left\{ r(\mu) + \frac{\partial r}{\partial \mu} \mid u : u \in R^n \right\}
\]

where \( \frac{\partial r}{\partial \mu} \mid u \) is a matrix whose element at \( i \)th row and \( j \)th column is \( \frac{\partial r_j}{\partial \mu_i} \mid u \). More precisely, for any other achievable rate vector \( r' \), we can find a vector \( u \in R^n \) such that \( r' \leq r + \frac{\partial r}{\partial \mu} \mid u \) element wise.

**Proof.** The intuition is quite clear. At any point \( r \) that is achieved by the maximal scheduling scheme with parameter \( \mu \), there is a tangent hyperplane. All the achievable rate region is below the tangent hyperplane. Next we give a rigorous proof.

Since \( r \) is on the boundary of the achievable rate region, which is convex, from Proposition B.12 in [5], there is a vector \( u \neq 0 \) such that

\[
b^T(r' - r) \leq 0
\]

(11)

for any \( r' \) in the achievable rate region. We now show two properties of the vector \( b \). First, \( b^T \frac{\partial r}{\partial \mu} \mid u \) for all \( k \). Let \( r' \) be the rate achieved using parameter \( \mu_j = (\mu_1', \ldots, \mu_n') \) where \( \mu_j' = \mu_j \) for all \( j \neq k \) and \( \mu_k' > \mu_k \). Divide both sides of Eq. (11) by \( \mu_k' - \mu_k \) we obtain that \( b^T(r' - r) \leq 0 \). Let \( \mu_k' > \mu_k \) and \( \mu_k' \to \mu_k + \) we have \( b^T \frac{\partial r}{\partial \mu} \mid u \to 0 \). Similarly, if we let \( \mu_k' < \mu_k \) and \( \mu_k' \to \mu_k - \), we get \( b^T \frac{\partial r}{\partial \mu} \mid u \geq 0 \). Since \( r \) is continuously differentiable with respect to \( \mu \), we obtain \( b^T \frac{\partial r}{\partial \mu} \mid u = 0 \) for all \( 1 \leq k \leq n \).

The second important property is \( b \geq 0 \) element wise. Suppose \( b = (b_1, \ldots, b_n) \). We choose \( r'_k < r_k \) and \( r'_j = r_j \) for all \( j \neq k \). Clearly, \( r' \) is a achievable rate vector. So from Eq. (11), we get \( b^T(r' - r) = b_k(r'_k - r_k) \). Since \( r'_k < r_k \), we have \( b_k \geq 0 \).

Next we look at the intersection point of the line \( \{ r' + bt : t \in R \} \) and the hyper plane (10). At the intersection point, we have

\[
r' + bt = r + \frac{\partial r}{\partial \mu} \mid u
\]

(12)

Multiplying both sides by \( b^T \) from the left and doing a little re-arrangement, we get

\[
b^T bt = b^T(r - r') + b^T \frac{\partial r}{\partial \mu} \mid u = b^T(r - r') \geq 0
\]

(13)

where the second equality is because of the first property of vector \( b \) and the last inequality is from Eq. (11). Since \( b^T b > 0 \), we obtain \( t \geq 0 \). Therefore, the intersection point \( r' + bt \) is on the hyper plane (10) and is larger than or equal to \( r' \) element wise (because \( t \geq 0 \) is a scalar and \( b_j \geq 0 \) for all \( j \)).

Next we prove the main theorem. Note that we can always connect the two line segments of the function \( S_i(r_i) \) with a smooth curve such that the function \( S_i(r_i) \) is continuously differentiable.

**Theorem 1** The limit point of the algorithm A1 is a global maximum of function \( Y \) assuming that the PSNR function \( S_i(r_i) \) is non-decreasing, concave, and continuously differentiable with respect to \( r_i \).

**Proof.** Let \( \mu \) and \( r(\mu) \) be the limit point of the algorithm A1. Clearly \( r(\mu) \) is a boundary point of the achievable rate region. Applying Lemma 3, for any achievable rate vector \( r' \), there exists a vector \( u \) such that \( r' \leq r + \frac{\partial r}{\partial \mu} \mid u \) element wise. Because of the non-decreasing property of the function \( Y \) over each \( r_i \), we have

\[
Y(r') \leq Y(r + \frac{\partial r}{\partial \mu} \mid u)
\]

(14)

We next show that, for any point \( r + \frac{\partial r}{\partial \mu} \mid u \) on the hyperplane (10),

\[
Y(r(\mu) + \frac{\partial r}{\partial \mu} \mid u) \leq Y(r(\mu))
\]

(15)
given that $\mu$ is the limit point of algorithm A1. With the assumption that $Y$ is continuously differentiable with respect to both $r$ and $\mu$, from Lemma 2, the limit point $\mu$ of algorithm A1 satisfies, for any $1 \leq k \leq n$, 
\[
0 = \frac{\partial Y}{\partial \mu_k} = \sum_{i=1}^{n} \frac{\partial Y}{\partial r_i} \frac{\partial r_i}{\partial \mu_k} 
\] (16) 
where the second equality is from the chain rule of partial derivative.

Because of the concavity of $Y$ over $r$,
\[
Y(r + \alpha \frac{\partial r}{\partial \mu} u) \leq Y(r) + \nabla Y(r)^T \frac{\partial r}{\partial \mu} u 
\]
\[
= Y(r) + \sum_{j=1}^{n} \frac{\partial Y}{\partial r_j} \sum_{k=1}^{n} \frac{\partial r_j}{\partial \mu_k} u_k 
\]
\[
= Y(r) + \sum_{k=1}^{n} \sum_{j=1}^{n} \frac{\partial Y}{\partial r_j} \frac{\partial r_j}{\partial \mu_k} u_k 
\]
\[
= Y(r) 
\] (17) 
where the last equality is from Eq. (16).

Combining Eq. (14) and (17), we obtain that $Y(r') \leq Y(r(\mu))$ for any achievable rate vector $r'$ provided that $\mu$ is the limit point of the algorithm A1.

In general, for a non-convex (or non-concave) problem, a stationary point is only a local maximum (or minimum). It appears surprising that although the objective function $Y$ is not a concave function with respect to $\mu$, the stationary point is the global maximum of $Y$. To further understand the problem, we consider a two-user scenario and fix $\mu_1 = 1$.

The objective function $Y$ is then the function of $\mu_2$. Using parameters derived from two real video sequences, we plot the sum PSNR of the two users vs $\mu_2$ in Fig 2. We can observe from the figure that the function $Y$ is neither convex nor concave, and that the stationary points of the function are not even unique. However, all the stationary points are indeed the global maximum of $Y$.

IV. ONLINE SCHEDULING FOR SVC VIDEO STREAMING

An online scheduling algorithm for real-time video applications needs to address three issues:

1. **User scheduling**: at each time slot, which user should be scheduled?
2. **Frame scheduling**: after a user is selected, which packets/frames of the selected user should be transmitted?
3. **Dropping strategy**: when does it need to drop frames and which frames should be dropped?

The resource allocation algorithm presented in the section produces two results: 1) the vector $\mu$ used for user-scheduling, 2) the achievable average rate $r_i$ for each user $i$. We next describe two online scheduling schemes exploiting these results. In the first scheme, we simply apply the results obtained from the resource allocation algorithm. In the second scheme, we also consider the bursty and dynamic arrival and the deadline of video frames.

A. Static scheduling scheme

In this first scheme, the vector $\mu$ is computed from the previous section and is fixed during the process of streaming ($\mu$ may be re-computed when new users join or some users leave). At each time slot, the user with the largest $\mu_i C_i$ is chosen for scheduling, where $C_i$ is the current channel capacity of user $i$, and the vector $\mu = \{ \mu_1, \mu_2, \ldots, \mu_n \}$ is computed from the long-term resource allocation algorithm in the previous section.

Frame scheduling for the selected user is based on both the deadline and priority of the packets. We differentiate two types of deadlines. **Playout deadline** is the time a frame need to be displayed. **Decoding deadline** is the earliest time that a frame is needed for decoding itself or other frames. The decoding deadline of a frame can be computed as the minimum playout deadline of all frames that depend on it. We then schedule packets of a given user in the order of their decoding deadline. Those packets with the same decoding deadline are scheduled in the order of their priority, which is obtained in Section II.

As to the dropping strategy, there are two types of dropping. The first is late dropping, which happens when the playout deadline of a packet is passed. If the base layer of a frame is dropped, all dependent frames are dropped too. Note that when all packets of a frame are either successfully transmitted or dropped, the decoding deadline of the frames that it depends on need to be re-computed.

The second type of dropping is early dropping. With the achievable rate computed from the previous subsection, we can pre-determine which layers should be dropped based on the rate requirement. We find the minimum priority such that the average data rate of the packets with priority higher than or equal to the minimum priority does not exceed the achievable rate computed from the previous section. All packets with priority lower than the minimum priority are dropped at the beginning of the video streaming.

B. Dynamic scheduling scheme

The dynamic scheduling is built on top of the static scheduling scheme with two additional enhancements. The first enhancement is on the user scheduling. At each time slot, still, the user with the largest $\mu_i C_i$ is selected for scheduling. However, the vector $\mu$ is periodically updated to reflect both the bursty arrival and the deadline of video traffic.
Assume that for user $i$, the size of total packets that need to be transmitted before the deadline $T_j$ is $Q_j$. We define the target rate $\bar{r}_i$ for user $i$ to be

$$\bar{r}_i = \max_{j=1}^n \frac{Q_j}{T_j - t}$$

(18)

where $t$ is the current time.

Now with the target rate $\bar{r}_i$ for each user $i$, we ask whether there exists a vector of $\mu$ such that the target rate $\bar{r}_i$ can be satisfied for every user $i$. We consider the following max-min problem.

$$\max \min \frac{r_i(\mu)}{\bar{r}_i}$$

over all possible choices of $\mu$. Clearly, if the optimal value of Eq. (19) is larger than or equal to 1, the target rate $\bar{r}$ is schedulable and vice versa. The problem (19) is not easy to solve as i) the problem is non-convex and ii) the derivative does not exist and the gradient-based approach cannot be directly applied. But the following result was obtained in [6]:

**Lemma 4** Solving problem (19) is equivalent to solving the following problem: find $\mu$ such that

$$\frac{r_1(\mu)}{r_1} = \frac{r_2(\mu)}{r_2} = \ldots = \frac{r_n(\mu)}{r_n},$$

(20)

assuming that the channel distribution function $f_{\gamma_i}(\gamma)$ is a continuous function of $\gamma$ for all $i$.

To solve problem (20), we define $g = (g_1, \ldots g_n)$, $\bar{g}$, and $h(\mu)$ as follows:

$$g_i(\mu) = \frac{r_i(\mu)}{\bar{r}_i}$$

$$\bar{g} = \sum_{i=1}^n g_i(\mu)/n$$

$$h(\mu) = \frac{1}{2} \sum_{i=1}^n (g_i(\mu) - \bar{g})^2.$$  

(21)

In each iteration $k$, we compute $\bar{g}^{(k)}$ and treat it as a fixed value during the iteration. We then solve the problem of minimizing the function $h(\mu)$ in Eq. (21) using Gauss-Newton method. To do so, we choose the direction

$$d^{(k)} = (\nabla g(\mu) \nabla g(\mu)^T)^{-1} \nabla g(\mu) (g(\mu) - \bar{g}^{(k)}).$$

(22)

and update $\mu^{(k)}$ as

$$\mu^{(k+1)} = \mu^{(k)} - \alpha^{(k)} d^{(k)}$$

where $\alpha^{(k)}$ is the step size chosen by Armijo rule. The pseudo-code of the algorithm A2 is presented in Table II. The following lemma follows from the stationarity of limit points of gradient-based approach (see e.g. [5], page 43).

**Lemma 5** Algorithm A2 converges to a stationary point of the function $h(\mu)$ (defined in Eq. (21)) assuming that the finite stopping condition in the outer-loop is removed.

In general, the stationary point of a function $h(\mu)$ is not necessarily a global minimum of the function because of the non-convexity of the function. But, surprisingly, we can prove that the limit point of the algorithm A2 is a global minimum (i.e., 0) of the function $h(\mu)$, which is also the unique solution to problems (19) and (20), even though the function $h(\mu)$ may not be convex.

The proof is presented in the appendix.

**Theorem 2** Algorithm A2 converges to the optimum solution to the problems (19) and (20) assuming that the finite stopping condition in the outer-loop is removed.

The second enhancement of this dynamic scheduling scheme is on the frame dropping strategy. Note that the algorithm A2 not only produces the new vector $\mu$ that is used for user scheduling, but also the value $\eta = \max \min_{i=1}^n \frac{\bar{r}_i}{\bar{r}}$. If $\eta > 1$, it indicates that the target rate vector $\bar{r}$ is un-achievable and some video frames need to be dropped. To overcome the short-term uncertainty of wireless channels, in this dynamic scheduling scheme, we maintain a target range $(\bar{\eta}, \bar{\eta})$ of $\eta$. During the periodic reevaluation of the vector $\mu$ and $\eta$, if $\eta < \bar{\eta}$, we will start to drop packets, and if $\eta > \bar{\eta}$, we will put some dropped packets (whose playout deadline is not passed yet) back to the queue. To support the function of putting dropped packets back to the queue, we do not really drop a packet unless its playout deadline is passed. Instead, we simply mark it to be dropped.

When we need to drop some packets (i.e., $\eta < \bar{\eta}$) or need to put some dropped packets back to the queue (i.e., $\eta > \bar{\eta}$), we first choose the user using round robin. After the user is selected, we choose the packets that have the lowest priority within a window from now when dropping packets, and choose the packets that have the highest priority among those marked as dropped when putting dropped packets back to the queue. Then we re-compute the vector $\mu$ and $\eta$ using Algorithm A2 and repeat the process until the value $\eta$ falls between $\bar{\eta}$ and $\eta$. We use the final result $\mu$ for subsequent user scheduling.

**V. SIMULATION RESULTS**

The following settings are used for simulations. All video sequences are encoded at 30Hz with GOP size of 16 pictures and an intra period of 64 frames (about 0.5Hz). Wireless channels are generated based on Rayleigh fading model unless specified otherwise. Channel bandwidth is assumed to be

<table>
<thead>
<tr>
<th>TABLE II Algorithm A2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A2: Pseudo-code to solve problem (20)</td>
</tr>
<tr>
<td>$\mu$, $\bar{\mu}$, $\bar{\mu}$ are positive constant values, $\epsilon$ is close to 0, and $0 &lt; \epsilon &lt; 1/2$</td>
</tr>
<tr>
<td>1: Select a starting point $\mu^{(0)} = 1$ for all $1 \leq i \leq n$</td>
</tr>
<tr>
<td>2: Compute $g(\mu^{(0)})$, $\bar{g}^{(0)}$, and $h(\mu^{(0)})$ using Eq. (21)</td>
</tr>
<tr>
<td>3: Compute $d^{(0)}$ according to Eq. (22)</td>
</tr>
<tr>
<td>4: $k = 0$</td>
</tr>
<tr>
<td>5: while $</td>
</tr>
<tr>
<td>6: $\mu^{(0)}$ Choose the step size $\alpha^{(k)}$</td>
</tr>
<tr>
<td>7: $\alpha = \alpha^{(k)}$</td>
</tr>
<tr>
<td>8: while $h(\mu^{(k)}) - h(\mu^{(k)} - \alpha d^{(k)}) &lt; \sigma \alpha \cdot (\nabla h(\bar{\mu})^T) d^{(k)}$ do</td>
</tr>
<tr>
<td>9: $\alpha = \alpha^{(k)}$</td>
</tr>
<tr>
<td>10: end while</td>
</tr>
<tr>
<td>11: $\mu^{(k+1)} = h(\mu^{(k)}) - \alpha^{(k)} d^{(k)}$</td>
</tr>
<tr>
<td>12: $k = k + 1$</td>
</tr>
<tr>
<td>13: Recompute $d^{(0)}(\mu^{(k)}), g(\mu^{(k)}), \bar{g}^{(k)}$ and $h(\mu^{(k)})$</td>
</tr>
<tr>
<td>14: end while</td>
</tr>
</tbody>
</table>
1MHz unless specified otherwise and slot duration is set to 2ms. For the dynamic scheme, dynamic procedure is activated once every 4 video frames. Our objective is to maximize the sum Y-PSNR of all users. In other words, the weights for all users are set to 1. All results are average of ten simulation runs.

We consider three reference schemes. The first is the scheme in [14] where the user selection is based on maximum channel capacity and packets are dropped based on their priority at the time of buffer overflow. The buffer limit for each link is 110KBytes as used in [14]. Packets with the lowest priority are dropped first at the time of buffer overflow. This scheme is termed as Maximum capacity scheduling w/ FD (FD refers to frame dropping). The other two reference schemes are enhanced version of the Maximum capacity w/ FD where a different user scheduling algorithm is employed. The second scheme uses proportional fairness scheduling [23] and the third uses Modified Largest Weighted Delay First (M-LWDF) scheduling [3]. Note that in all the reference algorithms we employ the same frame prioritization and dropping strategy as in [14]. We choose these two enhanced scheduling algorithms because i) the Proportional Fairness scheduling is very widely used in wireless access networks, and ii) the M-LWDF scheduling is designed for real-time traffic and is shown in [13] to be one of the best scheduling algorithms for video streaming. In the following we evaluate the developed algorithms under four scenarios.

A. Variable mean SINR and different video sequences

In this scenario, we encode 8 video sequences with the SVC extension [16] of H.264/MPEG4-AVC: News, Hall, Silent, City, Foreman, Crew, Harbour, and Mobile, all of which are downloaded from [21]. The average SINR value of each user is uniformly distributed from 5dB to 20dB. The initial buffer duration is randomly generated from 700 milliseconds to 800 milliseconds. For fair comparison, we obtain the same initial buffer duration and the same SINR values for different schemes by using the same pseudo random number seed.

Figure 3 shows the obtained average PSNR of Y,U,V-components of all eight video sequences for different schemes (we connect the points that belong to the same scheme with lines simply to group them together). Although our algorithms are applied to improve the Y-PSNR in the simulations, they actually improve the PSNR of all other components.

The average PSNR over all all video sequences is summarized in Table III. It can be seen that both of our proposed schemes achieve significant gains over existing schemes. For the Y-PSNR, the static scheme achieves a gain of 1.5-5.4 dB and the dynamic scheme achieves a gain of 2.2-6.1 dB compared to the existing schemes. The improvement on U-PSNR and V-PSNR is less because i) it is not our objective to optimize the U and V-components, and ii) the color components appear less affected by dropping frames.

B. Same mean SINR, different video sequences

In the second scenario, we use the same 8 video sequences as in the previous scenario, but choose the average channel SINR for all users to be equal. We then investigate the video quality when the average channel SINR varies. Figure 4(a) shows the sum of the Y-PSNR of all eight video sequences under different SINR values. We can see that the proposed dynamic scheme always achieves the best video quality and the static scheme is slightly worse the the dynamic scheme. When the channel conditions are good, the M-LWDF scheduling algorithm with frame dropping obtains slightly higher PSNR than the static scheduling algorithm. However, when the average channel SINR decreases, the video quality obtained by all reference schemes including M-LWDF degrades very quickly. This is because at bad channel conditions, all reference schemes do not perform early dropping and may end up dropping important (low-layer) frames, which affects the decoding process of high-layer video packets. In the case of very low SINR, both of our schemes achieve an average gain of more than 6 dB compared to the three reference schemes.

Moreover, when the SINR is low, most other schemes
cannot decode the video sequences completely because of the heavy packet loss. Figure 4(b) shows the number of video sequences that are not decodable under each scheduling algorithm for different SINR values. Because of the early dropping strategy employed, the dynamic scheme can decode all video sequences for all SINR values greater or equal to 4dB, and the number of un-decodable sequences for the static scheme is much smaller than that of all reference schemes at the low SINR regime. For the static scheme, occasionally some video sequences may not be decoded completely. This is because the static scheme does not consider the instantaneous deadline requirement, which indicates that the dynamic scheme is required in order to have the best performance.

C. Variable mean SINR, same video sequences

In this set of simulations, all the users request the same Mobile video sequence (but the content is transmitted through unicast) and we let the number of users vary. The average SINRs (in dB) of the users are generated randomly using uniform distribution from 5dB to 20dB. To accommodate more users, we set the channel bandwidth to be 2.5MHz. Figure 5 shows the average PSNR of different schemes when the number of users ranges from 2 to 16. When the number of users is small, all scheduling algorithms perform very well except for the Maximum-capacity algorithm w/ FD. But when the number of users increases, our proposed schemes perform much better than the reference schemes. The static scheme and the dynamic one improve the average video quality by 8 dB and 10 dB, respectively, compared to the best of the three reference schemes when there are 16 users. This again demonstrates the efficacy of our proposed schemes.

D. Robustness under inaccurate channel model

In Sections III and IV, we assume precise information on the fading distributions. We now evaluate the performance of the algorithms when the fading distribution information is inaccurate. We assume Rayleigh distribution in our scheduling algorithms but let the actual fading distribution be Rician

with PDF function \( f(x|v, \sigma) = \frac{v}{\pi} \exp\left(-\frac{v^2+x^2}{\sigma^2}\right) I_0\left(\frac{2vx}{\sigma^2}\right) \), where \( I_0(z) \) is the modified Bessel function of the first kind with order zero. When \( v = 0 \), the distribution reduces to a Rayleigh distribution. A non-zero \( v \) indicates the deviation from Rayleigh distribution, and normally \( v^2 \leq \sigma^2 \). Other simulation setups are identical to those in Section V-A.

Figure 6 shows the average PSNR of the eight video sequences vs. \( v^2/\sigma^2 \). We maintain fixed SINR values when \( v^2/\sigma^2 \) changes. We can see that with the static scheme, the average PSNR drops about 0.8dB when \( v = \sigma \), and there is nearly no drop in the PSNR values for the dynamic scheme. Therefore, our algorithms (especially the dynamic one) is robust to the channel estimation errors.

VI. CONCLUSION

In this paper we study the problem of scalable video streaming in fading wireless environments. We exploit both the application layer video characteristics and the wireless channel fading information to obtain a cross-layer solution. We first develop a model to characterize the relationship between the average rate and average PSNR of a video stream. We then formulate the problem as a long-term radio resource allocation problem in a fading environment in order to maximize the weighted sum of average PSNR of all users. We develop an algorithm to find the optimal scheduling policy and the parameters used by the scheduling policy, and rigorously prove the optimality of the solution. We next design two scheduling algorithms based on the results of the long-term resource allocation scheme. Simulation results show that our scheduling algorithms are much superior to existing solutions and are robust to channel estimation errors.

REFERENCES

Proof of Theorem 2

We first prove a technical lemma.

Lemma 6 For any $1 \leq i \leq n$, $\sum_{j=1}^{n} \mu_j \frac{\partial g_i(\mu)}{\partial \mu_j} = 0$.

To prove the lemma, notice that if we re-scale the vector $\mu$ by a constant, the function $g_i(\mu)$ remains fixed. In other words, $g_i(\mu) = g_i(\tau \mu)$, where $\tau = 1/\mu_1$. Now consider the partial derivative $(\partial g_i(\mu))/\partial \mu_1$.

\[
\frac{\partial g_i(\mu)}{\partial \mu_1} = \lim_{\mu_1' \to \mu_1} \frac{g_i(\mu_1', \mu_2, \ldots, \mu_n) - g_i(\mu_1, \mu_2, \ldots, \mu_n)}{\mu_1' - \mu_1}
\]

\[
= \lim_{\mu_1' \to \mu_1} \sum_{j=2}^{n} \frac{\partial g_i(\mu)}{\partial \mu_j} \frac{\mu_j(\tau - 1)}{\mu_1' - \mu_1}
\]

\[
= \lim_{\tau \to 1} \sum_{j=2}^{n} \frac{\partial g_i(\mu)}{\partial \mu_j} \frac{\mu_j}{\mu_1}
\]

\[
= - \sum_{j=2}^{n} \frac{\partial g_i(\mu)}{\partial \mu_j} \frac{\mu_j}{\mu_1}.
\]

(23)

Re-arrange the equations, we obtain the result in the lemma.

We now prove theorem 2. From Lemma 5, algorithm A2 converges to a stationary point of the function $h(\mu)$. That means at the limit point $\mu$

\[
\nabla g(\mu) - \bar{g} = 0.
\]

(24)

In order to show the limit point $\mu$ is the optimal solution to problem (19) and (20), it is sufficient to show that $g_i(\mu) = \bar{g}$ for all $1 \leq i \leq n$. We prove using contradiction. Suppose that there exist $g_i(\mu)$'s that are not all equal and satisfy Eq. (24). Denote $w_i = g_i(\mu) - \bar{g}$ and $v_i = (w_1, \ldots, w_n)$. Then, we have $\nabla g \cdot w = 0$ where $w$ is a vector with zero mean. Without loss of generality, we can assume $w_i$'s are sorted in the decreasing order. Because $w$ is a zero-mean vector and its elements are not all equal, at least one element of $w$ is positive. Now suppose that $k$ is the index such that $w_1 \geq \ldots \geq w_k \geq 0 \geq w_{k+1} \geq \ldots \geq w_n$. For simplicity, we use the matrix $A$ to represent $\nabla g$ and $a_{ij} = \frac{\partial g_i}{\partial \mu_j}$. Note that in the matrix $A = \nabla g$, the diagonal elements are positive and the rest are negative. If $k = 1$, the first element of $Aw$ cannot be zero because it is equal to $\sum_{j=1}^{n} a_{1j}w_j$ where all terms are non-negative and the first term is positive. This is a contradiction. For $k \geq 1$, we look at the $i$th ($i \leq k$) element of $A$, which is $\sum_{j=1}^{n} a_{ij}w_j = 0$. Note that $a_{ij}w_j > 0$ for $j > k$. Therefore, $\sum_{j=1}^{k} a_{ij}w_j < 0$ and $a_{ii}w_i < \sum_{j=1}^{k} a_{ij}w_j$. Writing it in matrix form, we obtain

\[
\text{Diag}(w_1, \ldots, w_k) [a_{11}, a_{22}, \ldots, a_{kk}]^T < A_k w[k],
\]

where $A_k$ is a matrix with $k$ rows and $k$ columns and the element at the $i$th row and $j$th column of $A_k$ is $[a_{ij}]$ except that the diagonal elements are zero, $w[k]$ is a vector with the first $k$ elements of $w$, and the sign "<" is element-wise.

Since $w_1, \ldots, w_k$ are all positive, we have

\[
[a_{11}, a_{22}, \ldots, a_{kk}]^T < \text{Diag}(w_1, \ldots, w_k)^{-1} A_k w[k].
\]

(25)

If we write the results of Lemma 6 in matrix form and note that $A = \nabla g$, we have $\mu^TA = 0$ and so $A^T\mu = 0$. Still look at the $i$th ($i \leq k$) element of $A^T\mu$, and we have $\sum_{j=1}^{n} a_{ij}\mu_j = 0$. Note that all terms for $j > k$ are negative (because $\mu_j > 0$ and $a_{ij} < 0$ for $j > i$). So the summation of the first $k$ terms must be positive. That is, $\sum_{j=1}^{k} a_{ij}\mu_j > 0$.

Therefore, $\mu_i a_{ii} > \sum_{j=1}^{k} a_{ij}\mu_j a_{ji}$. Again, writing them in matrix form, we obtain (recall the definition of $A_k$ and $\mu[k]$ is a vector containing the first $k$ elements of $\mu$)

\[
\text{Diag}(\mu_1, \ldots, \mu_k) [a_{11}, a_{22}, \ldots, a_{kk}]^T > A_k^T \mu[k].
\]
Since \( \mu_i \)'s are all positive, we have
\[
[a_{11}, a_{22}, \ldots, a_{kk}]^T > \text{Diag}(\mu_1, \ldots, \mu_k)^{-1} A_k^T \mu_{[k]}.
\] (26)

Combining the two equations (25) and (26), we have
\[
\text{Diag}(\mu_1, \ldots, \mu_k)^{-1} A_k^T \mu_{[k]} < \text{Diag}(w_1, \ldots, w_k)^{-1} A_k w_{[k]}
\] (27)

Multiplying both sides in the left by \( \text{Diag}(\mu_1 w_1, \ldots, \mu_k w_k) \),
and note \( \mu_{[k]} = \text{Diag}(\mu_1, \ldots, \mu_k) J \) and \( w_{[k]} = \text{Diag}(w_1, \ldots, w_k) J \) where \( J = [1, 1, \ldots, 1]^T \) has \( k \) elements, we obtain
\[
\text{Diag}(w_1, \ldots, w_k) A_k^T \text{Diag}(\mu_1, \ldots, \mu_k) J < \text{Diag}(\mu_1, \ldots, \mu_k) A_k \text{Diag}(w_1, \ldots, w_k) J
\] (28)

Let \( B = \text{Diag}(\mu_1, \ldots, \mu_k) A_k \text{Diag}(w_1, \ldots, w_k) \). Eq. (28) can be written as
\[
B^T J < BJ.
\]

Thus, \( \text{sum}(B^T J) < \text{sum}(BJ) \). However, \( \text{sum}(B^T J) = \text{sum}(BJ) \) because both are equal to the summation of all elements in the matrix \( B \). Therefore, this is a contradiction and the proof is complete. 

\[\blacksquare\]